



## ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ ΙΙ

Ανάπτυξη Λογισμικού για την Kavala Oil

### ΕΛΕΓΚΤΕΣ (TESTERS)



Ελεγκτής Α' : Υφαντής Γεώργιος

Ελεγκτής Β' : Αναστασιάδου Έλενα

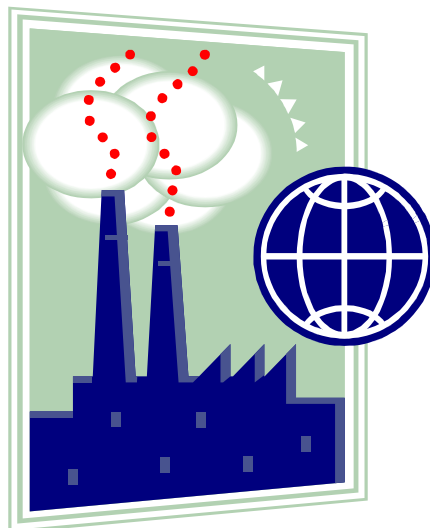
# ΠΕΡΙΕΧΟΜΕΝΑ

Σελίδα

<b>Κεφάλαιο 1 - Γενικό σχέδιο Ελέγχου.....</b>	<b>3</b>
<b>Κεφάλαιο 2 - Έλεγχος Λογισμικού.....</b>	<b>21</b>
Α' Έλεγχος – Κλειστό κουτί : .....	21
Α' Έλεγχος – Ανοιχτό κουτί : .....	38
Β' Έλεγχος – Κλειστό κουτί : .....	48
Β' Έλεγχος – Ανοιχτό κουτί : .....	87
Γ' Έλεγχος – Κλειστό κουτί : .....	108
Γ' Έλεγχος – Ανοιχτό κουτί : .....	157
<b>Κεφάλαιο 3 - Τεκμηρίωση Ελέγχου.....</b>	<b>182</b>

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΤΗΝ ΜΕΛΕΤΗ ΦΑΙΝΟΜΕΝΩΝ ΠΟΥ  
ΜΠΟΡΕΙ ΝΑ ΔΗΜΙΟΥΡΓΗΘΟΥΝ ΜΕΤΑ ΑΠΟ ΕΚΡΗΞΗ ΣΕ ΚΑΠΟΙΑ  
ΔΕΞΑΜΕΝΗ ΤΗΣ ΒΙΟΜΗΧΑΝΙΑΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΠΕΤΡΕΛΑΙΟΥ  
ΚΑΒΑΛΑ Oil

## Γενικό Σχέδιο Ελέγχου



Ελεγκτής Α' : Υφαντής Γεώργιος  
Ελεγκτής Β' : Αναστασιάδου Έλενα

## **ΓΕΝΙΚΟ ΣΧΕΔΙΟ ΤΕΣΤ**

✚ **Σκοπός του Τεστ :** Είναι έλεγχος του λογισμικού για την βιομηχανία επεξεργασίας πετρελαίου Kavala Oil, που βρίσκεται στην Βιομηχανική περιοχή Καβάλας, και ειδικότερα ο έλεγχος των φαινομένων:

- Flame Jet / Flare,
- BLEVE (Boiling Liquid Expanding Vapor Explosion / Fire Ball),
- και Pool Fire..

τα οποία μπορεί να συμβούν μετά από έκρηξη σε κάποια δεξαμενή της.

Ο έλεγχος θα αφορά :

- Ύπαρξη λαθών στον κώδικα του προγράμματος .
- Ανεπαρκής πληρότητα λογισμικού βάσει των απαιτήσεων που έδωσε ο πελάτης .
- Ύπαρξη λαθών, προερχόμενα από εξωτερικούς παράγοντες όπως οι χρήστες του λογισμικού.

✚ **Προσέγγιση :** Για την πραγματοποίηση του παραπάνω ελέγχου θα χρησιμοποιηθούν οι εξής μέθοδοι :

- Ισοδύναμες κλάσεις
- Οριακές τιμές
- Εικασίες λανθασμένων τιμών
- Εικασίες λαθών

✚ **Μέσα :** Για την πραγματοποίηση του ελέγχου θα εργαστούν οι ελεγκτές:

- ΥΦΑΝΤΗΣ ΓΕΩΡΓΙΟΣ
- ΑΝΑΣΤΑΣΙΑΔΗ ΕΛΕΝΗ

οι οποίοι θα καλύψουν όλες τις φάσεις ανάπτυξης test. Τέλος οι ελεγκτές καθ' όλη την διάρκεια διεκπεραίωσης του λογισμικού θα πρέπει να είναι ενημερωμένοι για τυχόν αλλαγές στις προδιαγραφές, οι οποίες θα επηρεάσουν την περαιτέρω έκβαση του test.

✚ **Κριτήρια τερματισμού :** Ο έλεγχος του λογισμικού θα τερματίσει για να παραδοθεί από τους ελεγκτές στον πελάτη όταν :

- Δεν θα εντοπιστούν από τους ελεγκτές άλλα λάθη στον κώδικα του προγράμματος .
- Το λογισμικό θα πληρεί όλες τις απαιτήσεις του πελάτη .
- Η ύπαρξη λαθών από εξωτερικούς παράγοντες όπως οι χρήστες του λογισμικού θα ελέγχεται από το πρόγραμμα κατάλληλα.

## Περίπτώσεις TEST -> επιλογή φαινομένου

### Ισοδύναμες κλάσεις :

- Εισαγωγή τιμής int π.χ. 0 → Κανονική αποδοχή
- Εισαγωγή τιμής int π.χ. 1 → Κανονική αποδοχή
- Εισαγωγή τιμής int π.χ. 2 → Κανονική αποδοχή
- Εισαγωγή τιμής int π.χ. 3 → Κανονική αποδοχή
- Εισαγωγή τιμής int π.χ. 4 → Μη αποδοχή και επανάληψη της επιλογής φαινομένου.
- Εισαγωγή τιμής int π.χ. 15 → Μη αποδοχή και επανάληψη της επιλογής φαινομένου.
- Εισαγωγή τιμής int π.χ. 0.1 → Μη αποδοχή και επανάληψη της επιλογής φαινομένου.
- Εισαγωγή τιμής int π.χ. 1.5 → Μη αποδοχή και επανάληψη της επιλογής φαινομένου.
- Εισαγωγή γράμματος π.χ. α → Μη αποδοχή και επανάληψη της επιλογής φαινομένου.
- Εισαγωγή συμβόλου π.χ. / → Μη αποδοχή και επανάληψη της επιλογής φαινομένου.

### Οριακές τιμές :

- Για τιμές int από 0 έως 3 → Κανονική αποδοχή τιμής
- Για οποιαδήποτε τιμή int εκτός από 0,1,2,3 → Μη αποδοχή και επανάληψη της επιλογής φαινομένου.
- Για οτιδήποτε εκτός τιμής int → Μη αποδοχή και επανάληψη της επιλογής φαινομένου.

### Εικασίες λαθών:

- Ποιες τιμές γίνονται αποδεκτές για την *επιλογή φαινομένου*;
- Γίνεται πρόσβαση στο σύστημα χωρίς *επιλογή φαινομένου*;
- Τι γίνεται στην περίπτωση που εισαχθεί γράμμα ή σύμβολο π.χ. (=,#,^....)ή άλλος αριθμός (4,100,0.3,2.8...);
- Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

## Διαδικασίες TEST

- ◆ Περίπτωση : Ποίες τιμές γίνονται αποδεκτές για την *επιλογή φαινομένου*;

Διαδικασία :

- Οι τιμές 0,1,2,3 τύπου int (ακέραιος).

- ◆ Περίπτωση : Γίνεται πρόσβαση στο σύστημα χωρίς *επιλογή φαινομένου*;

Διαδικασία :

- Μη αποδοχή και επανάληψη της επιλογής φαινομένου.

- ◆ Περίπτωση : Τι γίνεται στην περίπτωση που εισαχθεί γράμμα ή σύμβολο π.χ. (=,#,^....)ή άλλος αριθμός (4,100,0.3,2.8...);

Διαδικασία

- Μήνυμα λάθους μη αποδοχή και επανάληψη της επιλογής φαινομένου.

- ◆ Περίπτωση : Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

Διαδικασία :

- Μήνυμα λάθους μη αποδοχή και επανάληψη της επιλογής φαινομένου.

## Περιπτώσεις TEST -> επιλογή ρυθμού διαρροής(BLEVE)

### Ισοδύναμες κλάσεις :

- Εισαγωγή τιμής π.χ. 150000 → Κανονική αποδοχή
- Εισαγωγή τιμής π.χ. 264670,92 → Κανονική αποδοχή
- Εισαγωγή τιμής π.χ. 650000 → Κανονική αποδοχή
- Εισαγωγή τιμής π.χ. 150001 → Κανονική αποδοχή
- Εισαγωγή τιμής π.χ. 649999 → Κανονική αποδοχή
- Εισαγωγή τιμής π.χ. 149999 → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
- Εισαγωγή τιμής π.χ. 650001 → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
- Εισαγωγή τιμής π.χ. -1234 → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
- Εισαγωγή γράμματος π.χ. α → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
- Εισαγωγή συμβόλου π.χ. / → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.

### Οριακές τιμές :

- Για τιμές από 150000 έως 650000 → Κανονική αποδοχή τιμής
- Για οποιαδήποτε τιμή <150000 και >650000 → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
- Για οτιδήποτε εκτός από αριθμούς → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.

### Εικασίες λαθών:

- Ποιές τιμές γίνονται αποδεκτές για την *επιλογή* ρυθμού διαρροής;
- Γίνεται πρόσβαση στο σύστημα χωρίς *επιλογή* ρυθμού διαρροής;
- Τι γίνεται στην περίπτωση που εισαχθεί γράμμα ή σύμβολο π.χ. (=,#,^....)ή άλλος αριθμός (1.4,100,7.1...);
- Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

## Διαδικασίες TEST

- ◆ Περίπτωση : Ποιές τιμές γίνονται αποδεκτές για την *επιλογή* ρυθμού διαρροής;  
Διαδικασία :
  - Οι τιμές από 150000 έως 650000 .
  
- ◆ Περίπτωση : Γίνεται πρόσβαση στο σύστημα χωρίς *επιλογή* ρυθμού διαρροής;  
Διαδικασία :
  - Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
  
- ◆ Περίπτωση : Τι γίνεται στην περίπτωση που εισαχθεί γράμμα ή σύμβολο π.χ. (=,#,^....)ή άλλος αριθμός (1.4,100,7.1...);  
Διαδικασία :
  - Μήνυμα λάθους μη αποδοχή και επανάληψη της ρυθμού διαρροής.
  
- ◆ Περίπτωση : Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;  
Διαδικασία :
  - Μήνυμα λάθους μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.

## Περιοπτώσεις TEST -> επιλογή ζώνης

### Ισοδύναμες κλάσεις :

- Εισαγωγή τιμής int π.χ. 0 → Κανονική αποδοχή
- Εισαγωγή τιμής int π.χ. 1 → Κανονική αποδοχή
- Εισαγωγή τιμής int π.χ. 2 → Κανονική αποδοχή
- Εισαγωγή τιμής int π.χ. 3 → Κανονική αποδοχή
- Εισαγωγή τιμής int π.χ. 4 → Κανονική αποδοχή
- Εισαγωγή τιμής int π.χ. 5 → Μη αποδοχή και επανάληψη της επιλογής ζώνης.
- Εισαγωγή τιμής int π.χ. 15 → Μη αποδοχή και επανάληψη της επιλογής ζώνης.
- Εισαγωγή τιμής int π.χ. 0.1 → Μη αποδοχή και επανάληψη της επιλογής ζώνης.
- Εισαγωγή τιμής int π.χ. 1.5 → Μη αποδοχή και επανάληψη της επιλογής ζώνης.
- Εισαγωγή γράμματος π.χ. α → Μη αποδοχή και επανάληψη της επιλογής ζώνης.
- Εισαγωγή συμβόλου π.χ. / → Μη αποδοχή και επανάληψη της επιλογής ζώνης.

### Οριακές τιμές :

- Για τιμές int από 0 έως 4 → Κανονική αποδοχή τιμής
- Για οποιαδήποτε τιμής int εκτός από 0,1,2,3,4 → Μη αποδοχή και επανάληψη της επιλογής ζώνης.
- Για οτιδήποτε εκτός τιμής int → Μη αποδοχή και επανάληψη της επιλογής ζώνης.

### Εικασίες λαθών:

- Ποιές τιμές γίνονται αποδεκτές για την *επιλογή* ζώνης;
- Γίνεται πρόσβαση στο σύστημα χωρίς *επιλογή* ζώνης;
- Τι γίνεται στην περίπτωση που εισαχθεί γράμμα ή σύμβολο π.χ. (=,#,^....)ή άλλος αριθμός (5,100,0.3,2.8...);
- Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

## Διαδικασίες TEST

◆ Περίπτωση : Ποιες τιμές γίνονται αποδεκτές για την *επιλογή* ζώνης;  
Διαδικασία :

- Οι τιμές 0,1,2,3,4 τύπου int (ακέραιος).

◆ Περίπτωση : Γίνεται πρόσβαση στο σύστημα χωρίς *επιλογή* ζώνης;  
Διαδικασία :

- Μη αποδοχή και επανάληψη της επιλογής ζώνης.

◆ Περίπτωση : Τι γίνεται στην περίπτωση που εισαχθεί γράμμα ή σύμβολο π.χ. (=,#,^....)ή άλλος αριθμός (5,100,0.3,2.8...);  
Διαδικασία

- Μήνυμα λάθους μη αποδοχή και επανάληψη της επιλογής ζώνης.

◆ Περίπτωση : Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;  
Διαδικασία :

- Μήνυμα λάθους μη αποδοχή και επανάληψη της επιλογής ζώνης.

## Περιπτώσεις TEST -> επιλογή ρυθμού διαρροής (FLAME JET / FLARE)

### Ισοδύναμες κλάσεις :

- Εισαγωγή τιμής π.χ. 0,17 → Κανονική αποδοχή
- Εισαγωγή τιμής π.χ. 0,92 → Κανονική αποδοχή
- Εισαγωγή τιμής π.χ. 1,00 → Κανονική αποδοχή
- Εισαγωγή τιμής π.χ. 1,01 → Κανονική αποδοχή
- Εισαγωγή τιμής π.χ. 1,30 → Κανονική αποδοχή
- Εισαγωγή τιμής π.χ. 0,16 → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
- Εισαγωγή τιμής π.χ. 1,31 → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
- Εισαγωγή τιμής π.χ. 8 → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
- Εισαγωγή γράμματος π.χ. α → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
- Εισαγωγή συμβόλου π.χ. / → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.

### Οριακές τιμές :

- Για τιμές από 0,17 έως 1,30 → Κανονική αποδοχή τιμής
- Για οποιαδήποτε τιμή <0,17 και >1,30 → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
- Για οτιδήποτε εκτός από αριθμούς → Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.

### Εικασίες λαθών:

- Ποιές τιμές γίνονται αποδεκτές για την *επιλογή* ρυθμού διαρροής;
- Γίνεται πρόσβαση στο σύστημα χωρίς *επιλογή* ρυθμού διαρροής;
- Τι γίνεται στην περίπτωση που εισαχθεί γράμμα ή σύμβολο π.χ. (=,#,^,...)ή άλλος αριθμός (1.4,100,7.1...);
- Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

## Διαδικασίες TEST

- ◆ Περίπτωση : Ποιές τιμές γίνονται αποδεκτές για την *επιλογή* ρυθμού διαρροής;  
Διαδικασία :
  - Οι τιμές από 0,17 έως 1,30 .
- ◆ Περίπτωση : Γίνεται πρόσβαση στο σύστημα χωρίς *επιλογή* ρυθμού διαρροής;  
Διαδικασία :
  - Μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.
- ◆ Περίπτωση : Τι γίνεται στην περίπτωση που εισαχθεί γράμμα ή σύμβολο π.χ. (=,#,^....)ή άλλος αριθμός (1.4,100,7.1...);  
Διαδικασία
  - Μήνυμα λάθους μη αποδοχή και επανάληψη της ρυθμού διαρροής.
- ◆ Περίπτωση : Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;  
Διαδικασία :
  - Μήνυμα λάθους μη αποδοχή και επανάληψη της επιλογής ρυθμού διαρροής.

## Περιπτώσεις TEST -> *όνομα*

### Ισοδύναμες κλάσεις:

- Εισαγωγή τιμής String στο όνομα π.χ PAUL → Κανονική αποδοχή
- Εισαγωγή τιμής String στο όνομα π.χ ΠΑΥΛΟΣ → Κανονική αποδοχή
- Εισαγωγή τιμής String στο όνομα π.χ paul → Κανονική αποδοχή
- Εισαγωγή τιμής String στο όνομα π.χ παύλος → Κανονική αποδοχή
- Εισαγωγή τιμής String στο όνομα π.χ Paul → Κανονική αποδοχή
- Εισαγωγή τιμής String στο όνομα π.χ Παύλος → Κανονική αποδοχή
- Εισαγωγή τιμής String στο όνομα π.χ 123 → Μη αποδοχή και επανάληψη της εισαγωγής του ονόματος.
- Εισαγωγή τιμής String στο όνομα π.χ / → Μη αποδοχή και επανάληψη της εισαγωγής του ονόματος.

### Οριακές τιμές:

- Για μήκος ονόματος από 3 έως 10 χαρακτήρες → Κανονική αποδοχή τιμής
- Για μήκος ονόματος μικρότερο από 3 χαρακτήρες ή μήκος ονόματος μεγαλύτερο από 10 χαρακτήρες → Μη αποδοχή και επανάληψη της εισαγωγής του ονόματος.

### Εικασίες λαθών:

- Ποιες τιμές γίνονται αποδεκτές σαν *όνομα*;
- Που καταχωρείται το *όνομα*;
- Γίνεται πρόσβαση στο σύστημα χωρίς *όνομα*;
- Τι γίνεται στην περίπτωση που εισαχθεί αριθμός ή άλλος χαρακτήρας π.χ. (=, #, ^, ...) στην αρχή ή στην μέση του ονόματος;
- Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

## Διαδικασίες TEST

- ◆ Περίπτωση : Ποιες τιμές γίνονται αποδεκτές σαν *όνομα*;

Διαδικασία :

- Όλες οι τιμές τύπου String (γράμματα )στα ελληνικά ή στα αγγλικά.
- Το πλήθος των χαρακτήρων να είναι μέχρι 10 χωρίς κενά ενδιάμεσα.

- ◆ Περίπτωση : Που καταχωρείται το *όνομα*;

Διαδικασία :

- Καταχώρηση σε ένα αρχείο .

- ◆ Περίπτωση : Γίνεται πρόσβαση στο σύστημα χωρίς *όνομα*;

Διαδικασία :

- Ειδοποίηση για λανθασμένο *όνομα* επανάληψη της εισαγωγής.

- ◆ Περίπτωση:Τι γίνεται στην περίπτωση που εισαχθεί αριθμός ή άλλος χαρακτήρας π.χ. (=,#,^....) στην αρχή ή στην μέση του ονόματος;

Διαδικασία :

- Ειδοποίηση στην οθόνη για μη έγκυρη εισαγωγή ονόματος και επανάληψη της εισαγωγής.

- ◆ Περίπτωση : Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

Διαδικασία :

- Ειδοποίηση στην οθόνη για μη εισαγωγή τιμής και επανάληψη της εισαγωγής.

## Περιπτώσεις TEST -> επώνυμο

### Ισοδύναμες κλάσεις:

- Εισαγωγή τιμής String στο επώνυμο π.χ. PΑΡΑΔΟΡΟΥΛΟΣ  
➔ Κανονική αποδοχή
- Εισαγωγή τιμής String στο επώνυμο π.χ. ΠΑΠΑΔΟΠΟΥΛΟΣ ➔ Κανονική αποδοχή
- Εισαγωγή τιμής String στο επώνυμο π.χ. παπαδόπουλος ➔ Κανονική αποδοχή
- Εισαγωγή τιμής String στο επώνυμο π.χ. papadopoulos ➔ Κανονική αποδοχή
- Εισαγωγή τιμής String στο επώνυμο π.χ. Papadopoulos ➔ Κανονική αποδοχή
- Εισαγωγή τιμής String στο επώνυμο π.χ. Παπαδόπουλος ➔ Κανονική αποδοχή
- Εισαγωγή τιμής String στο επώνυμο π.χ 123 ➔ Μη αποδοχή και επανάληψη της εισαγωγής του επώνυμου.
- Εισαγωγή τιμής String στο επώνυμο π.χ / ➔ Μη αποδοχή και επανάληψη της εισαγωγής του επώνυμου.

### Οριακές τιμές:

- Για μήκος επώνυμου από 3 έως 15 χαρακτήρες ➔ Κανονική αποδοχή τιμής
- Για μήκος επώνυμου μικρότερο από 3 χαρακτήρες ή μήκος επώνυμου μεγαλύτερο από 10 χαρακτήρες ➔ Μη αποδοχή και επανάληψη της εισαγωγής του επώνυμου.

### Εικασίες λαθών:

- Ποιές τιμές γίνονται αποδεκτές σαν *επώνυμο*;
- Που καταχωρείται το *επώνυμο*;
- Γίνεται πρόσβαση στο σύστημα χωρίς *επώνυμο*;
- Τι γίνεται στην περίπτωση που εισαχθεί αριθμός ή άλλος χαρακτήρας π.χ. (=,#,^....) στην αρχή ή στην μέση του ονόματος;
- Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

## Διαδικασίες TEST

- ◆ Περίπτωση : Ποιές τιμές γίνονται αποδεκτές σαν *επώνυμο*;

### Διαδικασία :

- Όλες οι τιμές τύπου String (γράμματα )στα ελληνικά ή στα αγγλικά.
- Το πλήθος των χαρακτήρων να είναι μέχρι 10 χωρίς κενά ενδιάμεσα.

- ◆ Περίπτωση : Που καταχωρείται το *επώνυμο*;

### Διαδικασία :

- Καταχώρηση σε ένα αρχείο .

- ◆ Περίπτωση : Γίνεται πρόσβαση στο σύστημα χωρίς *επώνυμο*;

### Διαδικασία :

- Ειδοποίηση για λανθασμένο *επώνυμο* επανάληψη της εισαγωγής.

- ◆ Περίπτωση:Τι γίνεται στην περίπτωση που εισαχθεί αριθμός ή άλλος χαρακτήρας π.χ. (=,#,^....) στην αρχή ή στην μέση του επωνύμου;

### Διαδικασία :

- Ειδοποίηση στην οθόνη για μη έγκυρη εισαγωγή επωνύμου και επανάληψη της εισαγωγής.

- ◆ Περίπτωση : Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

### Διαδικασία :

- Ειδοποίηση στην οθόνη για μη εισαγωγή τιμής και επανάληψη της εισαγωγής.

## Περιοπτώσεις TEST -> *UserName*

### Ισοδύναμες κλάσεις:

- Εισαγωγή τιμής String στο *UserName* π.χ ΡΑΡΑΔΟ → Κανονική αποδοχή
- Εισαγωγή τιμής String στο *UserName* π.χ ΠΑΠΑΔΟ → Κανονική αποδοχή
- Εισαγωγή τιμής String στο *UserName* π.χ rapado → Κανονική αποδοχή
- Εισαγωγή τιμής String στο *UserName* π.χ παπαδο → Κανονική αποδοχή
- Εισαγωγή τιμής String στο *UserName* π.χ Papado → Κανονική αποδοχή
- Εισαγωγή τιμής String στο *UserName* π.χ Παπαδο → Κανονική αποδοχή
- Εισαγωγή τιμής String στο *UserName* π.χ 123 → Μη αποδοχή και επανάληψη της εισαγωγής του *UserName*.
- Εισαγωγή τιμής String στο *UserName* π.χ / → Μη αποδοχή και επανάληψη της εισαγωγής του *UserName*.

### Οριακές τιμές:

- Για μήκος *UserName* από 3 έως 10 χαρακτήρες → Κανονική αποδοχή τιμής
- Για μήκος *UserName* μικρότερο από 3 χαρακτήρες ή μήκος *UserName* μεγαλύτερο από 10 χαρακτήρες → Μη αποδοχή και επανάληψη της εισαγωγής του *UserName*.

### Εικασίες λαθών:

- Ποιες τιμές γίνονται αποδεκτές σαν *UserName*;
- Που καταχωρείται το *UserName*;
- Γίνεται πρόσβαση στο σύστημα χωρίς *UserName*;
- Τι γίνεται στην περίπτωση που εισαχθεί αριθμός ή άλλος χαρακτήρας π.χ. (=,#,^....) στην αρχή ή στην μέση του *UserName*;
- Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

## Διαδικασίες TEST

- ◆ Περίπτωση : Ποιές τιμές γίνονται αποδεκτές σαν UserName;

### Διαδικασία :

- Όλες οι τιμές τύπου String (γράμματα )στα ελληνικά ή στα αγγλικά.
- Το πλήθος των χαρακτήρων να είναι μέχρι 10 χωρίς κενά ενδιάμεσα.

- ◆ Περίπτωση : Που καταχωρείται το UserName;

### Διαδικασία :

- Καταχώρηση σε ένα αρχείο .

- ◆ Περίπτωση : Γίνεται πρόσβαση στο σύστημα χωρίς UserName;

### Διαδικασία :

- Ειδοποίηση για λανθασμένο UserName επανάληψη της εισαγωγής.

- ◆ Περίπτωση:Τι γίνεται στην περίπτωση που εισαχθεί αριθμός ή άλλος χαρακτήρας π.χ. (=,#,^....) στην αρχή ή στην μέση του UserName;

### Διαδικασία :

- Ειδοποίηση στην οθόνη για μη έγκυρη εισαγωγή UserName και επανάληψη της UserName.

- ◆ Περίπτωση : Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

### Διαδικασία :

- Ειδοποίηση στην οθόνη για μη εισαγωγή τιμής και επανάληψη της εισαγωγής.

## Περιοπτώσεις TEST -> password

### Ισοδύναμες κλάσεις:

- Εισαγωγή τιμής String στο password π.χ PAP12 → Κανονική αποδοχή
- Εισαγωγή τιμής String στο password π.χ ΠΑΠ1 → Κανονική αποδοχή
- Εισαγωγή τιμής String στο password π.χ pap456 → Κανονική αποδοχή
- Εισαγωγή τιμής String στο password π.χ πα7 → Κανονική αποδοχή
- Εισαγωγή τιμής String στο password π.χ Pap129 → Κανονική αποδοχή
- Εισαγωγή τιμής String στο password π.χ Πα6788 → Κανονική αποδοχή
- Εισαγωγή τιμής String στο password π.χ 123 → Μη αποδοχή και επανάληψη της εισαγωγής του password
- Εισαγωγή τιμής String στο password π.χ / → Μη αποδοχή και επανάληψη της εισαγωγής του password.
- Εισαγωγή τιμής String στο password π.χ PAPADO → Μη αποδοχή και επανάληψη της εισαγωγής του password.

### Οριακές τιμές:

- Για μήκος password από 3 έως 7 μήκος → Κανονική αποδοχή τιμής
- Από 1 έως 6 χαρακτήρες και από 1 έως 6 αριθμούς. → Κανονική αποδοχή τιμής
- Για μήκος password μικρότερο από 3 ή μεγαλύτερο από 7 → Μη αποδοχή και επανάληψη της εισαγωγής του password
- Για μήκος χαρακτήρων μικρότερο από 1 ή μεγαλύτερο από 6 ή για μήκος αριθμών μικρότερο από 1 ή μεγαλύτερο από 6 → Μη αποδοχή και επανάληψη της εισαγωγής του password.

### Εικασίες λαθών:

- Ποιές τιμές γίνονται αποδεκτές σαν password;
- Που καταχωρείται το password;
- Γίνεται πρόσβαση στο σύστημα χωρίς password;
- Τι γίνεται στην περίπτωση που εισαχθεί αριθμός ή άλλος χαρακτήρας π.χ. (=,#,^....) στην αρχή ή στην μέση του password;
- Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

## Διαδικασίες TEST

- ◆ Περίπτωση : Ποιές τιμές γίνονται αποδεκτές σαν password;

### Διαδικασία :

- Όλες οι τιμές τύπου String (γράμματα )στα ελληνικά ή στα αγγλικά.
- Το πλήθος των χαρακτήρων να είναι μέχρι 7 χωρίς κενά ενδιάμεσα.
- Το πλήθος από 1 έως 6 χαρακτήρες και από 1 έως 6 αριθμούς.

- ◆ Περίπτωση : Που καταχωρείται το password;

### Διαδικασία :

- Καταχώρηση σε ένα αρχείο .

- ◆ Περίπτωση : Γίνεται πρόσβαση στο σύστημα χωρίς password;

### Διαδικασία :

- Ειδοποίηση για λανθασμένο password επανάληψη της εισαγωγής.

- ◆ Περίπτωση:Τι γίνεται στην περίπτωση που εισαχθεί αριθμός ή άλλος χαρακτήρας π.χ. (=,#,^....) στην αρχή ή στην μέση του password;

### Διαδικασία :

- Ειδοποίηση στην οθόνη για μη έγκυρη εισαγωγή password και επανάληψη της password.

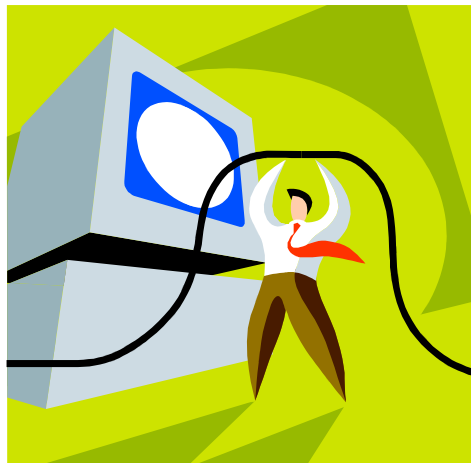
- ◆ Περίπτωση : Τι γίνεται στην περίπτωση που δεν εισαχθεί καμία τιμή και πατηθεί το Enter;

### Διαδικασία :

- Ειδοποίηση στην οθόνη για μη εισαγωγή τιμής και επανάληψη της εισαγωγής.

# Α' Έλεγχος

## Κλειστού Κουτιού



Ελεγκτής Α' : Υφαντής Γεώργιος  
Ελεγκτής Β' : Αναστασιάδου Έλενα

## **ΗΜΕΡΟΛΟΓΙΟ ΕΛΕΓΚΤΗ: 9 Μαΐου 2004**

**Ver 1.0**

### **Ελέγχθηκαν τα αρχεία :**

● **Bleve2.cpp**

### **και οι συναρτήσεις :**

- **Bleve**
- **FlameJet**
- **MainMenu**
- **ZoneSelect**
- **Default**

**Στο αρχείο bleve2.cpp**

### **Συνάρτηση MainMenu :**

Στην συνάρτηση MainMenu οι τιμές που θα δοθούν στον αναλυτή για την εξέταση ορίζονται από τον χρήστη του λογισμικού και είναι αυστηρά καθορισμένες απ' τις επιλογές που καθορίζονται απ' το λογισμικό. Έτσι λοιπόν, οι τιμές που θα εισάγονται δεν υπάρχει περίπτωση να δημιουργήσουν εικασίες λανθασμένων τιμών όπως για παράδειγμα:

- ◆ Καμία τιμή (Enter)
- ◆ Γράμματα ή άλλοι χαρακτήρες (π.χ. @,#, ",δ,a,?,B )
- ◆ Κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Μεγάλο πλήθος ακεραίων που δέχεται
- ◆ Σωστή ενημέρωση χρήστη
- ◆ Μη ακέραια τιμή
- ◆ Αρνητική τιμή
- ◆ Τιμές εκτός των ορίων

## Παρατηρήσεις:

Έλεγχος συνάρτησης int MainMenu() → **MainMenu→Επιλογήφαινομένου**

### Ισοδύναμες κλάσεις τιμών :

- ◆ < 0 (Άκυρες κλάσεις)
- ◆ > 3 (Άκυρες κλάσεις)
- ◆  $0 \leq \text{τιμή} \leq 3$  (Έγκυρες κλάσεις)

### Οριακές τιμές :

- ◆ 0.1, -0.1, 0.0 (Οριακές τιμές)
- ◆ 1.1, 0.9, 1.0 (Οριακές τιμές)
- ◆ 2.1, 2.0, 1.9 (Οριακές τιμές)
- ◆ 3.1, 3.0, 2.9 (Οριακές τιμές)

### Εικασίες τιμών :

- ◆ Καμία τιμή (Enter)
- ◆ Μεγάλη τιμή (πχ 10000)
- ◆ Αρνητική τιμή (πχ -254)
- ◆ Χαρακτήρας (πχ α, s, %)
- ◆ Κενό

Ισοδύναμες κλάσεις τιμών (τιμές εισόδου)	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
0 (έγκυρη κλάση)	Έξοδος 0	Έξοδος 0	✓
1(έγκυρη κλάση)	ZoneSelect 1	ZoneSelect 1	✓
2(έγκυρη κλάση)	FlameJet 2	FlameJet 2	✓
3(έγκυρη κλάση)	MainMenu	PoolFire	✗
-5(άκυρη κλάση)	MainMenu	Εμφάνιση μηνύματος λάθους & επιστροφή στην MainMenu	✗
5(άκυρη κλάση)	MainMenu	Εμφάνιση μηνύματος λάθους & επιστροφή στην MainMenu	✗
Οριακές τιμές	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
0.0	Έξοδος 0	Έξοδος 0	✓
0.1	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
- 0.1	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
1.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
1.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
0.9	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
2.0	Εμφάνιση λάθος αποτελεσμάτων και επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	✗
2.1	Εμφάνιση λάθος αποτελεσμάτων και επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	✗
1.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
2.9	Εμφάνιση λάθος αποτελεσμάτων και επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	✗

Εικασίες τιμών	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
Καμία τιμή (Enter)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×
1000 (Μεγάλη τιμή)	Επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	×
- 254 (Αρνητική τιμή)	Επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	×
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
_ (κενο)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ δεν παίρνει ως τιμή χαρακτήρες
- ◆ δεν παίρνει ως τιμή δεκαδικούς αριθμούς
- ◆ δεν παίρνει ως τιμή αρνητικούς αριθμούς
- ◆ παίρνει τιμές από 0 έως 3 μόνο

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν αρκετά λάθη, από την εισαγωγή τιμών εκτός ορίων, μη επιτρεπόμενων τιμών (όπως για παράδειγμα χαρακτήρες, αρνητικοί αριθμοί, δεκαδικά ψηφία κλπ). Υπάρχουν όμως κάποιες παρατηρήσεις που η εφαρμογή τους είναι απαραίτητη σε κάποια σημεία της μεθόδου αυτά είναι:

- Κατά την εισαγωγή των τιμών απ' τον χρήστη, δεν υπάρχει η αντίστοιχη κλάση εξαιρέσεων για τον πλήρη έλεγχο των τιμών που εισάγονται.
- Το φαινόμενο 3. Pool Fire δεν είναι ακόμη έτοιμο και γι' αυτό υπάρχουν λάθη που αντιστοιχούν σ' αυτή την επιλογή.
- Είναι προτεινόμενο να δημιουργηθούν κάποιοι έλεγχοι για όλες αυτές τις περιπτώσεις σφαλμάτων που προκύπτουν.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 3-λάθη 21)

## Συνάρτηση Bleve, ZoneSelect και Default:

Στην συνέχεια, απ' την επιλογή του MainMenu, πληκτρολογώντας 1 για το φαινόμενο Bleve, ο χρήστης, καλεί στην ουσία την συνάρτηση *void Default()* όπου η συνάρτηση αυτή καλεί και τις άλλες δύο την *void Bleve(int zone, double &Rs, double &Ts, double &F, double &I, double &L)* και την *int ZoneSelect()*, όπου η συνάρτηση Bleve, υπολογίζει από προκαθορισμένους τύπους α) την ακτίνα πύρινης σφαίρας, β) την χρονική διάρκεια του φαινομένου, γ) το ποσοστό εκπεμπόμενης ακτινοβολίας, δ) την θερμική ακτινοβολία και ε) την απόσταση του παρατηρητή. Εδώ ο χρήστης έχει να εισάγει ποια απ' τις προτεινόμενες τέσσερις ζώνες επιθυμεί να ελέγξει. Πληκτρολογώντας και την επιθυμητή ζώνη, το λογισμικό πρέπει να του εμφανίσει κάποια στοιχεία τα οποία θα ελεγχθούν παρακάτω. Στην ουσία λοιπόν, θα ελέγξουμε την εισαγωγή τιμών μέσω της συνάρτησης *ZoneSelect* και την εγκυρότητα των τιμών που επιστρέφει η συνάρτηση *Bleve*. Έτσι λοιπόν, οι τιμές που θα εισάγονται δεν υπάρχει περίπτωση να δημιουργήσουν εικασίες λανθασμένων τιμών όπως για παράδειγμα:

- ◆ Καμία τιμή (Enter)
- ◆ Γράμματα ή άλλοι χαρακτήρες (π.χ. @,#,"δ,a,?,B )
- ◆ Κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Μεγάλο πλήθος ακεραίων που δέχεται
- ◆ Σωστή ενημέρωση χρήστη
- ◆ Μη ακέραια τιμή
- ◆ Αρνητική τιμή
- ◆ Τιμές εκτός των ορίων

## Παρατηρήσεις:

Έλεγχος συνάρτησης *int ZoneSelect()* --> **ZoneSelect → Επιλογή ζώνης**

## Ισοδύναμες κλάσεις τιμών :

- ◆ < 0 (Άκυρες κλάσεις)
- ◆ > 4 (Άκυρες κλάσεις)
- ◆  $0 \leq \text{τιμή} \leq 4$  (Εγκυρες κλάσεις)

## Οριακές τιμές :

- ◆ 0.1, -0.1, 0.0 (Οριακές τιμές)
- ◆ 1.1, 0.9, 1.0 (Οριακές τιμές)
- ◆ 2.1, 2.0, 1.9 (Οριακές τιμές)
- ◆ 3.1, 3.0, 2.9 (Οριακές τιμές)
- ◆ 4.1, 4.0, 3.9 (Οριακές τιμές)

## Εικασίες τιμών :

- ◆ Καμία τιμή (Enter)
- ◆ Μεγάλη τιμή (πχ 10000)
- ◆ Αρνητική τιμή (πχ -254)
- ◆ Χαρακτήρας (πχ α, s, %)
- ◆ Κενό

<b>Ισοδύναμες κλάσεις τιμών (τιμές εισόδου)</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0(έγκυρη κλάση)	Επιστροφή στη MainMenu 0	Επιστροφή στη MainMenu 0	✓
1(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 1	Εμφάνιση αποτελεσμάτων 1	✓
2(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 2	Εμφάνιση αποτελεσμάτων 2	✓
3(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 3	Εμφάνιση αποτελεσμάτων 3	✓
4(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 4	Εμφάνιση αποτελεσμάτων 4	✓
-5(άκυρη κλάση)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	✗
5(άκυρη κλάση)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	✗
<b>Οριακές τιμές</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0	Επιστροφή στη MainMenu 0	Επιστροφή στη MainMenu 0	✓
0.1	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
- 0.1	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
1.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
1.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
0.9	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
2.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
2.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
1.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
2.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
4.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
4.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

Εικασίες τιμών	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
Καμία τιμή (Enter)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×
1000 (Μεγάλη τιμή)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	×
- 254 (Αρνητική τιμή)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	×
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
_ (κενο)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ δεν παίρνει ως τιμή χαρακτήρες
- ◆ δεν παίρνει ως τιμή δεκαδικούς αριθμούς
- ◆ δεν παίρνει ως τιμή αρνητικούς αριθμούς
- ◆ παίρνει τιμές από 0 έως 4 μόνο

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν αρκετά λάθη, από την εισαγωγή τιμών εκτός ορίων, μη επιτρεπόμενων τιμών (όπως για παράδειγμα χαρακτήρες, αρνητικοί αριθμοί, δεκαδικά ψηφία κλπ). Υπάρχουν όμως κάποιες παρατηρήσεις που η εφαρμογή τους είναι απαραίτητη σε κάποια σημεία της μεθόδου αυτά είναι:

- Κατά την εισαγωγή των τιμών απ' τον χρήστη, δεν υπάρχει η αντίστοιχη κλάση εξαιρέσεων για τον πλήρη έλεγχο των τιμών που εισάγονται.
- Είναι προτεινόμενο να δημιουργηθούν κάποιοι έλεγχοι για όλες αυτές τις περιπτώσεις σφαλμάτων που προκύπτουν.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2-λάθη 23)

**Έλεγχος συνάρτησης void Bleve(int zone, double &Rs, double &Ts, double &F, double &l, double &L) --> **Bleve**→Φαινόμενο Bleve**

**Έλεγχος συνάρτησης Ακτίνας Πύρινης Σφαίρας :**

Τύπος :  $R_S = 0.5 * 6.48 * M_R^{0.325}$  , όπου  $R_S$  η ακτίνα πύρινης σφάρας (m) και  $M_R$  η ολική ποσότητα εκροής (kgr)

Είσοδος : **Ολική ποσότητα εκροής  $M_R$**

Έξοδος : Ακτίνα Πύρινης σφαίρας  $R_S$

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : **Ολική ποσότητα εκροής  $M_R$**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1 (1500 m)	198,337		
2 (450 m)	198,337		
3 (170 m)	198,337		
4 (40 m)	198,337		

**Παρατηρήσεις.**

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- Εισαγωγή της Ακτίνας ζώνης
- Υπολογισμός Ακτίνας Πύρινης σφαίρας
- Εγκυρότητα του τύπου υπολογισμού της ακτίνας πύρινης σφαίρας

Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει το  $M_R$  απ' τον χρήστη, δημιουργήθηκε συνάρτηση με σταθερό το  $M_R = 314794$  και προκύπτει συνέχεια σταθερό αποτέλεσμα.

Οι προγραμματιστές θα πρέπει να δημιουργήσουν επιλογή στον χρήστη ώστε να εισάγει τιμές και να προκύπτουν διαφορετικά αποτελέσματα για κάθε εισαγωγή τιμών απ' τον χρήστη.

**ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

### **Έλεγχος συνάρτησης Χρονικής Διάρκειας Bleve :**

Τύπος :  $T_S = 0.852 * M_R^{0.26}$  , όπου  $T_S$  η χρονική διάρκεια BLEVE (sec) και  $M_R$  η ολική ποσότητα εκροής (kg)

Είσοδος : Ολική ποσότητα εκροής  $M_R$

Έξοδος : Χρονική διάρκεια Bleve

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Ολική ποσότητα εκροής  $M_R$

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1 (1500 m)	22,9048		
2 (450 m)	22,9048		
3 (170 m)	22,9048		
4 (40 m)	22,9048		

### **Παρατηρήσεις.**

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- Εισαγωγή της Ακτίνας ζώνης
- Υπολογισμός χρονικής διάρκειας Bleve
- Εγκυρότητα του τύπου υπολογισμού της χρονικής διάρκειας Bleve

Και συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει το  $M_R$  απ' τον χρήστη, δημιουργήθηκε συνάρτηση με σταθερό το  $M_R = 314794$  και προκύπτει συνέχεια σταθερό αποτέλεσμα.

Οι προγραμματιστές θα πρέπει να δημιουργήσουν επιλογή στον χρήστη ώστε να εισάγει τιμές στην μεταβλητή  $M_R$  και να προκύπτουν διαφορετικά αποτελέσματα για κάθε εισαγωγή τιμών απ' τον χρήστη.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

### **Έλεγχος συνάρτησης Ποσοστού Εκπεμπόμενης Ακτινοβολίας :**

Τύπος :  $F = 0.27 * P^{0.32}$  , όπου F το ποσοστό εκπεμπόμενης ακτινοβολίας (%) και P η πίεση της δεξαμενής (Μρα)

Είσοδος : Πίεση δεξαμενής P

Έξοδος : Ποσοστό Εκπεμπόμενης Ακτινοβολίας

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Πίεση δεξαμενής P

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1 (1500 m)	0,12923		
2 (450 m)	0,12923		
3 (170 m)	0,12923		
4 (40 m)	0,12923		

### **Παρατηρήσεις.**

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της Ακτίνας ζώνης
- ◆ Υπολογισμός ποσοστού εκπεμπόμενης ακτινοβολίας
- ◆ Εγκυρότητα του τύπου υπολογισμού του ποσοστού εκπεμπόμενης ακτινοβολίας

Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει η πίεση της δεξαμενής P απ' τον χρήστη, δημιουργήθηκε συνάρτηση με σταθερό το P και προκύπτει συνέχεια σταθερό αποτέλεσμα.

Οι προγραμματιστές θα πρέπει να δημιουργήσουν επιλογή στον χρήστη ώστε να εισάγει τιμές στην μεταβλητή P και να προκύπτουν διαφορετικά αποτελέσμα-τα για κάθε εισαγωγή τιμών απ' τον χρήστη.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

### Έλεγχος συνάρτησης Θερμικής Ακτινοβολίας :

Τύπος :  $I = \left( \frac{\zeta \omega \nu \eta}{T_s} \right)^{0,75}$  , όπου I η θερμική ακτινοβολία, ζώνη η ζώνη που δηλώνουμε

και  $T_s$  η χρονική διάρκεια BLEVE

Είσοδος : Ακτίνα Ζώνης και χρονική διάρκεια BLEVE  $T_s$

Εξοδος : Θερμική Ακτινοβολία I

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Ακτίνα Ζώνης και χρονική διάρκεια BLEVE  $T_s$

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1 (1500 m)	23,021		
2 (450 m)	9,33178		
3 (170 m)	4,49668		
4 (40 m)	1,51915		

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της Ακτίνας ζώνης
- ◆ Υπολογισμός της θερμικής ακτινοβολίας
- ◆ Εγκυρότητα του τύπου υπολογισμού της θερμικής ακτινοβολίας

Και εδώ στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι η συνάρτηση  $T_s$  που υπολογίζεται με τον πιο παραπάνω τύπο, παίρνει ως είσοδο την μεταβαλλόμενη τιμή  $M_R$  απ' τον χρήστη, ενώ εδώ δημιουργήθηκε συνάρτηση με σταθερό το  $M_R = 314794$  και προκύπτουν λανθασμένα αποτελέσματα.

Οι προγραμματιστές θα πρέπει να διορθώσουν την συνάρτηση  $T_s$ .

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

## Έλεγχος συνάρτησης Απόστασης Παρατηρητή :

$$\text{Τύπος : } L = \sqrt{\frac{F * M_R * E_B * T_A}{4 * \pi * I * 1000 * T_S}}, \text{ όπου } F \text{ το ποσοστό εκπεμπόμενης ακτινοβολίας,}$$

$M_R$  η ολική ποσότητα εκροής,  $E_B$  η ενέργεια καύσης,  $T_A$  ο ατμοσφαιρικός συντελεστής μετάδοσης  $\cong 0,7$ ,  $T_S$  η χρονική διάρκεια BLEVE,  $I$  η θερμική ακτινοβολία και  $L$  η απόσταση του παρατηρητή.

Είσοδος : θερμική ακτινοβολία  $I$

Έξοδος : Απόσταση παρατηρητή  $L$

Σταθερές παράμετροι :  $\pi$ ,  $E_B$  η ενέργεια καύσης,  $T_A$  ο ατμοσφαιρικός συντελεστής μετάδοσης

Μεταβλητές Παράμετροι :  $F$  το ποσοστό εκπεμπόμενης ακτινοβολίας,  $M_R$  η ολική ποσότητα εκροής,  $I$  η θερμική ακτινοβολία,  $T_S$  η χρονική διάρκεια BLEVE

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1 (1500 m)	439,765		
2 (450 m)	690,717		
3 (170 m)	995,03		
4 (40 m)	1711,91		

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- Εισαγωγή της Ακτίνας ζώνης
- Υπολογισμός της απόστασης του παρατηρητή
- Εγκυρότητα του τύπου υπολογισμού της απόστασης του παρατηρητή

Και σ' αυτή τη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει το  $M_R$  απ' τον χρήστη και το  $P$ , ο υπολογισμός του  $T_S$  και του  $P$ , δεν είναι σωστός.

Οι προγραμματιστές θα πρέπει να δίνουν την δυνατότητα στον χρήστη ώστε να εισάγει τιμές στην μεταβλητή  $M_R$  και  $P$  και να διορθώσουν τα  $F$  και  $T_S$ .

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

## Συνάρτηση FlameJet και Default:

Στην συνέχεια, απ' την επιλογή του MainMenu, πληκτρολογώντας 2 για το φαινόμενο FlameJet, ο χρήστης, καλεί στην ουσία την συνάρτηση *void Default()* όπου η συνάρτηση αυτή καλεί την *double FlameJet(double FF)*, όπου η συνάρτηση *FlameJet*, υπολογίζει από προκαθορισμένο τύπο το μήκος της φλόγας για διαρροή λόγω διάβρωσης. Εδώ ο χρήστης έχει να εισάγει το ρυθμό διαρροής. Πληκτρολογώντας και τον επιθυμητό ρυθμό διαρροής, το λογισμικό πρέπει να του εμφανίσει το μήκος της φλόγας που θα προκύψει. Στην ουσία λοιπόν, θα ελέγξουμε την εισαγωγή τιμών μέσω της συνάρτησης *Default* για την εισαγωγή του ρυθμού διάβρωσης και την εγκυρότητα των τιμών που επιστρέφει η συνάρτηση *FlameJet*. Έτσι λοιπόν, οι τιμές που θα εισάγονται δεν υπάρχει περίπτωση να δημιουργήσουν εικασίες λανθασμένων τιμών όπως για παράδειγμα:

- ◆ Καμία τιμή (Enter)
- ◆ Γράμματα ή άλλοι χαρακτήρες (π.χ. @,#, ",δ,a,?,B )
- ◆ Κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Μεγάλο πλήθος ακεραίων που δέχεται
- ◆ Σωστή ενημέρωση χρήστη
- ◆ Αρνητική τιμή
- ◆ Τιμές εκτός των ορίων

## Παρατηρήσεις:

Έλεγχος συνάρτησης `int Default()`, `case(2)` → **Επιλογή ρυθμού διαρροής**

## Ισοδύναμες κλάσεις τιμών :

- ◆  $< 0,17$  (Άκυρες κλάσεις)
- ◆  $> 1,30$  (Άκυρες κλάσεις)
- ◆  $0,17 \leq \text{τιμή} \leq 1,30$  (Έγκυρες κλάσεις)

## Οριακές τιμές :

- ◆ 0.17, 0.16, 0.18 (Οριακές τιμές)
- ◆ 1.30, 1.29, 1.31 (Οριακές τιμές)
- ◆ 1.00, 0.99, 1.01 (Οριακές τιμές)
- ◆ 1.01, 1.00, 1.02 (Οριακές τιμές)

## Εικασίες τιμών :

- ◆ Καμία τιμή (Enter)
- ◆ Μεγάλη τιμή (πχ 10000)
- ◆ Αρνητική τιμή (πχ -254)
- ◆ Μηδενική τιμή (0)
- ◆ Χαρακτήρας (πχ α, s, %)
- ◆ Κενό
- ◆ Τυχαίες τιμές μεταξύ των ορίων

<b>Ισοδύναμες κλάσεις τιμών (τιμές εισόδου)</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0.17(έγκυρη κλάση)	Επιστροφή στη MainMenu	Εμφάνιση αποτελεσμάτων	✗
1.30(έγκυρη κλάση)	Επιστροφή στη MainMenu	Εμφάνιση αποτελεσμάτων	✗
-5(άκυρη κλάση)	Επιστροφή στη MainMenu	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
5(άκυρη κλάση)	Επιστροφή στη MainMenu	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
<b>Οριακές τιμές</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0.17	Επιστροφή στη MainMenu	Εμφάνιση αποτελεσμάτων	✗
0.16	Επιστροφή στη MainMenu	Εμφάνιση μηνύματος λάθους	✗
0.18	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.30	Επιστροφή στη MainMenu	Εμφάνιση αποτελεσμάτων	✗
1.29	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.31	Επιστροφή στη MainMenu	Εμφάνιση μηνύματος λάθους	✗
1.00	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
0.99	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.01	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.01	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.00	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.02	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
<b>Εικασίες τιμών</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
Καμία τιμή (Enter)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	✗
1000 (Μεγάλη τιμή)	Επιστροφή στη MainMenu	Εμφάνιση μηνύματος λάθους & επιστροφή στην MainMenu	✗
- 254 (Αρνητική τιμή)	Επιστροφή στη MainMenu	Εμφάνιση μηνύματος λάθους & επιστροφή στην MainMenu	✗
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
_ (κενο)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	✗
0,53 (Τυχαίες τιμές)	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1,19 (Τυχαίες τιμές)	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ δεν παίρνει ως τιμή χαρακτήρες
- ◆ δεν παίρνει ως τιμή δεκαδικούς αριθμούς με περισσότερα από 2 δεκαδικά ψηφία
- ◆ δεν παίρνει ως τιμή αρνητικούς αριθμούς
- ◆ παίρνει τιμές από 0,17 έως 1,30 μόνο

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν κάποια λάθη, από την εισαγωγή τιμών εκτός ορίων, μη επιτρεπόμενων τιμών (όπως για παράδειγμα χαρακτήρες, αρνητικοί αριθμοί, δεκαδικά ψηφία κλπ). Υπάρχουν όμως κάποιες παρατηρήσεις που η εφαρμογή τους είναι απαραίτητη σε κάποια σημεία της μεθόδου αυτά είναι:

- Κατά την εισαγωγή των τιμών απ' τον χρήστη, δεν υπάρχει η αντίστοιχη κλάση εξαιρέσεων για τον πλήρη έλεγχο των τιμών που εισάγονται.
- Είναι προτεινόμενο να δημιουργηθούν κάποιοι έλεγχοι για όλες αυτές τις περιπτώσεις σφαλμάτων που προκύπτουν.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2-λάθη 15)

## Έλεγχος συνάρτησης double FlameJet(double FF)-->FlameJet→Φαινόμενο FlameJet

### Έλεγχος συνάρτησης Μήκους φλόγας για διαρροή :

Τύπος :  $F_F = 0,0006785 * L_F^{2,5}$  , όπου  $F_F$  η ροή του καυσίμου και  $L_F$  το μήκος της

$$\text{φλόγας, οπότε είναι } L_F = \sqrt[2,5]{\frac{F_F}{0,0006785}}$$

Είσοδος :  $F_F$  η ροή του καυσίμου

Εξοδος : Μήκος φλόγας για διαρροή  $L_F$

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Ρυθμός Διαρροής  $F_F$

Είσοδος	Εξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1,19	19,8426	19,8426	✓
0,59	14,9872	14,9872	✓
0,18	9,32157	9,32157	✓
1,29	20,4935	20,4935	✓
1,00	18,5089	18,5089	✓
1,01	18,5827	18,5827	✓

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή του ρυθμού διαρροής
- ◆ Υπολογισμός του μήκους της πόρινης φλόγας
- ◆ Εγκυρότητα του τύπου υπολογισμού του μήκους της πόρινης φλόγας

Ελέγχθηκαν οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο δεν προέκυψαν κάποια λάθη, από την εισαγωγή τιμών, ώστε να οφείλονται σε λανθασμένο υπολογισμό του συγκεκριμένου τύπου.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 0 – λάθη 0)

Α' Έλεγχος  
Ανοιχτού Κουτιού



Ελεγκτής Α' : Υφαντής Γεώργιος  
Ελεγκτής Β' : Αναστασιάδου Έλενα

Ανάπτυξη Λογισμικού για την Βιομηχανία Πετρελαίου Καναλα Oil

**ΗΜΕΡΟΛΟΓΙΟ ΕΛΕΓΚΤΗ: 10 Μαΐου 2004**

**Ver 1.0**

**Ελέγχθηκαν τα αρχεία :**

● **bleve.cpp**

**και οι συναρτήσεις :**

- **Bleve**
- **PoolFire**
- **FlameJet**
- **MainMenu**
- **ZoneSelect**
- **Default**

Θα τις εξετάσουμε λοιπόν μια προς μια:

## 1. Έλεγχος της συνάρτησης Blevé(int zone, double &Rs, double &Ts, double &F, double &I, double &L)

```
void Blevé(int zone, double &Rs, double &Ts, double &F, double &I, double &L)
{
    double t1,t2,t3;

    t3= pow(314794,0.325);
    Rs=((0.5*6.48)*t3);

    Ts=k2*(pow(Mr,0.26));

    F=k3*(pow(P,0.32));

    I=pow((zone/Ts),0.75);

    t1= F*Mr*Eb*Ta;
    t2=4*M_PI*I*1000*Ts;
    L=sqrt((t1/t2));
}
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στον κώδικα αυτό δεν έχουμε κάποια από τις παραπάνω περιπτώσεις, παρά μόνο τύπους οι οποίοι ελέγχθηκαν ένας προς ένας και βρέθηκαν τα εξής λάθη: Οι παρακάτω τύποι δεν έχουν μεταβλητό μέρος το οποίο θα πρέπει να δίνετε από το χρήστη:

```
t3= pow(314794,0.325)
Rs=((0.5*6.48)*t3)
Ts=k2*(pow(Mr,0.26))
F=k3*(pow(P,0.32))
```

**ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

## 2. Έλεγχος της συνάρτησης PoolFire0

```
void PoolFire()
{
    double CL,DA,DL,EP,G,HF,HL,I,K1,K2,K3,MBR,MR,Q,RP,X,degrees;
    double pi=3.14;
    double t1,t2,t3;
    HL=0.05;
    K1=42;
    K2=5.1*pow(10,-5);
    K3=2.5*pow(10,-5);
    MBR=0.05;
    DA=600;
    CL=45000;

    RP=sqrt(MR/(pi*HL*DL));
    cout<<" RP :"<<RP<<endl;

    t1= sqrt(2*RP*G);
    t2= 2*RP*K1;
    t3= MBR/(DA*t1);
    HF=t2*(pow(t3 , 0.6 ));
    cout<<" HF :"<<HF<<endl;

    Q=K2*EP*DL*CL;
    cout<<"Q :"<<Q<<endl;

    EP=pi*pow(RP,2);
    cout<<" EP :"<<EP<<endl;

    I=K3*((EP*DL*CL)/(pi*RP*HF*EP));
    cout<<" I :"<<I<<endl;
    X=sqrt( (Q*cos(degrees))/(4*pi*I));
    cout<<"X :"<<X<<endl;
    //return RP;
}
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στον κώδικα αυτό δεν έχουμε κάποια από τις παραπάνω περιπτώσεις, παρά μόνο τύπους οι οποίοι ελέγχθηκαν ένας προς ένας και δεν παρατηρήθηκε κανένα λάθος.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

### 3. Έλεγχος της συνάρτησης FlameJet(double FF)

```
double FlameJet(double FF)
{
    double LF;
    LF=pow((FF/0.0006785),(1/2.5));

    return LF;
}
```

Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

Παρατηρήσεις:

Στον κώδικα αυτό δεν έχουμε κάποια από τις παραπάνω περιπτώσεις, παρά μόνο ένας τύπος ο οποίος ελέγχθηκε και δεν παρατηρήθηκε κανένα λάθος.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

#### 4. Έλεγχος της συνάρτησης MainMenu()

```
int MainMenu()
{
    int choice;
    do{
        cout<<"\nEPELEXE FAINOMRNO"<<endl;
        cout<<"1.GIA FAINOMRNO BLEVE"<<endl;
        cout<<"2.GIA FAINOMRNO FLAME JET / FLARE"<<endl;
        cout<<"3.GIA FAINOMRNO POOL FIRE"<<endl;
        cout<<"0.EXIT"<<endl;
        cout<<"DOSE EPILOGI 0..3 : ";
        cin>>choice;
    } while (choice<0 || choice>3);
    return choice;
}
```

#### Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

#### Παρατηρήσεις:

Στην παραπάνω συνάρτηση έχουμε:

- Μια συνθήκη **do while** η οποία δίνει τη δυνατότητα στο χρήστη να επιλέξει μια από της 4 επιλογές. Η συνθήκη υλοποιήθηκε σωστά και δεν υπάρχει πρόβλημα στην εκτέλεση της
- Όμως δεν ελέγχονται όλες οι περιπτώσεις και δεν δίνονται τα κατάλληλα μηνύματα  
Όπως όταν ο χρήστης δώσει οτιδήποτε άλλο εκτός από αριθμό( γράμμα ή σύμβολο).

#### **Αντιστοιχος κώδικας :**

```
do{
    cout<<"\nEPELEXE FAINOMRNO"<<endl;
    cout<<"1.GIA FAINOMRNO BLEVE"<<endl;
    cout<<"2.GIA FAINOMRNO FLAME JET / FLARE"<<endl;
    cout<<"3.GIA FAINOMRNO POOL FIRE"<<endl;
    cout<<"0.EXIT"<<endl;
    cout<<"DOSE EPILOGI 0..3 : ";
    cin>>choice;
} while (choice<0 || choice>3);
```

## 5. Έλεγχος της συνάρτησης ZoneSelect()

```
int ZoneSelect()
{
  int choice;
  do{
    cout<<"\nEPELEXE ZWNH "<<endl;
    cout<<"1.ZWNH I (1500)"<<endl;
    cout<<"2.ZWNH II (450)"<<endl;
    cout<<"3.ZWNH III (170)"<<endl;
    cout<<"4.ZWNH IV (40)"<<endl;
    cout<<"0.EXIT"<<endl;
    cout<<"DOSE EPILOGH (0..4) : ";
    cin>>choice;
  } while(choice<0 || choice>4);

  switch(choice)
  {
    case(0):{ return 0; }
    case(1):{ return 1500;}
    case(2):{ return 450;}
    case(3):{ return 170;}
    case(4):{ return 40;}
  }
  //return;
}
```

### Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στην παραπάνω συνάρτηση έχουμε:

- Μια συνθήκη **do while με την switch(choice)** η οποία δίνει τη δυνατότητα στο χρήστη να επιλέξει μια από της 5 επιλογές. Η συνθήκη υλοποιήθηκε σωστά και δεν υπάρχει πρόβλημα στην εκτέλεση της
- Όμως δεν ελέγχονται όλες οι περιπτώσεις και δεν δίνονται τα κατάλληλα μηνύματα  
Όπως όταν ο χρήστης δώσει οτιδήποτε άλλο εκτός από αριθμό( γράμμα ή σύμβολο).

### Αντιστοιχος κώδικας :

```
do{
    cout<<"\nEPELEXE ZWNH "<<endl;
    cout<<"1.ZWNH I (1500)"<<endl;
    cout<<"2.ZWNH II (450)"<<endl;
    cout<<"3.ZWNH III (170)"<<endl;
    cout<<"4.ZWNH IV (40)"<<endl;
    cout<<"0.EXIT"<<endl;
    cout<<"DOSE EPILOGH (0.4) : ";
    cin>>choice;
} while(choice<0 || choice>4);
```

### ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ

### 6. Έλεγχος της συνάρτησης Default()

```
void Default(){
    int zone;
    double Rs,Ts,F,I,L;
    double FF,LF;
    int choice;

    do{

        choice=MainMenu();

        switch(choice)
        {
            case(1){
                zone=ZoneSelect();
                if (zone!=0)
                {
                    Bleve(zone,Rs,Ts,F,I,L);
                    cout<<"\n FAINOMENO BLEVE "<<endl;
                    cout<<"AKTINA PURINHS SFAIRAS Rs (m) : "<<Rs<<endl;
                    cout<<"XRONIKI DIARKEIA BLEVE Ts (sec) : "<<Ts<<endl;
                    cout<<"POSOSTO EKPEMPOMENHS AKTINOVOLIAS F (%) : "<<F<<endl;
                    cout<<"THERMIKI AKTINOVOLIA I (W/m2) : "<<I<<endl;
                    cout<<"APOSTASH PARATHRHRH L (m) : "<<L<<endl<<endl<<endl;
                }
                break;
            }
        }
    }
}
```

```

    }
    case(2):{
        cout<<"\n          FAINOMRNO  FLAME  JET  /
FLARE"<<endl;
        cout<<"DIAROH LOGO DIABROSHS    O,17 - 1,00 kgr/sec"<<endl;
        cout<<"DIATRSHSH  H  MHXANIKH  ASTOXIA    1,01 - 1,30
kgr/sec"<<endl;
        cout<<" DWSE RYTHMO DIAROHW SE (kgr/sec): ";
        cin>>FF;
        if(FF>0.17 && FF<=1.0)
        {
            LF=FlameJet(FF);
            cout<<endl;
            cout<<"\n  MHKOS  FLOGAS  (m)  GIA  DIAROH  LOGO
DIABROSHS:"<<LF<<endl;
        }
        else if(FF>=1.01 && FF<1.30)
        {
            LF=FlameJet(FF);
            cout<<"\n  MHKOS  FLOGAS  (m)  GIA  DIATRSHSH  H
MHXANIKH  ASTOXIA:"<<LF<<endl;
        }
        // else throw

        break;
    }
    case(3):{
        //PoolFire();
        break;
    }

} //switch

} while(choice!=0);

}

```

### Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

## Παρατηρήσεις:

Στον συνάρτηση αυτή έχουμε:

- Μια συνθήκη **do while** η οποία με την **switch(choice)** ελέγχει τις 3 περιπτώσεις
  - Έχουμε μια if που ελέγχει αν η ζώνη είναι '0' για να κάνει exit ενώ αν δεν είναι με την συνάρτηση **bleve** υπολογίζει τα αποτελέσματα
  - Έχουμε μια if else που ελέγχει τα όρια της διαρροής που δίνει ο χρήστης.
  -

Τα οποία όρια όμως δεν ελεγχοντε σωστα :

- Για 0.17 false(εμφάνιση MainMenu)
- Για 0.30 false(εμφάνιση MainMenu)
- Για >0.17 false(εμφάνιση MainMenu)
- Για <1.31 false(εμφάνιση MainMenu)
- Οτιδήποτε άλλο εκτός από αριθμό( γράμμα ή σύμβολο)false.

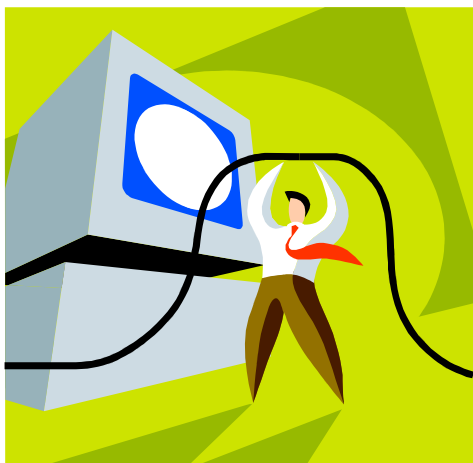
**Η 3<sup>η</sup> περίπτωση** δεν έχει ελεγχθεί.

## **Αντίστοιχος κώδικας :**

```
case(2):{
    cout<<"\n  FAINOMRNO FLAME JET / FLARE"<<endl;
    cout<<"DIAROH LOGO DIABROSHS    0,17 - 1,00 kgr/sec"<<endl;
    cout<<"DIATRSHSH H MHXANIKH ASTOXIA 1,01 - 1,30  kgr/sec"<<endl;
    cout<<" DWSE RYTHMO DIAROHW SE (kgr/sec): ";
    cin>>FF;
    if(FF>0.17 && FF<=1.0)
    {
        LF=FlameJet(FF);
        cout<<endl;
        cout<<"\n ΜΗΚΟΣ ΦΛΟΓΑΣ (m) GIA DIAROH LOGO
            DIABROSHS:"<<LF<<endl;
    }
    else if(FF>=1.01 && FF<1.30)
    {
        LF=FlameJet(FF);
        cout<<"\n ΜΗΚΟΣ ΦΛΟΓΑΣ (m) GIA  DIATRSHSH H MHXANIKH
            ASTOXIA:"<<LF<<endl;
    }
    // else throw
    break;
}
```

# Β' Έλεγχος

## Κλειστού Κουτιού



Ελεγκτής Α' : Υφαντής Γεώργιος  
Ελεγκτής Β' : Αναστασιάδου Έλενα

## ΗΜΕΡΟΛΟΓΙΟ ΕΛΕΓΚΤΗ: 16 Μαΐου 2004

Ver 1.0b

### Ελέγχθηκαν τα αρχεία :

- Blev2.cpp
- 16\_running.cpp

### και οι συναρτήσεις :

- Blev
- FlameJet
- MainMenu(a)
- MainMenu(b)
- ZoneSelect
- Default
- changeAdmin
- pass
- ReadFromFile
- ReadFromFile
- PoolFire
- Class BXrtn (κλάση εξαιρέσεων)



### Στο αρχείο bleve2.cpp

### Συνάρτηση MainMenu(a), MainMenu(b) :

Στην συνάρτηση MainMenu(a) είναι το μενού επιλογών των administrators και MainMenu(b) είναι το μενού των απλών χρηστών. Οι τιμές που θα δοθούν στον αναλυτή για την εξέταση ορίζονται από τον χρήστη του λογισμικού και είναι αυστηρά καθορισμένες απ' τις επιλογές που καθορίζονται απ' το λογισμικό. Έτσι λοιπόν, οι τιμές που θα εισάγονται δεν υπάρχει περίπτωση να δημιουργήσουν εικασίες λανθασμένων τιμών όπως για παράδειγμα:

- Καμία τιμή (Enter)
- Γράμματα ή άλλοι χαρακτήρες (π.χ. @, #, ", δ, a, ?, B )
- Κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- Μεγάλο πλήθος ακεραίων που δέχεται
- Σωστή ενημέρωση χρήστη
- Μη ακέραια τιμή
- Αρνητική τιμή
- Τιμές εκτός των ορίων

Εδώ θα ελέγξουμε με τις επιλογές που έχουμε ως απλοί χρήστες <MainMenu(a)> όπως κάναμε και στον 1<sup>ο</sup> έλεγχο όπου έχουμε να εισάγουμε τιμές από 0 έως 3 και για την περίπτωση των administrator <MainMenu(a)> όπου έχουμε άλλες 2 επιλογές για αλλαγή καταχώρησης στοιχείων και αλλαγή κωδικών πρόσβασης των administrators, θα ελέγξουμε παρακάτω με τις αντίστοιχες συναρτήσεις τους.

## Παρατηρήσεις:

Έλεγχος συνάρτησης int MainMenu() --> **MainMenu(a) → Επιλογή φαινομένου**

### Ισοδύναμες κλάσεις τιμών :

- ◆ < 0 (Άκυρες κλάσεις)
- ◆ > 3 (Άκυρες κλάσεις)
- ◆  $0 \leq \text{τιμή} \leq 3$  (Έγκυρες κλάσεις)

### Οριακές τιμές :

- ◆ 0.1, -0.1, 0.0 (Οριακές τιμές)
- ◆ 1.1, 0.9, 1.0 (Οριακές τιμές)
- ◆ 2.1, 2.0, 1.9 (Οριακές τιμές)
- ◆ 3.1, 3.0, 2.9 (Οριακές τιμές)

### Εικασίες τιμών :

- ◆ Καμία τιμή (Enter)
- ◆ Μεγάλη τιμή (πχ 10000)
- ◆ Αρνητική τιμή (πχ -254)
- ◆ Χαρακτήρας (πχ α, s, %)
- ◆ Κενό

Ισοδύναμες κλάσεις τιμών (τιμές εισόδου)	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
0 (έγκυρη κλάση)	Έξοδος 0	Έξοδος 0	✓
1(έγκυρη κλάση)	ZoneSelect 1	ZoneSelect 1	✓
2(έγκυρη κλάση)	FlameJet 2	FlameJet 2	✓
3(έγκυρη κλάση)	PoolFire 3	PoolFire 3	✓
-5(άκυρη κλάση)	MainMenu	Εμφάνιση μηνύματος λάθους & επιστροφή στην MainMenu	✗
5(άκυρη κλάση)	MainMenu	Εμφάνιση μηνύματος λάθους & επιστροφή στην MainMenu	✗
Οριακές τιμές	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
0.0	Έξοδος 0	Έξοδος 0	✓
0.1	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
- 0.1	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
1.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
1.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
0.9	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
2.0	Εμφάνιση λάθος αποτελεσμάτων και επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	✗
2.1	Εμφάνιση λάθος αποτελεσμάτων και επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	✗
1.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.0	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
3.1	Εμφάνιση λάθος αποτελεσμάτων και επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	✗
2.9	Εμφάνιση λάθος αποτελεσμάτων και επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	✗

Εικασίες τιμών	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
Καμία τιμή (Enter)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×
1000 (Μεγάλη τιμή)	Επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	×
- 254 (Αρνητική τιμή)	Επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	×
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
_ (κενο)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ δεν παίρνει ως τιμή χαρακτήρες
- ◆ δεν παίρνει ως τιμή δεκαδικούς αριθμούς
- ◆ δεν παίρνει ως τιμή αρνητικούς αριθμούς
- ◆ παίρνει τιμές από 0 έως 3 μόνο

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν αρκετά λάθη, από την εισαγωγή τιμών εκτός ορίων, μη επιτρεπόμενων τιμών (όπως για παράδειγμα χαρακτήρες, αρνητικοί αριθμοί, δεκαδικά ψηφία κλπ). Υπάρχουν όμως κάποιες παρατηρήσεις που η εφαρμογή τους είναι απαραίτητη σε κάποια σημεία της μεθόδου αυτά είναι:

- Κατά την εισαγωγή των τιμών απ' τον χρήστη, ενώ υπάρχει κλάση εξαιρέσεων για τον πλήρη έλεγχο των τιμών που εισάγονται, δεν εκτελείται ποτε.
- Είναι προτεινόμενο να δημιουργηθούν κάποιοι έλεγχοι για όλες αυτές τις περιπτώσεις σφαλμάτων που προκύπτουν και να συμπεριληφθούν στην κλάση εξαιρέσεων.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 3-λάθη 21)

## Συνάρτηση Bleve, ZoneSelect και Default:

Στην συνέχεια, απ' την επιλογή του MainMenu, πληκτρολογώντας 1 για το φαινόμενο Bleve, ο χρήστης, καλεί στην ουσία την συνάρτηση *void Default()* όπου η συνάρτηση αυτή καλεί και τις άλλες δύο την *void Bleve(int zone, double &Rs, double &Ts, double &F, double &I, double &L)* και την *int ZoneSelect()*, όπου η συνάρτηση Bleve, υπολογίζει από προκαθορισμένους τύπους α) την ακτίνα πύρινης σφαίρας, β) την χρονική διάρκεια του φαινομένου, γ) το ποσοστό εκπεμπόμενης ακτινοβολίας, δ) την θερμική ακτινοβολία και ε) την απόσταση του παρατηρητή. Εδώ ο χρήστης έχει να εισάγει ποια απ' τις προτεινόμενες τέσσερις ζώνες επιθυμεί να ελέγξει. Πληκτρολογώντας και την επιθυμητή ζώνη, το λογισμικό πρέπει να του εμφανίσει κάποια στοιχεία τα οποία θα ελεγχθούν παρακάτω. Στην ουσία λοιπόν, θα ελέγξουμε την εισαγωγή τιμών μέσω της συνάρτησης *ZoneSelect* και την εγκυρότητα των τιμών που επιστρέφει η συνάρτηση *Bleve*. Έτσι λοιπόν, οι τιμές που θα εισάγονται δεν υπάρχει περίπτωση να δημιουργήσουν εικασίες λανθασμένων τιμών όπως για παράδειγμα:

- ◆ Καμία τιμή (Enter)
- ◆ Γράμματα ή άλλοι χαρακτήρες (π.χ. @,#,"δ,a,?,B )
- ◆ Κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Μεγάλο πλήθος ακεραίων που δέχεται
- ◆ Σωστή ενημέρωση χρήστη
- ◆ Μη ακέραια τιμή
- ◆ Αρνητική τιμή
- ◆ Τιμές εκτός των ορίων

## Παρατηρήσεις:

Έλεγχος συνάρτησης *int ZoneSelect()* --> **ZoneSelect → Επιλογή ζώνης**

## Ισοδύναμες κλάσεις τιμών :

- ◆ < 0 (Άκυρες κλάσεις)
- ◆ > 4 (Άκυρες κλάσεις)
- ◆  $0 \leq \text{τιμή} \leq 4$  (Έγκυρες κλάσεις)

## Οριακές τιμές :

- ◆ 0.1, -0.1, 0.0 (Οριακές τιμές)
- ◆ 1.1, 0.9, 1.0 (Οριακές τιμές)
- ◆ 2.1, 2.0, 1.9 (Οριακές τιμές)
- ◆ 3.1, 3.0, 2.9 (Οριακές τιμές)
- ◆ 4.1, 4.0, 3.9 (Οριακές τιμές)

## Εικασίες τιμών :

- ◆ Καμία τιμή (Enter)
- ◆ Μεγάλη τιμή (πχ 10000)
- ◆ Αρνητική τιμή (πχ -254)
- ◆ Χαρακτήρας (πχ α, s, %)
- ◆ Κενό

<b>Ισοδύναμες κλάσεις τιμών (τιμές εισόδου)</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0(έγκυρη κλάση)	Επιστροφή στη MainMenu 0	Επιστροφή στη MainMenu 0	✓
1(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 1	Εμφάνιση αποτελεσμάτων 1	✓
2(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 2	Εμφάνιση αποτελεσμάτων 2	✓
3(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 3	Εμφάνιση αποτελεσμάτων 3	✓
4(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 4	Εμφάνιση αποτελεσμάτων 4	✓
-5(άκυρη κλάση)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	✗
5(άκυρη κλάση)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	✗
<b>Οριακές τιμές</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0	Επιστροφή στη MainMenu 0	Επιστροφή στη MainMenu 0	✓
0.1	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
- 0.1	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
1.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
1.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
0.9	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
2.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
2.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
1.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
2.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
4.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
4.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

Εικασίες τιμών	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
Καμία τιμή (Enter)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×
1000 (Μεγάλη τιμή)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	×
- 254 (Αρνητική τιμή)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	×
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
_ (κενο)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- δεν παίρνει ως τιμή χαρακτήρες
- δεν παίρνει ως τιμή δεκαδικούς αριθμούς
- δεν παίρνει ως τιμή αρνητικούς αριθμούς
- παίρνει τιμές από 0 έως 4 μόνο

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν αρκετά λάθη, από την εισαγωγή τιμών εκτός ορίων, μη επιτρεπόμενων τιμών (όπως για παράδειγμα χαρακτήρες, αρνητικοί αριθμοί, δεκαδικά ψηφία κλπ). Υπάρχουν όμως κάποιες παρατηρήσεις που η εφαρμογή τους είναι απαραίτητη σε κάποια σημεία της μεθόδου αυτά είναι:

- Κατά την εισαγωγή των τιμών απ' τον χρήστη, δεν υπάρχει η αντίστοιχη κλάση εξαιρέσεων για τον πλήρη έλεγχο των τιμών που εισάγονται.
- Είναι προτεινόμενο να δημιουργηθούν κάποιοι έλεγχοι για όλες αυτές τις περιπτώσεις σφαλμάτων που προκύπτουν.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2-λάθη 24)

**Έλεγχος συνάρτησης void Bleve(int zone, double &Rs, double &Ts, double &F, double &I, double &L) --> **Bleve**→Φαινόμενο Bleve**

**Έλεγχος συνάρτησης Ακτίνας Πύρινης Σφαίρας :**

Τύπος :  $R_s = 0.5 * 6.48 * M_R^{0.325}$  , όπου  $R_s$  η ακτίνα πύρινης σφαίρας (m) και  $M_R$  η ολική ποσότητα εκροής (kgr)

Είσοδος : **Ολική ποσότητα εκροής  $M_R$**

Εξοδος : Ακτίνα Πύρινης σφαίρας  $R_s$

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : **Ολική ποσότητα εκροής  $M_R$**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1 (1500 m)	198,337		
2 (450 m)	198,337		
3 (170 m)	198,337		
4 (40 m)	198,337		

**Παρατηρήσεις.**

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της Ακτίνας ζώνης
- ◆ Υπολογισμός Ακτίνας Πύρινης σφαίρας
- ◆ Εγκυρότητα του τύπου υπολογισμού της ακτίνας πύρινης σφαίρας

Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει το  $M_R$  απ' τον χρήστη, δημιουργήθηκε συνάρτηση με σταθερό το  $M_R = 314794$  και προκύπτει συνέχεια σταθερό αποτέλεσμα.

Οι προγραμματιστές θα πρέπει να δημιουργήσουν επιλογή στον χρήστη ώστε να εισάγει τιμές και να προκύπτουν διαφορετικά αποτελέσματα για κάθε εισαγωγή τιμών απ' τον χρήστη.

**ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

### Έλεγχος συνάρτησης Χρονικής Διάρκειας Bleve :

Τύπος :  $T_S = 0.852 * M_R^{0.26}$  , όπου  $T_S$  η χρονική διάρκεια BLEVE (sec) και  $M_R$  η ολική ποσότητα εκροής (kg)

Είσοδος : Ολική ποσότητα εκροής  $M_R$

Έξοδος : Χρονική διάρκεια Bleve

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Ολική ποσότητα εκροής  $M_R$

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1 (1500 m)	22,9048		
2 (450 m)	22,9048		
3 (170 m)	22,9048		
4 (40 m)	22,9048		

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της Ακτίνας ζώνης
- ◆ Υπολογισμός χρονικής διάρκειας Bleve
- ◆ Εγκυρότητα του τύπου υπολογισμού της χρονικής διάρκειας Bleve

Και συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει το  $M_R$  απ' τον χρήστη, δημιουργήθηκε συνάρτηση με σταθερό το  $M_R = 314794$  και προκύπτει συνέχεια σταθερό αποτέλεσμα.

Οι προγραμματιστές θα πρέπει να δημιουργήσουν επιλογή στον χρήστη ώστε να εισάγει τιμές στην μεταβλητή  $M_R$  και να προκύπτουν διαφορετικά αποτελέσματα για κάθε εισαγωγή τιμών απ' τον χρήστη.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

### **Έλεγχος συνάρτησης Ποσοστού Εκπεμπόμενης Ακτινοβολίας :**

Τύπος :  $F = 0.27 * P^{0.32}$  , όπου F το ποσοστό εκπεμπόμενης ακτινοβολίας (%) και P η πίεση της δεξαμενής (Μρα)

Είσοδος : Πίεση δεξαμενής P

Έξοδος : Ποσοστό Εκπεμπόμενης Ακτινοβολίας

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Πίεση δεξαμενής P

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1 (1500 m)	0,12923		
2 (450 m)	0,12923		
3 (170 m)	0,12923		
4 (40 m)	0,12923		

### **Παρατηρήσεις.**

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της Ακτίνας ζώνης
- ◆ Υπολογισμός ποσοστού εκπεμπόμενης ακτινοβολίας
- ◆ Εγκυρότητα του τύπου υπολογισμού του ποσοστού εκπεμπόμενης ακτινοβολίας

Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει η πίεση της δεξαμενής P απ' τον χρήστη, δημιουργήθηκε συνάρτηση με σταθερό το P και προκύπτει συνέχεια σταθερό αποτέλεσμα.

Οι προγραμματιστές θα πρέπει να δημιουργήσουν επιλογή στον χρήστη ώστε να εισάγει τιμές στην μεταβλητή P και να προκύπτουν διαφορετικά αποτελέσμα-τα για κάθε εισαγωγή τιμών απ' τον χρήστη.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

### Έλεγχος συνάρτησης Θερμικής Ακτινοβολίας :

Τύπος :  $I = \left( \frac{\zeta \omega \nu \eta}{T_s} \right)^{0,75}$  , όπου I η θερμική ακτινοβολία, ζώνη η ζώνη που δηλώνουμε

και  $T_s$  η χρονική διάρκεια BLEVE

Είσοδος : Ακτίνα Ζώνης και χρονική διάρκεια BLEVE  $T_s$

Εξοδος : Θερμική Ακτινοβολία I

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Ακτίνα Ζώνης και χρονική διάρκεια BLEVE  $T_s$

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1 (1500 m)	23,021		
2 (450 m)	9,33178		
3 (170 m)	4,49668		
4 (40 m)	1,51915		

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της Ακτίνας ζώνης
- ◆ Υπολογισμός της θερμικής ακτινοβολίας
- ◆ Εγκυρότητα του τύπου υπολογισμού της θερμικής ακτινοβολίας

Και εδώ στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι η συνάρτηση  $T_s$  που υπολογίζεται με τον πιο παραπάνω τύπο, παίρνει ως είσοδο την μεταβαλλόμενη τιμή  $M_R$  απ' τον χρήστη, ενώ εδώ δημιουργήθηκε συνάρτηση με σταθερό το  $M_R = 314794$  και προκύπτουν λανθασμένα αποτελέσματα.

Οι προγραμματιστές θα πρέπει να διορθώσουν την συνάρτηση  $T_s$ .

**ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

### Έλεγχος συνάρτησης Απόστασης Παρατηρητή :

$$\text{Τύπος : } L = \sqrt{\frac{F * M_R * E_B * T_A}{4 * \pi * I * 1000 * T_S}}, \text{ όπου } F \text{ το ποσοστό εκπεμπόμενης ακτινοβολίας,}$$

$M_R$  η ολική ποσότητα εκροής,  $E_B$  η ενέργεια καύσης,  $T_A$  ο ατμοσφαιρικός συντελεστής μετάδοσης  $\cong 0,7$ ,  $T_S$  η χρονική διάρκεια BLEVE,  $I$  η θερμική ακτινοβολία και  $L$  η απόσταση του παρατηρητή.

Είσοδος : θερμική ακτινοβολία  $I$

Έξοδος : Απόσταση παρατηρητή  $L$

Σταθερές παράμετροι :  $\pi$ ,  $E_B$  η ενέργεια καύσης,  $T_A$  ο ατμοσφαιρικός συντελεστής μετάδοσης

Μεταβλητές Παράμετροι :  $F$  το ποσοστό εκπεμπόμενης ακτινοβολίας,  $M_R$  η ολική ποσότητα εκροής,  $I$  η θερμική ακτινοβολία,  $T_S$  η χρονική διάρκεια BLEVE

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1 (1500 m)	439,765		
2 (450 m)	690,717		
3 (170 m)	995,03		
4 (40 m)	1711,91		

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της Ακτίνας ζώνης
- ◆ Υπολογισμός της απόστασης του παρατηρητή
- ◆ Εγκυρότητα του τύπου υπολογισμού της απόστασης του παρατηρητή

Και σ' αυτή τη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει το  $M_R$  απ' τον χρήστη και το  $P$ , ο υπολογισμός του  $T_S$  και του  $P$ , δεν είναι σωστός.

Οι προγραμματιστές θα πρέπει να δίνουν την δυνατότητα στον χρήστη ώστε να εισάγει τιμές στην μεταβλητή  $M_R$  και  $P$  και να διορθώσουν τα  $F$  και  $T_S$ .

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

## Συνάρτηση FlameJet και Default:

Στην συνέχεια, απ' την επιλογή του MainMenu, πληκτρολογώντας 2 για το φαινόμενο FlameJet, ο χρήστης, καλεί στην ουσία την συνάρτηση *void Default()* όπου η συνάρτηση αυτή καλεί την *double FlameJet(double FF)*, όπου η συνάρτηση *FlameJet*, υπολογίζει από προκαθορισμένο τύπο το μήκος της φλόγας για διαρροή λόγω διάβρωσης. Εδώ ο χρήστης έχει να εισάγει το ρυθμό διαρροής. Πληκτρολογώντας και τον επιθυμητό ρυθμό διαρροής, το λογισμικό πρέπει να του εμφανίσει το μήκος της φλόγας που θα προκύψει. Στην ουσία λοιπόν, θα ελέγξουμε την εισαγωγή τιμών μέσω της συνάρτησης *Default* για την εισαγωγή του ρυθμού διάβρωσης και την εγκυρότητα των τιμών που επιστρέφει η συνάρτηση *FlameJet*. Έτσι λοιπόν, οι τιμές που θα εισάγονται δεν υπάρχει περίπτωση να δημιουργήσουν εικασίες λανθασμένων τιμών όπως για παράδειγμα:

- ◆ Καμία τιμή (Enter)
- ◆ Γράμματα ή άλλοι χαρακτήρες (π.χ. @,#,"δ,a,?,B )
- ◆ Κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Μεγάλο πλήθος ακεραίων που δέχεται
- ◆ Σωστή ενημέρωση χρήστη
- ◆ Αρνητική τιμή
- ◆ Τιμές εκτός των ορίων

## Παρατηρήσεις:

Έλεγχος συνάρτησης `int Default(), case(2)` → **Επιλογή ρυθμού διαρροής**

### Ισοδύναμες κλάσεις τιμών :

- ◆  $< 0,17$  (Άκυρες κλάσεις)
- ◆  $> 1,30$  (Άκυρες κλάσεις)
- ◆  $0,17 \leq \text{τιμή} \leq 1,30$  (Έγκυρες κλάσεις)

### Οριακές τιμές :

- ◆ 0.17, 0.16, 0.18 (Οριακές τιμές)
- ◆ 1.30, 1.29, 1.31 (Οριακές τιμές)
- ◆ 1.00, 0.99, 1.01 (Οριακές τιμές)
- ◆ 1.01, 1.00, 1.02 (Οριακές τιμές)

### Εικασίες τιμών :

- ◆ Καμία τιμή (Enter)
- ◆ Μεγάλη τιμή (πχ 10000)
- ◆ Αρνητική τιμή (πχ -254)
- ◆ Μηδενική τιμή (0)
- ◆ Χαρακτήρας (πχ α, s, %)
- ◆ Κενό
- ◆ Τυχαίες τιμές μεταξύ των ορίων

<b>Ισοδύναμες κλάσεις τιμών (τιμές εισόδου)</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0.17(έγκυρη κλάση)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση αποτελεσμάτων	✗
1.30(έγκυρη κλάση)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση αποτελεσμάτων	✗
-5(άκυρη κλάση)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
5(άκυρη κλάση)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
<b>Οριακές τιμές</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0.17	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση αποτελεσμάτων	✗
0.16	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
0.18	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.30	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση αποτελεσμάτων	✗
1.29	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.31	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
1.00	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
0.99	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.01	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.01	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.00	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.02	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓

Εικασίες τιμών	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
Καμία τιμή (Enter)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	✗
1000 (Μεγάλη τιμή)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
- 254 (Αρνητική τιμή)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
a (χαρακτήρας)	Τερματισμός	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
Φ (χαρακτήρας)	Τερματισμός	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
\$ (χαρακτήρας)	Τερματισμός	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
_ (κενο)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	✗
0,53 (Τυχαίες τιμές)	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1,19 (Τυχαίες τιμές)	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ δεν παίρνει ως τιμή χαρακτήρες
- ◆ δεν παίρνει ως τιμή δεκαδικούς αριθμούς με περισσότερα από 2 δεκαδικά ψηφία
- ◆ δεν παίρνει ως τιμή αρνητικούς αριθμούς
- ◆ παίρνει τιμές από 0,17 έως 1,30 μόνο

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν κάποια λάθη, από την εισαγωγή τιμών εκτός ορίων, μη επιτρεπόμενων τιμών (όπως για παράδειγμα χαρακτήρες, αρνητικοί αριθμοί, δεκαδικά ψηφία κλπ).

Υπάρχουν όμως κάποιες παρατηρήσεις που η εφαρμογή τους είναι απαραίτητη σε κάποια σημεία της μεθόδου αυτά είναι:

- Δημιουργήθηκε μια κλάση εξαιρέσεων και όντως εμφανίζει μήνυμα λάθους κατά την εισαγωγή λανθασμένων τιμών, όμως εκεί όπου κατά τον έλεγχο βάλαμε επιτυχημένη και αποτυχημένη ενέργεια ταυτόχρονα, απλά σημαίνει ότι η κλάση εξαιρέσεων εμφανίζει το μήνυμα λανθασμένης εισαγωγής τιμών, όμως επιστρέφει αντί για να δώσεις νέα τιμή στην αρχή όπου βάζεις πάλι username και password. Αυτό μόνο προτείνουμε στους προγραμματιστές να διορθώσουν, ώστε μετά την αποτυχημένη προσπάθεια να μπορεί να γίνει πάλι εισαγωγή τιμών και να μην "βγαίνει" ο χρήστης απ' την συγκεκριμένη επιλογή.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 1-λάθη 16)

## Έλεγχος συνάρτησης double FlameJet(double FF)-->FlameJet→Φαινόμενο FlameJet

### Έλεγχος συνάρτησης Μήκους φλόγας για διαρροή :

Τύπος :  $F_F = 0,0006785 * L_F^{2,5}$  , όπου  $F_F$  η ροή του καυσίμου και  $L_F$  το μήκος της

$$\text{φλόγας, οπότε είναι } L_F = 2,5\sqrt{\frac{F_F}{0,0006785}}$$

Είσοδος :  $F_F$  η ροή του καυσίμου

Εξοδος : Μήκος φλόγας για διαρροή  $L_F$

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Ρυθμός Διαρροής  $F_F$

Είσοδος	Εξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1,19	19,8426	19,8426	✓
0,59	14,9872	14,9872	✓
0,18	9,32157	9,32157	✓
1,29	20,4935	20,4935	✓
1,00	18,5089	18,5089	✓
1,01	18,5827	18,5827	✓

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή του ρυθμού διαρροής
- ◆ Υπολογισμός του μήκους της πόρινης φλόγας
- ◆ Εγκυρότητα του τύπου υπολογισμού του μήκους της πόρινης φλόγας

Ελέγχθηκαν οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο δεν προέκυψαν κάποια λάθη, από την εισαγωγή τιμών, ώστε να οφείλονται σε λανθασμένο υπολογισμό του συγκεκριμένου τύπου.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 0 – λάθη 0)

## Στο αρχείο 16\_running.cpp

### Συνάρτηση pass, ReadFromFile και class BXptn :

Εδώ έχουμε τις 2 συναρτήσεις:

```
bool pass(string username,string password)
```

```
bool ReadFromFile(string user,string password )
```

και την κλάση εξαιρέσεων:

```
class BXptn
```

Θα ελέγξουμε παρακάτω τον τρόπο που λειτουργούν και που αλληλοσυνεργάζονται μεταξύ τους αυτές οι 2 συναρτήσεις και η 1 κλάση. Αρχικά να πούμε ότι κατά την εκκίνηση του συστήματος, ο χρήστης ανάλογα με το αν είναι administrator ή κοινός χρήστης, καλείται να πληκτρολογήσει ένα password και ένα username. Έτσι αρχίζει λοιπόν η κλήση της συνάρτησης *pass*, η οποία για να ελέγξει αν εισήχθηκε ο σωστός χρήστης, ελέγχει ένα αρχείο στο σκληρό δίσκο, με την συνάρτηση *ReadFromFile* και ανάλογα αν δεν εισήχθηκε ο σωστός χρήστης ή δεν υπάρχει ο χρήστης στο αρχείο, καλείται η κλάση εξαιρέσεων και εμφανίζει ένα μήνυμα σφάλματος.

Έτσι λοιπόν, για τον έλεγχο αυτών των συναρτήσεων, έχουμε τους παρακάτω περιορισμούς.

- ◆ Να υπάρχει στο αρχείο ο χρήστης
- ◆ Για εισαγωγή administrator, το username πρέπει να ξεκινά με ^, ενώ το password με ~
- ◆ Να μην υπάρχει κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Να μην είναι πολύ μεγάλο το πλήθος των χαρακτήρων που δέχεται
- ◆ Να μην γίνεται εισαγωγή χρήστη με ειδικούς χαρακτήρες, εκτός απ' τους administrator, όπου το username ξεκινά με ^ και το password με ~
- ◆ Να μην υπάρχει χρήστης 2<sup>ος</sup> με τα ίδια username και password
- ◆ Ο administrator μπορεί να θεωρείται και χρήστης, ο κοινός χρήστης όμως δε μπορεί να θεωρείται και administrator
- ◆ Σωστή ενημέρωση χρήστη

## Παρατηρήσεις:

### Έλεγχος συναρτήσεων

`bool pass(string username,string password)`

**Εισαγωγή κωδικών ασφαλείας**

`bool ReadFromFile(string user,string password )`

### Εικασίες τιμών :

- ◆ Ανύπαρκτα username και password
- ◆ username και password σωστά
- ◆ εισαγωγή ως administrator χωρίς όμως ^ και ~
- ◆ Ειδικόί Χαρακτήρες (πχ α, s, %)
- ◆ Μόνο αριθμοί
- ◆ Κενό ανάμεσα στις λέξεις

Για να πραγματοποιηθεί ο απαραίτητος έλεγχος, εισάγω στο αρχείο όπου βρίσκονται οι κωδικοί διάφορους κωδικούς όπως:

#### ***username      password***

- ❖ ^K              ~K      (έγκυρο, administrator)
- ❖ !!              !!      (άκυρο, μόνο ειδικόί χαρακτήρες)
- ❖ 12              34      (άκυρο, μόνο αριθμοί)
- ❖ α 0              ρλ      (άκυρο, κενό μέσα στη λέξη)
- ❖ gioufa          yfgefa (έγκυρο, κοινός χρήστης)
- ❖ makhs          lamprou (έγκυρο, κοινός χρήστης)
- ❖ asasasdfasdfaadfasdfasdfadfa adsfadsfasdfadfadfdafasdf (άκυρο, μεγάλοι κωδικοί)

Εικασίες τιμών Username-Password	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Σχόλιο	Επιτυχία / Αποτυχία
^K ~K	Εισαγωγή	Εισαγωγή	administrator	✓
^K ^K	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Δεν υπάρχει στο αρχείο	✓
~K ~K	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Δεν υπάρχει στο αρχείο	✓
~K ^K	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Δεν υπάρχει στο αρχείο	✓
gioufa yfgefa	Εισαγωγή	Εισαγωγή	Απλός χρήστης	✓
!! !!	Εισαγωγή	Εμφάνιση μηνύματος λάθους	Μόνο ειδικοί χαρακτήρες	✗
12 34	Εισαγωγή	Εμφάνιση μηνύματος λάθους	Μόνο αριθμοί	✗
** α 0 ρλ	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Κενό μεταξύ των χαρακτήρων	✓
makhs lamprou	Εισαγωγή	Εισαγωγή	Απλός χρήστης	✓
asdfsadfaadfasdfasdfadfa adsfasfasdfadfadfdafasdf	Εισαγωγή	Εμφάνιση μηνύματος λάθους	Πολύ μεγάλοι κωδικοί	✗
lewnidas pelopidas	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Δεν υπάρχει στο αρχείο	✓
	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Δεν υπάρχει στο αρχείο	✓

## Συνάρτηση changeAdmin :

Εδώ έχουμε την συνάρτηση:

```
void changeAdmin(string NAdmiUser,string NAdmipassword)
```

Αρχικά να πούμε ότι η συνάρτηση αυτή καλείται και εμφανίζεται ως επιλογή μόνο εάν έχει γίνει εισαγωγή ως administrator. Έτσι λοιπόν όταν κληθεί η συγκεκριμένη συνάρτηση, ο administrator έχει δικαίωμα ν'αλλάξει το username και το password και ν'αποθηκευτούν στο αρχείο.βασική προϋπόθεση είναι να ελεγχθεί αν υπάρχουν στο αρχείο ίδιοι κωδικοί και ΜΟΝΟ αν δεν υπάρχουν να γίνει η αλλαγή.

Έτσι λοιπόν, για τον έλεγχο της συνάρτησης,έχουμε τους παρακάτω περιορισμούς.

- ◆ Να μην υπάρχουν ίδιοι κωδικοί στο αρχείο
- ◆ Για εισαγωγή administrator,το username πρέπει να ξεκινά με ^, ενώ το password με ~
- ◆ Να μην υπάρχει κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Να μην είναι πολύ μεγάλο το πλήθος των χαρακτήρων που δέχεται
- ◆ Να μην υπάρχει χρήστης 2<sup>ος</sup> με τα ίδια username και password
- ◆ Σωστή ενημέρωση χρήστη

Έλεγχος συνάρτησης void changeAdmin(string NAdmiUser,string NAdmipassword)

→ **changeAdmin → Αλλαγή κωδικών πρόσβασης Administrator**

### Εικασίες τιμών :

- ◆ Εισαγωγή ήδη υπάρχοντων username και password
- ◆ Εισαγωγή κωδικών χωρίς ^ και ~ στην αρχή του κάθε κωδικού
- ◆ Εισαγωγή πολύ μεγάλου πλήθους χαρακτήρων κωδικού
- ◆ Εισαγωγή μόνο αριθμών

\* Στο αρχείο ήδη υπάρχει καταχωρημένος administrator με username : ^K και password : ~K.

Εικασίες τιμών Username-Password	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Σχόλιο	Επιτυχία / Αποτυχία
^K ~K	Μη ανταπόκριση συστήματος και σταμάτημα κάθε διεργασίας	Εμφάνιση μηνύματος λάθους και επανάκληση	Ήδη υπάρχουσες τιμές	×
K K	Μη ανταπόκριση συστήματος και σταμάτημα κάθε διεργασίας	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρη τιμή χωρίς ^ και ~	×
gioufa yfgefa	Μη ανταπόκριση συστήματος και σταμάτημα κάθε διεργασίας	Εμφάνιση μηνύματος λάθους και επανάκληση	Ήδη υπάρχουσες τιμές	×
1111 1111	Μη ανταπόκριση συστήματος και σταμάτημα κάθε διεργασίας	Εμφάνιση μηνύματος λάθους και επανάκληση	Εισαγωγή μόνο αριθμών	×
^manos ~kwtos	Μη ανταπόκριση συστήματος και σταμάτημα κάθε διεργασίας	Καταχώρηση αλλαγής	Έγκυρη τιμή	×
- ξκα	Μη ανταπόκριση συστήματος και σταμάτημα κάθε διεργασίας	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρη τιμή	×
Aaa bbb	Μη ανταπόκριση συστήματος και σταμάτημα κάθε διεργασίας	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρη τιμή	×
Saasfasfsssdff asfasfasffhgh	Μη ανταπόκριση συστήματος και σταμάτημα κάθε διεργασίας	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρη τιμή(μεγάλο πλήθος χαρακτήρων)	×
^Saasfasfsssdff ~asfasfasffhgh	Μη ανταπόκριση συστήματος και σταμάτημα κάθε διεργασίας	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρη τιμή(μεγάλο πλήθος χαρακτήρων)	×

### Παρατηρήσεις.

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν κάποια λάθη που δημιουργήσαμε εμείς για να δούμε κατά πόσο ανταποκρίνεται το πρόγραμμα στα λάθη και έχουμε να τονίσουμε κάποιες παρατηρήσεις:

- Κατά τον έλεγχο προέκυψαν μόνο λάθη και μη ανταπόκριση του συστήματος. Προφανώς θα πρέπει να συμπεριληφθούν τα λάθη στην κλάση εξαιρέσεων και να διορθωθεί η συνάρτηση ώστε να μπορεί να γίνει καταχώρηση των σωστών ενεργειών

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 1-λάθη 9)

## Συνάρτηση ReadFromFile:

Εδώ έχουμε την συνάρτηση:  
`void ReadFromFile()`

Και η συνάρτηση αυτή καλείται μόνο όταν γίνει εισαγωγή ως administrator και καλείται για καταχώρηση στοιχείων νέων χρηστών και έλεγχο αν υπάρχουν στ'αρχείο τα στοιχεία αυτά που θα εισαχθούν. Έτσι λοιπόν όταν κληθεί η συγκεκριμένη συνάρτηση, ο administrator έχει δικαίωμα ν'αλλάξει το username και το password και ν'αποθηκευτούν στο αρχείο.βασική προϋπόθεση είναι να ελεγχθεί αν υπάρχουν στο αρχείο ίδιοι κωδικοί και MONO αν δεν υπάρχουν να γίνει η αλλαγή.

Έτσι λοιπόν, για τον έλεγχο της συνάρτησης,έχουμε τους παρακάτω περιορισμούς.

- ◆ Να μην υπάρχουν ίδιοι κωδικοί στο αρχείο
- ◆ Να μην υπάρχει κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Να μην είναι πολύ μεγάλο το πλήθος των χαρακτήρων που δέχεται
- ◆ Να μην υπάρχει χρήστης 2<sup>ος</sup> με τα ίδια username και password
- ◆ Πληκτρολόγηση STOP για τερματισμό καταχώρησης στοιχείων
- ◆ Σωστή ενημέρωση χρήστη
- ◆ Δεν υπάρχει περιορισμός για κεφαλαία ή πεζούς χαρακτήρες

Έλεγχος συνάρτησης void ReadFromFile() → **ReadFromFile → Αλλαγή στοιχείων**

### Εικασίες τιμών :

- ◆ Εισαγωγή ήδη υπαρχόντων username και password
- ◆ Εισαγωγή στοιχείων που ξεκινούν με κενό ή έχουν ενδιάμεσα στους χαρακτήρες
- ◆ Λανθασμένη πληκτρολόγηση stop για τερματισμο
- ◆ Εισαγωγή πολύ μεγάλου πλήθους χαρακτήρων κωδικού
- ◆ Εισαγωγή μόνο αριθμών

**(Σε επόμενους ελέγχους, μετατράπηκε σε συνάρτηση void WriteToFile() )**

\* Για να πραγματοποιηθεί ο απαραίτητος έλεγχος, υπάρχουν στο αρχείο όπου βρίσκονται οι κωδικοί τα εξής στοιχεία:

<b>όνομα</b>	<b>επώνυμο</b>	<b>username</b>	<b>password</b>
George	Yfanths	gioyfa	yfgefa
Nikolaos	Bogiatzhs	nibogiat	mplampla
Marianna	Polymerou	mapolyme	mania
Babis	Yfanths	layf14	oxbow

Εικασίες τιμών Όνομα-Επώνυμο- Username-Password	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Σχόλιο	Επιτυχία / Αποτυχία
John Petrou john25 johnie	Καταχώρηση στοιχειων στο αρχείο	Καταχώρηση στοιχειων στο αρχείο	Έγκυρα στοιχεία	✓
George Yfanths gioyfa yfgefa	Καταχώρηση στοιχειων στο αρχείο	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρα στοιχεία (ήδη υπάρχουν)	✗
Manos Kwstas john25 johnie	Καταχώρηση στοιχειων στο αρχείο	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρα στοιχεία (ήδη υπάρχουν το username και το password)	✗
Alekos Alexiou ela_aleko yfgefa	Καταχώρηση στοιχειων στο αρχείο	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρα στοιχεία (ήδη υπάρχει το password)	✗
Mar8a Mar8akiou layf14 crazygirl	Καταχώρηση στοιχειων στο αρχείο	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρα στοιχεία (ήδη υπάρχει το username)	✗
Geo rge Karamhtros ougka ougka	Καταχώρηση στοιχειων στο αρχείο	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρα στοιχεία (χει κενό στο όνομα και το παίρνει ως όνομα: Geo, επώνυμο: rge, user: karamhtros και pass: ougka)	✗
STOP	Καταχώρηση στοιχειων στο αρχείο ως Όνομα	Έξοδος	(για έξοδο)	✗
Stop	Καταχώρηση στοιχειων στο αρχείο ως Όνομα	Έξοδος	(για έξοδο)	✗
stop	Έξοδος	Έξοδος	(για έξοδο)	✓

### Παρατηρήσεις.

Ελέγχθηκαν αρκετές περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα και κατά τον έλεγχο προέκυψαν αρκετά λάθη που δημιουργήσαμε εμείς για να δούμε κατά πόσο ανταποκρίνεται το πρόγραμμα στα λάθη και έχουμε να τονίσουμε κάποιες παρατηρήσεις:

- Κατά τον έλεγχο συμπεράναμε ότι το πρόγραμμα δεν ελέγχει στο αρχείο για ύπαρξη ίδιων στοιχείων και έτσι μπορεί να κάνει εισαγωγή τιμών που ήδη υπάρχουν, ενώ δε θα πρέπει.
- Για τον τερματισμό απ' την διαδικασία, πρέπει να τονίζεται αν θα τερματίζει με κεφαλαία ή μικρα γράμματα η λέξη STOP και θα πρέπει να υπάρχει έλεγχος, ώστε όπως και αν το δώσει ο χρήστης, αυτό να τερματίζεται.
- Επίσης, θα πρέπει να υπάρχουν έλεγχοι και να συμπεριληφθούν και αυτοί στη κλάση εξαιρέσεων, ώστε να εμφανίζει μηνύματα σφαλμάτων όπου υπάρχουν.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 3-λάθη 7)

## Συνάρτηση PoolFire:

Εδώ έχουμε την συνάρτηση:

```
void PoolFire(const double MR,double &RP,double &HF,double &Q,double  
&EP,  
              double &l,double &TDU_I1,double &TDU_I2,double  
&TDU_I3,double &TDU_I4,double  
&TDU_I5,double &X1,double &X2,double &X3,double  
&X4,double &X5)
```

Η συνάρτηση αυτή καλείται μέσω της συνάρτησης default και ανάλογα με την επιλογή που δίνει ο χρήστης, αν κατά την κλήση της επιλογής φαινομένου επιλέξει 3, τότε ο χρήστης εισάγει την ποσότητα διαρροής. Έτσι λοιπόν κατά την κλήση της συνάρτησης PoolFire, γίνεται ο υπολογισμός α) της ακτίνας λίμης υγρού, β) του ύψους της φλόγας, γ) του εμβαδού της λίμνης του υγρού, δ) της ολικής θερμότητας εκπομπής, ε) της έντασης της θερμικής ακτινοβολίας και τέλος για κάθε μια απ' τις 5 προκαθορισμένες ζώνες της έντασης της θερμικής ακτινοβολίας ανά ζώνη και της απόστασης απ' το κέντρο της φλόγας.

Έτσι λοιπόν, θα ελέγξουμε για εικασίες τιμών για την ποσότητα διαρροής που δίνει ο χρήστης, μιας και *δεν έχουν καθοριστεί ακόμη τα επιτρεπτά όρια*. Έτσι δε μπορούμε ακόμη να θέσουμε τους περιορισμούς όπως κάναμε στους προηγούμενους ελέγχους, αλλά να κάνουμε μόνο εικασίες τιμών εισαγωγής. Και έπειτα θα ελέγξουμε έναν προς έναν τους τύπους που χρησιμοποιούνται.

```
Έλεγχος συνάρτησης void PoolFire(const double MR,double  
    &RP,double &HF,double &Q,double &EP,double &l,double  
    &TDU_I1,double &TDU_I2,double &TDU_I3,double  
    &TDU_I4,double &TDU_I5,double &X1,double &X2,double  
    &X3,double &X4,double &X5)  
--> PoolFire→Φαινόμενο PoolFire
```

### Εικασίες τιμών :

- ◆ Εισαγωγή αρνητικών τιμών
- ◆ Εισαγωγή του μηδέν (0)
- ◆ Εισαγωγή μεγάλων αριθμών
- ◆ Εισαγωγή πραγματικών και ακεραίων τιμών
- ◆ Εισαγωγή χαρακτήρων

Εικασίες τιμών	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
0 (Μηδέν)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
10000 (Μεγάλη τιμή)	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
- 254 (Αρνητική τιμή)	Εμφάνιση λανθασμένων τιμών	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επιστροφή στην Default	✗
0,53 (Τυχαίες τιμές)	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1,19 (Τυχαίες τιμές)	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓

### Παρατηρήσεις.

Ελέγχθηκαν αρκετές περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα και κατά τον έλεγχο προέκυψαν αρκετά λάθη που δημιουργήσαμε εμείς για να δούμε κατά πόσο ανταποκρίνεται το πρόγραμμα στα λάθη και έχουμε να τονίσουμε κάποιες παρατηρήσεις:

- Κατά τον έλεγχο συμπεράναμε ότι το πρόγραμμα δεν έχει συγκεκριμένα όρια καθορισμένα και θα πρέπει να καθοριστούν ώστε να γίνει ακόμη πιο συγκεκριμένος ο έλεγχος.
- Επίσης, θα πρέπει να υπάρχουν έλεγχοι και να συμπεριληφθούν και αυτοί στη κλάση εξαιρέσεων, ώστε να εμφανίζει μηνύματα σφαλμάτων όπου υπάρχουν.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2-λάθη 6)

### Έλεγχος συνάρτησης Ακτίνας λίμνης υγρού :

Τύπος :  $R_p = \sqrt{\frac{M_R}{\pi * H_L * D_L}}$  , όπου  $R_p$  η ακτίνα λίμνης του υγρού,  $M_R$  η ολική ποσότητα εκροής,  $H_L$  το βάθος της λίμνης υγρού και  $D_L$  η πυκνότητα του υγρού, οπότε είναι

Είσοδος :  $M_R$  η ολική ποσότητα εκροής

Εξόδος :  $R_p$  η ακτίνα λίμνης του υγρού

Σταθερές παράμετροι :  $\pi$ ,  $H_L$  το βάθος της λίμνης υγρού και  $D_L$  η πυκνότητα του υγρού

Μεταβλητές Παράμετροι :  $M_R$  η ολική ποσότητα εκροής

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	44.7784		
150000	38.7792		
500000	70.8008		
1000000	100.127		
0	Έξοδος και σφάλμα συστήματος		
-100	+NAN		
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη		

## Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της ολικής ποσότητας εκροής
- ◆ Υπολογισμός της ακτίνας λίμνης του υγρού
- ◆ Εγκυρότητα του τύπου υπολογισμού της ακτίνας λίμνης του υγρού

Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει το  $M_R$  απ' τον χρήστη, δημιουργήθηκε συνάρτηση με σταθερό το  $M_R = 314794$  και προκύπτει συνέχεια σταθερό αποτέλεσμα.

Επίσης, θα πρέπει οι προγραμματιστές να επιτρέπουν στον χρήστη εισαγωγή τιμών εντός των ορίων, τα οποία και θα ελέγχουν, καθώς επίσης θα ελέγχουν και περιπτώσεις εισαγωγής αρνητικών αριθμών και εισαγωγής χαρακτήρων και συνδυασμό αριθμών-χαρακτήρων.

Οι προγραμματιστές θα πρέπει να δημιουργήσουν επιλογή στον χρήστη ώστε να εισάγει τιμές στην μεταβλητή  $M_R$  και να προκύπτουν διαφορετικά αποτελέσματα για κάθε εισαγωγή τιμών απ' τον χρήστη.

**ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 3 – λάθη ..)

### Έλεγχος συνάρτησης Ύψους φλόγας :

$$\text{Τύπος : } H_F = 2 * R_p * K_1 * \left( \frac{M_{BR}}{D_A * \sqrt{2 * R_p * G}} \right)^{0.6}, \text{ όπου } R_p \text{ η ακτίνα λίμνης του υγρού,}$$

$H_F$  το ύψος της φλόγας,  $K_1$  σταθερά (=42),  $M_{BR}$  ειδικός ρυθμός καύσης,  $D_A$  η πυκνότητα του ατμοσφαιρικού αέρα,  $R_p$  η ακτίνα λίμνης υγρού και  $G$  η επιτάχυνση της βαρύτητας

*Είσοδος :*  $R_p$  η ακτίνα λίμνης του υγρού (που υπολογίζεται με το προηγούμενο τύπο)

*Εξοδος :*  $H_F$  το ύψος της φλόγας

*Σταθερές παράμετροι :*  $K_1$  σταθερά (=42),  $M_{BR}$  ειδικός ρυθμός καύσης,  $D_A$  η πυκνότητα του ατμοσφαιρικού αέρα,  $R_p$  η ακτίνα λίμνης υγρού και  $G$  η επιτάχυνση της βαρύτητας

*Μεταβλητές Παράμετροι :*  $R_p$  η ακτίνα λίμνης του υγρού

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	81.5829		
150000	73.7685		
500000	112.429		
1000000	143.298		
0	Έξοδος και σφάλμα συστήματος		
-100	+NAN		
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη		

## Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της ακτίνα λίμνης του υγρού (που υπολογίζεται με το προηγούμενο τύπο)
- ◆ Υπολογισμός του ύψους της φλόγας
- ◆ Εγκυρότητα του τύπου υπολογισμού του ύψους της φλόγας

Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει το  $M_R$  απ' τον χρήστη για τον υπολογισμό της ακτίνας λίμνης του υγρού, όπου είναι είσοδος σ' αυτή τη συνάρτηση, δημιουργήθηκε η συνάρτηση με σταθερό το  $M_R = 314794$  και προκύπτει συνέχεια λάθος αποτέλεσμα.

Οι προγραμματιστές θα πρέπει να δημιουργήσουν επιλογή στον χρήστη ώστε να εισάγει τιμές στην μεταβλητή  $M_R$  και να προκύπτουν διαφορετικά αποτελέσματα για κάθε εισαγωγή τιμών απ' τον χρήστη.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

### Έλεγχος συνάρτησης Εμβαδού λίμνης υγρού :

Τύπος :  $E_p = 2 * \pi * R_p^2$  , όπου  $R_p$  η ακτίνα λίμνης του υγρού και  $E_p$  το εμβαδό λίμνης υγρού

Είσοδος :  $R_p$  η ακτίνα λίμνης του υγρού

Έξοδος :  $E_p$  το εμβαδό λίμνης υγρού

Σταθερές παράμετροι :  $\pi$

Μεταβλητές Παράμετροι :  $R_p$  η ακτίνα λίμνης του υγρού

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	12598.4		
150000	9448.82		
500000	31496.1		
1000000	62992.1		
0	Έξοδος και σφάλμα συστήματος		
-100	+NAN		
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη		

#### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- Εισαγωγή της ακτίνα λίμνης του υγρού (που υπολογίζεται με το προηγούμενο τύπο)
- Υπολογισμός του εμβαδού λίμνης υγρού
- Εγκυρότητα του τύπου υπολογισμού του εμβαδού λίμνης υγρού

Και στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει το  $M_R$  απ' τον χρήστη για τον υπολογισμό της ακτίνας λίμνης του υγρού, όπου είναι είσοδος σ' αυτή τη συνάρτηση, δημιουργήθηκε η συνάρτηση με σταθερό το  $M_R = 314794$  και προκύπτει συνέχεια λάθος αποτέλεσμα.

Οι προγραμματιστές θα πρέπει να δημιουργήσουν επιλογή στον χρήστη ώστε να εισάγει τιμές στην μεταβλητή  $M_R$  και να προκύπτουν διαφορετικά αποτελέσματα για κάθε εισαγωγή τιμών απ' τον χρήστη. Καθώς επίσης να ορίζονται και τα όρια εισαγωγής των επιτρεπτών τιμών

#### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

### Έλεγχος συνάρτησης Ολικής Θερμότητας Εκπομπής :

Τύπος :  $Q = K_2 * E_p * D_L * C_L$  , όπου Q η ολική θερμότητα εκπομπής,  $K_2$  σταθερά (=5,1\*10<sup>-5</sup>),  $E_p$  το εμβαδό της λίμνης,  $D_L$  η πυκνότητα του υγρού και  $C_L$  η καθαρή θερμογόνος δύναμη του υγρού.

Είσοδος :  $E_p$  το εμβαδό της λίμνης (όπου υπολογίζεται απ' την προηγούμενη συνάρτηση)

Έξοδος : Q η ολική θερμότητα εκπομπής

Σταθερές παράμετροι :  $K_2$  σταθερά (=5,1\*10<sup>-5</sup>),  $E_p$  το εμβαδό της λίμνης,  $D_L$  η πυκνότητα του υγρού και  $C_L$  η καθαρή θερμογόνος δύναμη του υγρού

Μεταβλητές Παράμετροι :  $E_p$  το εμβαδό της λίμνης

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	1.836e+07		
150000	1.377e+07		
500000	4.59e+07		
1000000	9.18e+07		
0	Έξοδος και σφάλμα συστήματος		
-100	+NAN		
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη		

#### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή του εμβαδού λίμνης υγρού
- ◆ Υπολογισμός της ολικής θερμότητας εκπομπής
- ◆ Εγκυρότητα του τύπου υπολογισμού της ολικής θερμότητας εκπομπής

Και εδώ στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει το  $E_p$  (που σημαίνει ότι θα πρέπει να αλλάζει και το  $M_R$  απ' τον χρήστη για τον υπολογισμό του εμβαδού της ακτίνας λίμνης του υγρού, όπου είναι είσοδος σ' αυτή τη συνάρτηση), δημιουργήθηκε η συνάρτηση με σταθερό το  $M_R = 314794$  και προκύπτει συνέχεια λάθος αποτέλεσμα.

Οι προγραμματιστές θα πρέπει να δημιουργήσουν επιλογή στον χρήστη ώστε να εισάγει τιμές στην μεταβλητή  $M_R$  και να προκύπτουν διαφορετικά αποτελέσματα για κάθε εισαγωγή τιμών απ' τον χρήστη. Καθώς επίσης να ορίζονται και τα όρια εισαγωγής των επιτρεπτών τιμών

**ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

### Έλεγχος συνάρτησης Έντασης Θερμικής Ακτινοβολίας :

Τύπος :  $I = K_3 * \frac{E_p * D_L * C_L}{\pi * R_p * H_F + E_p}$  , όπου I είναι η ένταση της θερμικής ακτινοβολίας,

$K_3$  είναι σταθερά ( $=2,5*10^{-5}$ ),  $E_p$  είναι το εμβαδό της λίμνης υγρού,  $D_L$  η πυκνότητα του υγρού,  $C_L$  η καθαρή θερμογόνος δύναμη του υγρού,  $R_p$  η ακτίνα λίμνης του υγρού,  $H_F$  το ύψος της φλόγας

Είσοδος :  $E_p$  το εμβαδό της λίμνης υγρού και  $R_p$  η ακτίνα λίμνης του υγρού

Εξοδος : I η ένταση της θερμικής ακτινοβολίας

Σταθερές παράμετροι :  $\pi$ ,  $K_3$ ,  $D_L$  η πυκνότητα του υγρού,  $C_L$  η καθαρή θερμογόνος δύναμη του υγρού,  $H_F$  το ύψος της φλόγας

Μεταβλητές Παράμετροι :  $E_p$  το εμβαδό της λίμνης υγρού και  $R_p$  η ακτίνα λίμνης του υγρού

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	1.836e+07		
150000	1.377e+07		
500000	4.59e+07		
1000000	9.18e+07		
0	Έξοδος και σφάλμα συστήματος		
-100	+NAN		
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος		
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη		

#### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- Εισαγωγή του εμβαδού της λίμνης υγρού και την ακτίνα λίμνης του υγρού
- Υπολογισμός της έντασης της θερμικής ακτινοβολίας
- Εγκυρότητα του τύπου υπολογισμού της έντασης της θερμικής ακτινοβολίας

Και στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν ανεπιτυχής, διότι ενώ απ' τις προδιαγραφές προκύπτει ότι στον παραπάνω τύπο θα αλλάζει το  $E_p$  και το  $R_p$  (που σημαίνει ότι θα πρέπει να αλλάζει και το  $M_R$  απ' τον χρήστη για τον υπολογισμό του εμβαδού της ακτίνας λίμνης του υγρού, και του εμβαδού της λίμνης υγρού όπου είναι είσοδος σ' αυτή τη συνάρτηση), δημιουργήθηκε η συνάρτηση με σταθερό το  $M_R = 314794$  και προκύπτει συνέχεια λάθος αποτέλεσμα.

Οι προγραμματιστές θα πρέπει να δημιουργήσουν επιλογή στον χρήστη ώστε να εισάγει τιμές στην μεταβλητή  $M_R$  και να προκύπτουν διαφορετικά αποτελέσματα για κάθε εισαγωγή τιμών απ' τον χρήστη. Καθώς επίσης να ορίζονται και τα όρια εισαγωγής των επιτρεπών τιμών

**ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη ..)

### Έλεγχος συνάρτησης Δόσης Θερμικής Ακτινοβολίας (TDU) ανά Ζώνη :

Τύπος :  $TDU_I(x) = \left( \frac{Z\Omega NH}{T} \right)^{3/4}$  , όπου TDU\_I(x) η δόση θερμικής ακτινοβολίας και x η κάθε μια περίπτωση ζώνης, ZΩNH η κάθε μια απ' τις 4 ζώνες και T σταθερά (=40)

Είσοδος : ZΩNH η κάθε μια απ' τις 4 ζώνες

Έξοδος : TDU\_I(x) η δόση θερμικής ακτινοβολίας και x η κάθε μια περίπτωση ζώνης

Σταθερές παράμετροι : T

Μεταβλητές Παράμετροι : ZΩNH η κάθε μια απ' τις 4 ζώνες

#### Ζώνη I (1500 m) :

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	15,1539	15,1539	✓
150000	15,1539	15,1539	✓
500000	15,1539	15,1539	✓
1000000	15,1539	15,1539	✓
0	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
-100	+NAN	Εμφάνιση μηνύματος λάθους	✗
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

**Ζώνη II (450 m) :**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	6.14277	6.14277	✓
150000	6.14277	6.14277	✓
500000	6.14277	6.14277	✓
1000000	6.14277	6.14277	✓
0	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
-100	+NAN	Εμφάνιση μηνύματος λάθους	✗
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

**Ζώνη III (170 m) :**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	2.96	2.96	✓
150000	2.96	2.96	✓
500000	2.96	2.96	✓
1000000	2.96	2.96	✓
0	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
-100	+NAN	Εμφάνιση μηνύματος λάθους	✗
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

### Ζώνη IV (40 m) :

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	1	1	✓
150000	1	1	✓
500000	1	1	✓
1000000	1	1	✓
0	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
-100	+NAN	Εμφάνιση μηνύματος λάθους	✗
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή του ρυθμού διαρροής ανά ζώνη
- ◆ Υπολογισμός της δόσης θερμικής ακτινοβολίας ανά ζώνη
- ◆ Εγκυρότητα του τύπου υπολογισμού της δόσης θερμικής ακτινοβολίας ανά ζώνη

Ελέγχθηκαν οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο δεν προέκυψαν κάποια σοβαρά λάθη, από την εισαγωγή τιμών, ώστε να οφείλονται σε λανθασμένο υπολογισμό του συγκεκριμένου τύπου. Οπότε ο έλεγχος της εγκυρότητας του ταΐθπου για κάθε ζώνη θεωρείται επιτυχής.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 1 – λάθη 6/περίπτωση)

**Έλεγχος συνάρτησης Απόστασης απ' το κέντρο της Φλόγας (χ) ανά Ζώνη :**

Τύπος :  $X = \sqrt{\frac{Q * \cos \theta}{4 * \pi * TDU\_I}}$  , όπου X η απόσταση απ' το κέντρο της φλόγας, Q η

ολική θερμότητα εκπομπής, cosθ το συνήμιτονο της γωνίας θέασης και TDU\_I η ένταση της θερμικής ακτινοβολίας ανάλογα με την ζώνη

Είσοδος : TDU\_I η ένταση της θερμικής ακτινοβολίας ανάλογα με την ζώνη (στην ουσία η ζώνη αλλάζει δηλαδή)

Έξοδος : X η απόσταση απ' το κέντρο της φλόγας

Σταθερές παράμετροι : Q η ολική θερμότητα εκπομπής, cosθ το συνήμιτονο της γωνίας θέασης, π

Μεταβλητές Παράμετροι : : TDU\_I η ένταση της θερμικής ακτινοβολίας ανάλογα με την ζώνη

**Ζώνη I (1500 m) :**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	288.958	288.958	✓
150000	250.245	250.245	✓
500000	456.883	456.883	✓
1000000	646.13	646.13	✓
0	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
-100	+NAN	Εμφάνιση μηνύματος λάθους	✗
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

**Ζώνη II (450 m) :**

<b>Είσοδος</b>	<b>Έξοδος</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
200000	453.852	453.852	✓
150000	393.048	393.048	✓
500000	717.603	717.603	✓
1000000	1014.84	1014.84	✓
0	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
-100	+NAN	Εμφάνιση μηνύματος λάθους	✗
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

**Ζώνη III (170 m) :**

<b>Είσοδος</b>	<b>Έξοδος</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
200000	653.809	653.809	✓
150000	566.215	566.215	✓
500000	1033.76	1033.76	✓
1000000	1461.96	1461.96	✓
0	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
-100	+NAN	Εμφάνιση μηνύματος λάθους	✗
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

### Ζώνη IV (40 m) :

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	1124.86	1124.86	✓
150000	974.153	974.153	✓
500000	1778.55	1778.55	✓
1000000	2515.25	2515.25	✓
0	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
-100	+NAN	Εμφάνιση μηνύματος λάθους	✗
a (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
Φ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
\$ (χαρακτήρας)	Έξοδος και σφάλμα συστήματος	Εμφάνιση μηνύματος λάθους	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

#### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της έντασης θερμικής ακτινοβολίας ανά ζώνη
- ◆ Υπολογισμός της απόστασης απ' το κέντρο της φλόγας
- ◆ Εγκυρότητα του τύπου υπολογισμού της απόστασης απ' το κέντρο της φλόγας

Ελέγχθηκαν οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο δεν προέκυψαν κάποια σοβαρά λάθη, από την εισαγωγή τιμών, ώστε να οφείλονται σε λανθασμένο υπολογισμό του συγκεκριμένου τύπου. Οπότε ο έλεγχος της εγκυρότητας του τα΄θπου για κάθε ζώνη θεωρείται επιτυχής.

#### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 1 – λάθη 6/περίπτωση)

Β' Έλεγχος  
Ανοιχτού Κουτιού



Ελεγκτής Α' : Υφαντής Γεώργιος  
Ελεγκτής Β' : Αναστασιάδου Έλενα

Ανάπτυξη Λογισμικού για την Βιομηχανία Πετρελαίου Kavala Oil

**ΗΜΕΡΟΛΟΓΙΟ ΕΛΕΓΚΤΗ: 15 Μαΐου 2004**

**Ver 1.0**

**Ελέγχθηκαν τα αρχεία :**

● 16\_running.cpp

**και οι συναρτήσεις :**

- BXptn
- ChangeAdmin
- Pass
- ReadFromFile
- WriteToFile
- Blevé
- PoolFire
- FlameJet
- MainMenu
- ZoneSelect
- Default
- main

Θα τις εξετάσουμε λοιπόν μια προς μια:

## 1. Έλεγχος της κλάσης εξαιρέσεων BΧrtn

```
class BΧrtn {
private:
    string      synarthsh;
    unsigned int  eidos;
    int         timh1;
public:
    enum {ektosoriwn,maxstring,lathosuser};
    BΧrtn(string afethria, unsigned int problhma, int t1=0)
    {
        synarthsh=afethria;
        eidos=problhma;
        timh1=t1;
    }
    string getOnoma() const {return synarthsh;}
    unsigned int getEidos() const {return eidos;}
    int getTimh1() const {return timh1;}
};
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στον κώδικα αυτό δεν έχουμε κάποια από τις παραπάνω περιπτώσεις, παρά μόνο τη δήλωση της κλάσης εξαιρέσεων.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 2. Έλεγχος της συνάρτησης changeAdmin(string NAdmiUser,string NAdmipassword)

```
void changeAdmin(string NAdmiUser,string NAdmipassword)
{
    fstream o;
    o.open("Arxeio.txt",3);
    string User,Password,Name,Last;
    string temp="",temp1,temp2,temp3,temp4,stohos1,stohos2;

    o>>Name>>Last>>User>>Password;
    while(!o.eof())
    {
        if(User[0]!='^' && Password[0]!='~')
        {
            stohos1=User;
            stohos2=Password;
            goto label2;
        }
    }
    label2:
    o.close();
    o.open("Arxeio.txt",3);

    o>>temp1>>temp2>>temp3>>temp4;
    while(!o.eof())
    {
        temp=temp + temp1;
        temp=temp + " ";
        temp=temp +temp2;
        temp=temp + " ";
        temp=temp +temp3;
        temp=temp + " ";
        temp=temp +temp4;
        temp=temp +"\n";
        o>>temp1>>temp2>>temp3>>temp4;
    }

    temp.replace(temp.find(stohos1),stohos1.length(),'^'+NAdmiUser);
    temp.replace(temp.find(stohos2),stohos2.length(),'~'+NAdmipassword);

    o.close();
    o.open("Arxeio.txt",3);

    o<<temp;
    o.close();
}
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στην συνάρτηση αυτή έχουμε αλλαγή username και password.

- Έχουμε μια while η οποία διαβάζει το αρχείο "Αρχείο.txt" και βάζει σε ένα string το username και το password του χρηστη.
- Έχουμε μια while
- Τέλος με την replace αντικαθιστούμε τα στοιχεία

**ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

### 3. Έλεγχος της συνάρτησης pass(string username, string password)

```
bool pass(string username,string password){ //1

    bool a=false;
        fstream o;
        o.open("Arxeio.txt",3);
    string User,Password,Name,Last;
    o>>Name>>Last>>User>>Password;
    while(!o.eof()&& a==false)
    {
        if(username[0]=='^' && password[0]=='~')
            //username.length();
            //password.length();

            {
                a=true;
                goto label1;
            }
    o>>Name>>Last>>User>>Password;

    }//while

label1:

    return a;
}
```

#### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

#### Παρατηρήσεις:

Στην παραπάνω συνάρτηση έχουμε

- Μια while(!o.eof()&& a==false) η οποία διαβάζει ένα αρχείο
- Με την if ελέγχει αν το username και το password έχουν στην αρχή το συνθηματικό και αν το έχουν επιστρέφει true.
- Όταν ανοίγουμε το αρχείο δεν γίνεται έλεγχος αν το αρχείο υπάρχει και άνοιξε.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

#### 4. Έλεγχος της συνάρτησης ReadFromFile(string user,string password )

```
bool ReadFromFile(string user,string password ) //1
{
    bool a=false;
    fstream o;
    o.open("Arxeio.txt",3);
    string User,Password,Name,Last;
    o>>Name>>Last>>User>>Password;
    while(!o.eof()&& a==false)
    {

    if(user==User && password==Password)
        {
            a=true;
            goto label1;
        }
    o>>Name>>Last>>User>>Password;

    }//while

label1:

return a;
}
```

#### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

#### Παρατηρήσεις:

Στην παραπάνω συνάρτηση έχουμε:

- Μια while(!o.eof()&& a==false) η οποία διαβάζει ένα αρχείο
- Μια if ελέγχει αν το username και το password είναι το ίδιο με αυτό στο αρχείο και αν είναι επιστρέφει true.
- Όταν ανοίγουμε το αρχείο δεν γίνεται έλεγχος αν το αρχείο υπάρχει και άνοιξε.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 5. Έλεγχος της συνάρτησης WriteToFile()

```
void WriteToFile() //1
{
    unsigned int n1,l1,u1,p1;
    string Name,Last,User>Password,keno=" ",g="\n";
    char ch;
    fstream i;
    i.open("Arxeio.txt",3);

    while(!i.eof())
    {
        i.get(ch);

    }

    do
    {

    cout<<"GIA TELOS DWSE STOP"<<endl;
    cout<<"DOSE ONOMA : ";
    cin>>Name;

    if(Name=="stop")break;

    cout<<"\nDOSE EPWNYMO : ";
    cin>>Last;
    cout<<"\nDOSE USERNAME XRHSTH : ";
    cin>>User;
    cout<<"\nDOSE PASSWORD XRHSTH : ";
    cin>>Password;

    n1=Name.length();
    l1=Last.length();
    u1=User.length();
    p1>Password.length();
    if( n1>10 || l1>15 || u1>10 || p1>7 )
    {
        throw BXpnt("WriteToFile", BXpnt::maxstring);
    }
    break;
    }

    i<<(Name+keno)<<" " <<(Last+keno)<<" " <<(User+keno)<<"
    "<<(Password+keno+g);
    }while( Name !="stop");

}
```

## Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

## Παρατηρήσεις:

Στην συνάρτηση WriteToFile γράφουμε τα στοιχεία μας

- Έχουμε μια while που διαβάζει το αρχείο
- Μια do while στην οποία όσο ο χρήστης δεν δίνει stop ζητάει όνομα επώνυμο username και password
- Με μια if ελέγχετε το μήκος των στοιχείων και σε περίπτωση λάθους βγάζει το κατάλληλο μήνυμα αλλιώς το καταχωρεί στο αρχείο. Εκτός από το μήκος δεν ελέγχετε τη δίνει ο χρήστης .
- Θα έπρεπε να ελέγχονται τα εξής:
  - Για το όνομα να δίνονται μόνο χαρακτήρες (όχι αριθμοί ή σύμβολα), να μην υπάρχουν κενά μεταξύ των χαρακτήρων και το μήκος να είναι τουλάχιστον 3.
  - Για το επώνυμο να δίνονται μόνο χαρακτήρες (όχι αριθμοί ή σύμβολα), να μην υπάρχουν κενά μεταξύ των χαρακτήρων και το μήκος να είναι τουλάχιστον 3.
  - Για το username να δίνονται μόνο χαρακτήρες (όχι αριθμοί ή σύμβολα), να μην υπάρχουν κενά μεταξύ των χαρακτήρων και το μήκος να είναι τουλάχιστον 3.
  - Για το password να δίνονται από 1 έως 6 χαρακτήρες και από 1 έως 6 αριθμούς (όχι σύμβολα), να μην υπάρχουν κενά μεταξύ τους και το μήκος να είναι τουλάχιστον 3

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 6. Έλεγχος της συνάρτησης Blevé(int zone, double &Rs, double &Ts, double &F, double &I, double &L)

```
void Blevé(int zone, double &Rs, double &Ts, double &F, double &I, double &L)
{
    double t1,t2,t3;

    t3= pow(314794,0.325);
    Rs=((0.5*6.48)*t3);

    Ts=k2*(pow(Mr,0.26));

    F=k3*(pow(P,0.32));

    I=pow((zone/Ts),0.75);

    t1= F*Mr*Eb*Ta;
    t2=4*M_PI*I*1000*Ts;
    L=sqrt((t1/t2));
}
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στον κώδικα αυτό δεν έχουμε κάποια από τις παραπάνω περιπτώσεις, παρά μόνο τύπους οι οποίοι ελέγχθηκαν ένας προς ένας και βρέθηκαν τα εξής λάθη: Οι παρακάτω τύποι δεν έχουν μεταβλητό μέρος το οποίο θα πρέπει να δίνετε από το χρήστη:

t3= pow(314794,0.325)

Rs=((0.5\*6.48)\*t3)

Ts=k2\*(pow(Mr,0.26))

F=k3\*(pow(P,0.32))

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 7. Έλεγχος της συνάρτησης PoolFire0

```
void PoolFire(const double MR,double &RP,double &HF,double &Q,double
&EP,double &I,double &TDU_I1,double &TDU_I2,double &TDU_I3,double
&TDU_I4,double &TDU_I5,double &X1,double &X2,double &X3,double
&X4,double &X5)
{
  RP=sqrt(MR/(M_PI*HL*DL));

  HF=(2*RP*K1)*(pow((MBR/(DA*(sqrt(2*RP*G)))) , 0.6) );

  EP=2*M_PI*pow(RP,2);

  Q=K2*EP*DL*CL;

  I=K3*((EP*DL*CL)/((M_PI*RP*HF)+EP));

  TDU_I1=pow((zone1/T),(3./4));

  X1=sqrt( (Q*( cos(DEG*(M_PI/180)) ))/(4*M_PI*TDU_I1));

  TDU_I2=pow((zone2/T),0.75);

  X2=sqrt( (Q*cos(DEG*(M_PI/180)))/(4*M_PI*TDU_I2));

  TDU_I3=pow((zone3/T),0.75);

  X3=sqrt( (Q*cos(DEG*(M_PI/180)))/(4*M_PI*TDU_I3));

  TDU_I4=pow((zone4/T),0.75);

  X4=sqrt( (Q*cos(DEG*(M_PI/180)))/(4*M_PI*TDU_I4));

  TDU_I5=37.5;

  X5=sqrt( (Q*cos(DEG*(M_PI/180)))/(4*M_PI*TDU_I5));
}
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στον κώδικα αυτό δεν έχουμε κάποια από τις παραπάνω περιπτώσεις, παρά μόνο τύπους οι οποίοι ελέγχθηκαν ένας προς ένας και δεν παρατηρήθηκε κανένα λάθος.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 8. Έλεγχος της συνάρτησης FlameJet(double FF)

```
double FlameJet(double FF)
{
    double LF;
    LF=pow((FF/0.0006785),(1/2.5));

    return LF;
}
```

### Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στον κώδικα αυτό δεν έχουμε κάποια από τις παραπάνω περιπτώσεις, παρά μόνο ένας τύπος ο οποίος ελέγχθηκε και δεν παρατηρήθηκε κανένα λάθος.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 9. Έλεγχος της συνάρτησης MainMenu(string user,string pas)

```
int MainMenu(string user,string pas)
{
    int choice;

    bool a,b;

    a=pass(user,pas);
    b=ReadFromFile(user,pas);
    if(a)
    {
        do{
            cout<<"\nEPELEXE FAINOMRNO"<<endl;
            cout<<"1.GIA FAINOMENO BLEVE"<<endl;
            cout<<"2.GIA FAINOMENO FLAME JET / FLARE"<<endl;
            cout<<"3.GIA FAINOMENO POOL FIRE"<<endl;
            cout<<"4.KATAXWRHSH STOIXEIWN "<<endl;
            cout<<"5.ALAGH      PASSWORD      KAI      USERNAME
ADMINISTRATOR"<<endl;
            cout<<"0.EXIT"<<endl;
            cout<<"DOSE EPILOGI 0..5 : ";
            cin>>choice;
        } while (choice<0 || choice>5);

        return choice;
    }
    else if(b)
    {
        do{
            cout<<"\nEPELEXE FAINOMENO"<<endl;
            cout<<"1.GIA FAINOMENO BLEVE"<<endl;
            cout<<"2.GIA FAINOMENO FLAME JET / FLARE"<<endl;
            cout<<"3.GIA FAINOMENO POOL FIRE"<<endl;
            cout<<"0.EXIT"<<endl;
            cout<<"DOSE EPILOGI 0..3 : ";
            cin>>choice;
        } while (choice<0 || choice>3);

        return choice;
    }
    else throw BXptn("MAIN MENU", BXptn::lathosuser);
}
```

### Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στην συνάρτηση αυτή παίρνω το username και το password από το χρήστη και με τις δυο συναρτήσεις ελέγχω αν είναι σωστά .

- Με τις δυο if ελέγχω αν το username και το password υπάρχουν στο αρχείο μου.αλλιώς βγάζει μήνυμα λάθους
  - Για την πρώτη περίπτωση που ο χρήστης δεν δίνει το σωστό username και το password ή δεν το δίνει καθόλου παρά μόνο το σωστό συνθηματικό υπάρχει μια συνθήκη **do while** η οποία δίνει τη δυνατότητα στο χρήστη να επιλέξει μια από της 6 επιλογές. Η συνθήκη υλοποιήθηκε σωστά και δεν υπάρχει πρόβλημα στην εκτέλεση της.
  - Όμως όταν ο χρήστης δώσει οτιδήποτε άλλο εκτός από αριθμό( γράμμα ή σύμβολο).
- Για την δεύτερη περίπτωση που ο χρήστης δίνει το σωστό username και το password και το σωστό συνθηματικό υπάρχει μια συνθήκη **do while** η οποία δίνει τη δυνατότητα στο χρήστη να επιλέξει μια από της 4 επιλογές. Η συνθήκη υλοποιήθηκε σωστά και δεν υπάρχει πρόβλημα στην εκτέλεση της.
  - Όμως όταν ο χρήστης δώσει οτιδήποτε άλλο εκτός από αριθμό( γράμμα ή σύμβολο).

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 10. Έλεγχος της συνάρτησης ZoneSelect()

```
int ZoneSelect()
{
  int choice;
  do{
    cout<<"\nEPELEXE ZWNH "<<endl;
    cout<<"1.ZWNH I (1500)"<<endl;
    cout<<"2.ZWNH II (450)"<<endl;
    cout<<"3.ZWNH III (170)"<<endl;
    cout<<"4.ZWNH IV (40)"<<endl;
    cout<<"0.EXIT"<<endl;
    cout<<"DOSE EPILOGH (0..4) : ";
    cin>>choice;
  } while(choice<0 || choice>4);

  switch(choice)
  {
    case(0):{ return 0; }
    case(1):{ return 1500;}
    case(2):{ return 450;}
    case(3):{ return 170;}
    case(4):{ return 40;}
  }
  //return;
}
```

### Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

- Στον κώδικα αυτό υπάρχει μια συνθήκη **do while** με την **switch(choice)** η οποία δίνει τη δυνατότητα στο χρήστη να επιλέξει μια από της 5 επιλογές. Η συνθήκη υλοποιήθηκε σωστά και δεν υπάρχει πρόβλημα στην εκτέλεση της.
  - Όμως όταν ο χρήστης δώσει οτιδήποτε άλλο εκτός από αριθμό( γράμμα ή σύμβολο).

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 11. Έλεγχος της συνάρτησης Default(string user,string password)

```
void Default(string user,string password){

int zone;
double Rs,Ts,F,I,L;
double FF,LF;
int choice;
double MR;

double RP,HF,Q,EP,TDU_I1,TDU_I2,TDU_I3,TDU_I4,TDU_I5;
double X1,X2,X3,X4,X5;

string Npass,Nuser;
do{

choice=MainMenu(user,password);

switch(choice)
{
case(1):{
zone=ZoneSelect();
if (zone!=0)
{
Bleve(zone,Rs,Ts,F,I,L);
cout<<"\n FAINOMENO BLEVE "<<endl;
cout<<"AKTINA PURINHS SFAIRAS Rs (m) : "<<Rs<<endl;
cout<<"XRONIKI DIARKEIA BLEVE Ts (sec) : "<<Ts<<endl;
cout<<"POSOSTO EKPEMPOMENHS AKTINOVOLIAS F (%) : "<<F<<endl;
cout<<"THERMIKI AKTINOVOLIA I (W/m2) : "<<I<<endl;
cout<<"APOSTASH PARATHRHRH L (m) : "<<L<<endl<<endl<<endl;
}
break;
}
case(2):{
cout<<"\n FAINOMENO FLAME JET / FLARE"<<endl;
cout<<"DIAROH LOGO DIABROSHS O,17 - 1,00 kgr/sec"<<endl;
cout<<"DIATRHS H MHXANIKH ASTOXIA 1,01 - 1,30 kgr/sec"<<endl;
cout<<" DWSE RYTHMO DIAROHW SE (kgr/sec): ";
cin>>FF;
if(FF>0.17 && FF<=1.0)
{
LF=FlameJet(FF);
}
```

```

        cout<<endl;
        cout<<"\n MHKOS FLOGAS (m) GIA DIAROH LOGO
DIABROSHS:"<<LF<<endl;
    }
    else if(FF>=1.01 && FF<1.30)
    {
        LF=FlameJet(FF);
        cout<<"\n MHKOS FLOGAS (m) GIA DIATRHS H MHXANIKH
ASTOXIA:"<<LF<<endl;
    }
    else throw BXptn("FlameJet",BXptn::ektosoriwn);

    break;
}
case(3):{
        cout<<"DWSE POSOTHTA DIARROHS: ";
        cin>>MR;

        PoolFire(MR,RP,HF,Q,EP,I,TDU_I1,TDU_I2,TDU_I3,TDU_I4,TDU_I5,X
1,X2,X3,X4,X5);

        cout<<"AKTINA        LIMNHS        YGROU        NGL        RP(m)
:_____ "<<RP<<endl;
        cout<<"YPSOS                FLOGAS                HF(m)
:_____ "<<HF<<endl;
        cout<<"EMBADON        LIMNHS        YGROU        NGL        EP(m^2)
:_____ "<<EP<<endl;
        cout<<"OLIKH        THERMOTHTA        EKPOMPHS        Q(KW)
:_____ "<<Q<<endl;
        cout<<"ENTASH        THERMIKHS        AKTINOBOLIAS        I(KW/m^2)
:_____ "<<I<<endl<<endl;

        cout<<"ZONE 1:"<<endl;
        cout<<"ENTASH        THERMIKHS        AKTINOBOLIAS        TDU_I(KW/m^2)
ZONE1:___ "<<TDU_I1<<endl;
        cout<<"APOSTASH        APO TO        KENTRO        THS        FLOGAS        X(m)
ZONE1:_____ "<<X1<<endl<<endl;

        cout<<"ZONE 2:"<<endl;
        cout<<"ENTASH        THERMIKHS        AKTINOBOLIAS        TDU_I(KW/m^2)
ZONE2:___ "<<TDU_I2<<endl;
        cout<<"APOSTASH        APO TO        KENTRO        THS        FLOGAS        X(m)
ZONE2:_____ "<<X2<<endl<<endl;

        cout<<"ZONE 3:"<<endl;
        cout<<"ENTASH        THERMIKHS        AKTINOBOLIAS        TDU_I(KW/m^2)
ZONE3:___ "<<TDU_I3<<endl;

```

```

cout<<"APOSTASH APO TO KENTRO THS FLOGAS X(m)
ZONE3:_____ "<<X3<<endl<<endl;

    cout<<"ZONE 4:"<<endl;
    cout<<"ENTASH THERMIKHS AKTINOBOLIAS TDU_I(KW/m^2)
ZONE4:___ "<<TDU_I4<<endl;
    cout<<"APOSTASH APO TO KENTRO THS FLOGAS X(m)
ZONE4:_____ "<<X4<<endl<<endl;

    cout<<"ZONE 5:"<<endl;
    cout<<"ENTASH THERMIKHS AKTINOBOLIAS TDU_I(KW/m^2)
ZONE5:___ "<<TDU_I5<<endl;
    cout<<"APOSTASH APO TO KENTRO THS FLOGAS X(m)
ZONE5:_____ "<<X5<<endl<<endl;

    break;
}
case(4):{

                WriteToFile();

    break;
}
case(5):{

                cout<<"\nDOSE TO NEO USERNAME :";
    cin>>Nuser;
    cout<<"\nDOSE TO NEO PASSWORD :";
    cin>>Npass;
                changeAdmin(Nuser,Npass);
    cout<<"OI ALAGES KATAXRHTHIKAN "<<endl;
    break;
}
} //switch

} while(choice!=0);

    switch(choice)
    {
        case(0):exit(1);
    }
}

```

### Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στον κώδικα αυτό υπάρχει μια συνθήκη **do while με την switch(choice)** η οποία δίνει τις πέντε περιπτώσεις

- 0) Η οποία κάνει exit
- 1) Με την συνάρτηση ZoneSelect() διαλέγουμε ζώνη ελέγχοντας με μια if δόθηκε ζώνη 0 για να κάνει exit και με την συνάρτηση bleve υπολογίζονται τα αποτελέσματα.
- 2) Έχουμε μια if else που ελέγχει τα όρια της διαρροής που δίνει ο χρήστης.  
Τα οποία όρια όμως δεν ελεγχοντε σωστα :
  - Για 0.17 false(εμφάνιση MainMenu)
  - Για 0.30 false(εμφάνιση MainMenu)
  - Για >0.17 false(εμφάνιση MainMenu)
  - Για <1.31 false(εμφάνιση MainMenu)
  - Οτιδήποτε άλλο εκτός από αριθμό( γράμμα ή σύμβολο>false.
- 3) Ζητάει από το χρήστη ποσότητα διαρροής
  - Δεν ελέγχει καθόλου τα όρια της ποσότητας και με την συνάρτηση pool fire υπολογίζει τα αποτελέσματα.
- 4) Καλεί την συνάρτηση writeToFile για να γίνει καταχώρηση των στοιχείων
- 5) Ζητάει νέο username και password και με την συνάρτηση changeAdmin τα αλλάζει. Αλλά δεν ελέγχονται καθόλου τα στοιχεία που δίνει ο χρήστης. Θα έπρεπε να ελέγχονται τα εξής:
  - Για το username να δίνονται μόνο χαρακτήρες (όχι αριθμοί ή σύμβολα), να μην υπάρχουν κενά μεταξύ των χαρακτήρων και το μήκος να είναι από 3 έως 10.
  - Για το password να δίνονται από 1 εως 6 χαρακτήρες και από 1 έως 6 αριθμούς (όχι σύμβολα), να μην υπάρχουν κενά μεταξύ τους και το μήκος να είναι από 3 έως 7

### **Αντιστοιχος κώδικας :**

```
case(3){
```

```
    cout<<"DWSE POSOTHTA DIARROHS: ";  
    cin>>MR;
```

```
PoolFire(MR,RP,HF,Q,EP,I,TDU_I1,TDU_I2,TDU_I3,TDU_I4,TDU_I5,X1,X2,X3,X4,X5);
```

```
case(5){
```

```
    cout<<"\nDOSE TO NEO USERNAME :";  
    cin>>Nuser;  
    cout<<"\nDOSE TO NEO PASSWORD :";  
    cin>>Npass;  
    changeAdmin(Nuser,Npass);  
    cout<<"OI ALAGES KATAXWRHTHIKAN " <<endl;  
    break;  
}
```

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 12. Έλεγχος της συνάρτησης main()

```
main() {
    string user,pas;
    int c=0;
    int f;

    do
    {
        try{
            cout<<"DWSE USERNAME: ";
            cin>>user;
            cout<<"\nDWSE PASSWORD: ";
            cin>>pas;
            Default(user,pas);
        }
        catch(BXptn x)
        {
            switch(x.getEidos())
            {
                case (BXptn::ektosoriwn):
                {
                    cout<<x.getOnoma()<<": TIMH EKTOS ORIWN STO FAINOMENO
FLAME JET / FLARE"<<endl;
                    break;
                }
                case (BXptn::maxstring):
                {
                    cout<<x.getOnoma()<<": POLY MEGALO ONOMA, EPWNYMO,
USERNAME H PASSWORD"<<endl;
                    break;
                }
                case (BXptn::lathosuser):
                {
                    c++;
                    cout<<x.getOnoma()<<":      CONNECTION      FAILED:      MH
KATAXWRHMENOS XRHSTHS"<<endl;
                    break;
                }
            }
        }
    }while(c<3);
}
```

## Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

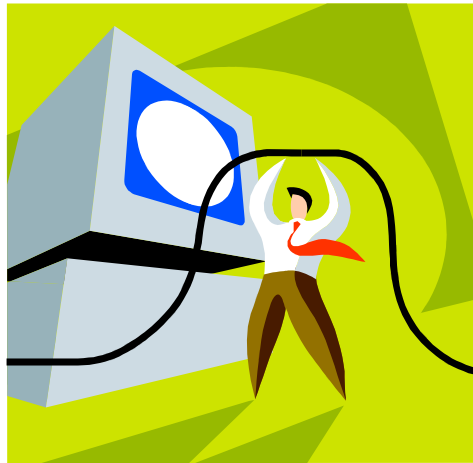
## Παρατηρήσεις:

- Στην main έχουμε μια do while η οποία δίνει τη δυνατότητα στο χρήστη να δώσει μέχρι τρεις φορές username και password μέχρι να δώσει το σωστό.
- Μετά με στη try παίρνει το username και password και καλεί την συνάρτηση default .
- Και τέλος ελέγχει τις περιπτώσεις λάθους
  - Για την περίπτωση που είναι εκτός ορίων αλλά μόνο για την flame jet και όχι για το pool fire
  - Για την περίπτωση μεγάλου string (Επρεπε να γίνετε ξεχωριστός έλεγχος για μεγάλο όνομα, επώνυμο,username, password)
  - Για την περίπτωση λάθους user.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

# Γ' Έλεγχος

## Κλειστού Κουτιού



Ελεγκτής Α' : Υφαντής Γεώργιος  
Ελεγκτής Β' : Αναστασιάδου Έλενα

## ΗΜΕΡΟΛΟΓΙΟ ΕΛΕΓΚΤΗ: 21 Μαΐου 2004

Ver 1.0c

### Ελέγχθηκαν τα αρχεία :

- Blev2.cpp
- 16\_running.cpp
- ScNGL.cpp

### και οι συναρτήσεις :

- Blev
- FlameJet
- MainMenu(a)
- MainMenu(b)
- ZoneSelect
- Default
- changeAdmin
- pass
- ReadFromFile
- WriteToFile (η παλιά void ReadFromFile())
- PoolFire
- Class BXrtn (κλάση εξαιρέσεων)
- DeleteUser

(Ο έλεγχος αρχίζει με τις συναρτήσεις των παλιών αρχείων όπως είναι πλέον διορθωμένες στα πιο πρόσφατα αρχεία. Δηλαδή σε κάθε αρχείο ελέγχουμε πλέον μόνο τις καινούργιες συναρτήσεις και απ' τα παλιά αρχεία τις αλλαγές που χουν γίνει στο πιο πρόσφατο αρχείο όπου μας έχει δωθεί απ' τους προγραμματιστές)



### Στο αρχείο bleve2.cpp

#### Συνάρτηση MainMenu(a), MainMenu(b) :

Στην συνάρτηση MainMenu(a) είναι το μενού επιλογών των administrators και MainMenu(b) είναι το μενού των απλών χρηστών. Οι τιμές που θα δοθούν στον αναλυτή για την εξέταση ορίζονται από τον χρήστη του λογισμικού και είναι αυστηρά καθορισμένες απ' τις επιλογές που καθορίζονται απ' το λογισμικό. Έτσι λοιπόν, οι τιμές που θα εισάγονται δεν υπάρχει περίπτωση να δημιουργήσουν εικασίες λανθασμένων τιμών όπως για παράδειγμα:

- Καμία τιμή (Enter)
- Γράμματα ή άλλοι χαρακτήρες (π.χ. @,#,"δ,a,?,B )
- Κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- Μεγάλο πλήθος ακεραίων που δέχεται
- Σωστή ενημέρωση χρήστη
- Μη ακέραια τιμή
- Αρνητική τιμή
- Τιμές εκτός των ορίων

Εδώ θα ελέγξουμε με τις επιλογές που έχουμε ως απλοί χρήστες <MainMenu(a)> όπως κάναμε και στον 1<sup>ο</sup> έλεγχο όπου έχουμε να εισάγουμε τιμές από 0 έως 3 και για την περίπτωση των administrator <MainMenu(a)> όπου έχουμε άλλες 2 επιλογές για αλλαγή καταχώρησης στοιχείων και αλλαγή κωδικών πρόσβασης των administrators, θα ελέγξουμε παρακάτω με τις αντίστοιχες συναρτήσεις τους.

## Παρατηρήσεις:

Έλεγχος συνάρτησης int MainMenu() --> **MainMenu(a) → Επιλογή φαινομένου**

### Ισοδύναμες κλάσεις τιμών :

- ◆ < 0 (Άκυρες κλάσεις)
- ◆ > 3 (Άκυρες κλάσεις)
- ◆  $0 \leq \text{τιμή} \leq 3$  (Έγκυρες κλάσεις)

### Οριακές τιμές :

- ◆ 0.1, -0.1, 0.0 (Οριακές τιμές)
- ◆ 1.1, 0.9, 1.0 (Οριακές τιμές)
- ◆ 2.1, 2.0, 1.9 (Οριακές τιμές)
- ◆ 3.1, 3.0, 2.9 (Οριακές τιμές)

### Εικασίες τιμών :

- ◆ Καμία τιμή (Enter)
- ◆ Μεγάλη τιμή (πχ 10000)
- ◆ Αρνητική τιμή (πχ -254)
- ◆ Χαρακτήρας (πχ α, s, %)
- ◆ Κενό

Ισοδύναμες κλάσεις τιμών (τιμές εισόδου)	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
0 (έγκυρη κλάση)	Έξοδος 0	Έξοδος 0	✓
1(έγκυρη κλάση)	ZoneSelect 1	ZoneSelect 1	✓
2(έγκυρη κλάση)	FlameJet 2	FlameJet 2	✓
3(έγκυρη κλάση)	PoolFire 3	PoolFire 3	✓
-5(άκυρη κλάση)	MainMenu	Εμφάνιση μηνύματος λάθους & επιστροφή στην MainMenu	✗
5(άκυρη κλάση)	MainMenu	Εμφάνιση μηνύματος λάθους & επιστροφή στην MainMenu	✗
Οριακές τιμές	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
0.0	Έξοδος 0	Έξοδος 0	✓
0.1	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
- 0.1	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
1.0	Λανθασμένη ενέργεια(περνάει το 0 ως τιμή στην επόμενη διαδικασία)	Εμφάνιση μηνύματος λάθους	✗
1.1	Λανθασμένη ενέργεια(περνάει το 1 ως τιμή στην επόμενη διαδικασία)	Εμφάνιση μηνύματος λάθους	✗
0.9	Έξοδος	Εμφάνιση μηνύματος λάθους	✗
2.0	Λανθασμένη ενέργεια(περνάει το 0 ως τιμή στην επόμενη διαδικασία) και ζητά κωδικούς πρόσβασης	Εμφάνιση μηνύματος λάθους και συνέχεια απ' την συγκεκριμένη επιλογή	✗
2.1	Λανθασμένη ενέργεια(περνάει το 1 ως τιμή στην επόμενη διαδικασία) και ζητά κωδικούς πρόσβασης	Εμφάνιση μηνύματος λάθους και συνέχεια απ' την συγκεκριμένη επιλογή	✗
1.9	Λανθασμένη ενέργεια(περνάει το 9 ως τιμή στην επόμενη διαδικασία)	Εμφάνιση μηνύματος λάθους	✗
3.0	Λανθασμένη ενέργεια(περνάει το 0 ως τιμή στην επόμενη διαδικασία)	Εμφάνιση μηνύματος λάθους	✗
3.1	Λανθασμένη ενέργεια(περνάει το 1 ως τιμή στην επόμενη διαδικασία)	Εμφάνιση μηνύματος λάθους	✗
2.9	Λανθασμένη ενέργεια(περνάει το 9 ως τιμή στην επόμενη διαδικασία)	Εμφάνιση μηνύματος λάθους	✗

Εικασίες τιμών	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
Καμία τιμή (Enter)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×
1000 (Μεγάλη τιμή)	Επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	×
- 254 (Αρνητική τιμή)	Επιστροφή στο MainMenu	Εμφάνιση μηνύματος λάθους	×
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
_ (κενο)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ δεν παίρνει ως τιμή χαρακτήρες
- ◆ δεν παίρνει ως τιμή δεκαδικούς αριθμούς
- ◆ δεν παίρνει ως τιμή αρνητικούς αριθμούς
- ◆ παίρνει τιμές από 0 έως 3 μόνο

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν αρκετά λάθη, από την εισαγωγή τιμών εκτός ορίων, μη επιτρεπόμενων τιμών (όπως για παράδειγμα χαρακτήρες, αρνητικοί αριθμοί, δεκαδικά ψηφία κλπ). Υπάρχουν όμως κάποιες παρατηρήσεις που η εφαρμογή τους είναι απαραίτητη σε κάποια σημεία της μεθόδου αυτά είναι:

- Κατά την εισαγωγή των τιμών απ'τον χρήστη, ενώ υπάρχει κλάση εξαιρέσεων για τον πλήρη έλεγχο των τιμών που εισάγονται, δεν εκτελείται ποτε.
- Είναι προτεινόμενο να δημιουργηθούν κάποιοι έλεγχοι για όλες αυτές τις περιπτώσεις σφαλμάτων που προκύπτουν και να συμπεριληφθούν στην κλάση εξαιρέσεων.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 3-λάθη 21)

## Συνάρτηση Blevе, ZoneSelect και Default:

Στην συνέχεια, απ' την επιλογή του MainMenu, πληκτρολογώντας 1 για το φαινόμενο Blevе, ο χρήστης, καλεί στην ουσία την συνάρτηση *void Default()* όπου η συνάρτηση αυτή καλεί και τις άλλες δύο την *void Blevе(int zone, double &Rs, double &Ts, double &F, double &l, double &L)* και την *int ZoneSelect()*, όπου η συνάρτηση Blevе, υπολογίζει από προκαθορισμένους τύπους α) την ακτίνα πύρινης σφαίρας, β) την χρονική διάρκεια του φαινομένου, γ) το ποσοστό εκπεμπόμενης ακτινοβολίας, δ) την θερμική ακτινοβολία και ε) την απόσταση του παρατηρητή. Εδώ ο χρήστης έχει να εισάγει ποια απ' τις προτεινόμενες τέσσερις ζώνες επιθυμεί να ελέγξει. Πληκτρολογώντας και την επιθυμητή ζώνη, το λογισμικό πρέπει να του εμφανίσει κάποια στοιχεία τα οποία θα ελεγχθούν παρακάτω. Στην ουσία λοιπόν, θα ελέγξουμε την εισαγωγή τιμών μέσω της συνάρτησης *ZoneSelect* και την εγκυρότητα των τιμών που επιστρέφει η συνάρτηση *Blevе*. Έτσι λοιπόν, οι τιμές που θα εισάγονται δεν υπάρχει περίπτωση να δημιουργήσουν εικασίες λανθασμένων τιμών όπως για παράδειγμα:

- ◆ Καμία τιμή (Enter)
- ◆ Γράμματα ή άλλοι χαρακτήρες (π.χ. @, #, ", δ, a, ?, B )
- ◆ Κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Μεγάλο πλήθος ακεραίων που δέχεται
- ◆ Σωστή ενημέρωση χρήστη
- ◆ Μη ακέραια τιμή
- ◆ Αρνητική τιμή
- ◆ Τιμές εκτός των ορίων

## Παρατηρήσεις:

Έλεγχος συναρτήσεων *int ZoneSelect()* → **ZoneSelect → Επιλογή ζώνης**  
*void Default(string user, string password)* → **case(3) → Εισαγωγή Ποσότητας Διαρροής**

### ■ int ZoneSelect()

#### Ισοδύναμες κλάσεις τιμών :

- ◆ < 0 (Άκυρες κλάσεις)
- ◆ > 4 (Άκυρες κλάσεις)
- ◆  $0 \leq \text{τιμή} \leq 4$  (Έγκυρες κλάσεις)

#### Οριακές τιμές :

- ◆ 0.1, -0.1, 0.0 (Οριακές τιμές)
- ◆ 1.1, 0.9, 1.0 (Οριακές τιμές)
- ◆ 2.1, 2.0, 1.9 (Οριακές τιμές)
- ◆ 3.1, 3.0, 2.9 (Οριακές τιμές)
- ◆ 4.1, 4.0, 3.9 (Οριακές τιμές)

#### Εικασίες τιμών :

- ◆ Καμία τιμή (Enter)
- ◆ Μεγάλη τιμή (πχ 10000)
- ◆ Αρνητική τιμή (πχ -254)
- ◆ Χαρακτήρας (πχ α, s, %)
- ◆ Κενό

<b>Ισοδύναμες κλάσεις τιμών (τιμές εισόδου)</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0(έγκυρη κλάση)	Επιστροφή στη MainMenu 0	Επιστροφή στη MainMenu 0	✓
1(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 1	Εμφάνιση αποτελεσμάτων 1	✓
2(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 2	Εμφάνιση αποτελεσμάτων 2	✓
3(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 3	Εμφάνιση αποτελεσμάτων 3	✓
4(έγκυρη κλάση)	Εμφάνιση αποτελεσμάτων 4	Εμφάνιση αποτελεσμάτων 4	✓
-5(άκυρη κλάση)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	✗
5(άκυρη κλάση)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	✗
<b>Οριακές τιμές</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0	Επιστροφή στη MainMenu 0	Επιστροφή στη MainMenu 0	✓
0.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
- 0.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
1.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
1.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
0.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
2.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
2.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
1.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
2.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
4.0	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
4.1	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
3.9	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗

Εικασίες τιμών	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
Καμία τιμή (Enter)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×
1000 (Μεγάλη τιμή)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	×
- 254 (Αρνητική τιμή)	ZoneSelect	Εμφάνιση μηνύματος λάθους & επιστροφή στην ZoneSelect	×
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×
_ (κενο)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	×
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	×

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ δεν παίρνει ως τιμή χαρακτήρες
- ◆ δεν παίρνει ως τιμή δεκαδικούς αριθμούς
- ◆ δεν παίρνει ως τιμή αρνητικούς αριθμούς
- ◆ παίρνει τιμές από 0 έως 4 μόνο

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν αρκετά λάθη, από την εισαγωγή τιμών εκτός ορίων, μη επιτρεπόμενων τιμών (όπως για παράδειγμα χαρακτήρες, αρνητικοί αριθμοί, δεκαδικά ψηφία κλπ). Υπάρχουν όμως κάποιες παρατηρήσεις που η εφαρμογή τους είναι απαραίτητη σε κάποια σημεία της μεθόδου αυτά είναι:

- Κατά την εισαγωγή των τιμών απ' τον χρήστη, δεν υπάρχει η αντίστοιχη κλάση εξαιρέσεων για τον πλήρη έλεγχο των τιμών που εισάγονται.
- Είναι προτεινόμενο να δημιουργηθούν κάποιοι έλεγχοι για όλες αυτές τις περιπτώσεις σφαλμάτων που προκύπτουν.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2-λάθη 24)

■ void Default(string user,string password) → Case(3)

Ισοδύναμες κλάσεις τιμών :

- ◆ < 150000 (Άκυρες κλάσεις)
- ◆ > 650000 (Άκυρες κλάσεις)
- ◆ 150000 ≤ τιμή ≤ 650000 (Έγκυρες κλάσεις)

Οριακές τιμές :

- ◆ 149999, 150001, 150000 (Οριακές τιμές)
- ◆ 649999, 650001, 650000 (Οριακές τιμές)

Εικασίες τιμών :

- ◆ Καμία τιμή (Enter)
- ◆ Μεγάλη τιμή (πχ 1000000)
- ◆ Αρνητική τιμή (πχ -254)
- ◆ Χαρακτήρας (πχ α, s, %)
- ◆ Μηδενική τιμή (0)
- ◆ Κενό

Ισοδύναμες κλάσεις τιμών (τιμές εισόδου)	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000 (έγκυρη κλάση)	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	✓
150000 (έγκυρη κλάση)	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	✓
650000 (έγκυρη κλάση)	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	✓
500000 (έγκυρη κλάση)	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	✓
-5(άκυρη κλάση)	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✓
1000000 (άκυρη κλάση)	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✓
Οριακές τιμές	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
149999	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✓
150001	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	✓
150000	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	✓
649999	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	Συνέχεια στην επόμενη διεργασία (επιλογή ζώνης)	✓
650001	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✓
650000	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✓

Εικασίες τιμών	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
Καμία τιμή (Enter)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	✗
1000000 (Μεγάλη τιμή)	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✓
- 254 (Αρνητική τιμή)	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✗
_ (κενο)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
Μηδενική τιμή (0)	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✓

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ δεν παίρνει ως τιμή χαρακτήρες
- ◆ δεν παίρνει ως τιμή δεκαδικούς αριθμούς
- ◆ δεν παίρνει ως τιμή αρνητικούς αριθμούς
- ◆ παίρνει τιμές από 150000 έως 650000 μόνο

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν ελάχιστα λάθη και σχετικά ασήμαντα σε σχέση με προηγούμενους ελέγχους. Υπάρχουν όμως κάποιες παρατηρήσεις που η εφαρμογή τους είναι απαραίτητη σε κάποια σημεία της μεθόδου αυτά είναι:

- Είναι προτεινόμενο να δημιουργηθούν κάποιοι επιπλέον έλεγχοι για όλες αυτές τις περιπτώσεις σφαλμάτων που προκύπτουν.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 1-λάθη 6)

Έλεγχος συνάρτησης void Bleve(int zone, double &Rs, double &Ts, double &F, double &I, double &L) --> **Bleve → Φαινόμενο Bleve**

**Έλεγχος συνάρτησης Ακτίνας Πύρινης Σφαίρας :**

Τύπος :  $R_S = 0.5 * 6.48 * M_R^{0.325}$  , όπου  $R_S$  η ακτίνα πύρινης σφαίρας (m) και  $M_R$  η ολική ποσότητα εκροής (kgr)

Είσοδος : Ολική ποσότητα εκροής  $M_R$

Έξοδος : Ακτίνα Πύρινης σφαίρας  $R_S$

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Ολική ποσότητα εκροής  $M_R$

**Ζώνη I (1500 m):**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	155.874	155.874	✓
650000	251.039	251.039	✓
500000	230.521	230.521	✓

**Ζώνη II (450 m):**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	155.874	155.874	✓
650000	251.039	251.039	✓
500000	230.521	230.521	✓

**Ζώνη III (170 m):**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	155.874	155.874	✓
650000	251.039	251.039	✓
500000	230.521	230.521	✓

### Ζώνη IV (40 m):

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	155.874	155.874	✓
650000	251.039	251.039	✓
500000	230.521	230.521	✓

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της Ολικής ποσότητας εκροής
- ◆ Υπολογισμός Ακτίνας Πύρινης σφαίρας
- ◆ Εγκυρότητα του τύπου υπολογισμού της ακτίνας πύρινης σφαίρας

Ελέγχθηκαν οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο δεν προέκυψαν κάποια λάθη, από την εισαγωγή τιμών, ώστε να οφείλονται σε λανθασμένο υπολογισμό του συγκεκριμένου τύπου.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 0 – λάθη 0)

**Έλεγχος συνάρτησης Χρονικής Διάρκειας Bleve :**

Τύπος :  $T_s = 0.852 * M_R^{0.26}$  , όπου  $T_s$  η χρονική διάρκεια BLEVE (sec) και  $M_R$  η ολική ποσότητα εκροής (kg)

Είσοδος : Ολική ποσότητα εκροής  $M_R$

Έξοδος : Χρονική διάρκεια Bleve

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Ολική ποσότητα εκροής  $M_R$

**Ζώνη I (1500 m):**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	18.8896	18.8896	✓
650000	27.6565	27.6565	✓
500000	25.8328	25.8328	✓

**Ζώνη II (450 m):**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	18.8896	18.8896	✓
650000	27.6565	27.6565	✓
500000	25.8328	25.8328	✓

**Ζώνη III (170 m):**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	18.8896	18.8896	✓
650000	27.6565	27.6565	✓
500000	25.8328	25.8328	✓

**Ζώνη IV (40 m):**

<b>Είσοδος</b>	<b>Έξοδος</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
150000	18.8896	18.8896	✓
650000	27.6565	27.6565	✓
500000	25.8328	25.8328	✓

**Παρατηρήσεις.**

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της Ολικής ποσότητας εκροής
- ◆ Υπολογισμός Χρονικής διάρκειας Bleve
- ◆ Εγκυρότητα του τύπου υπολογισμού της Χρονικής διάρκειας Bleve

Ελέγχθηκαν οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο δεν προέκυψαν κάποια λάθη, από την εισαγωγή τιμών, ώστε να οφείλονται σε λανθασμένο υπολογισμό του συγκεκριμένου τύπου.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 0 – λάθη 0)

**Έλεγχος συνάρτησης Ποσοστού Εκπεμπόμενης Ακτινοβολίας :**

Τύπος :  $F = 0.27 * P^{0.32}$  , όπου F το ποσοστό εκπεμπόμενης ακτινοβολίας (%) και P η πίεση της δεξαμενής (Μρα)

Είσοδος : Πίεση δεξαμενής P

Έξοδος : Ποσοστό Εκπεμπόμενης Ακτινοβολίας

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Πίεση δεξαμενής P

**Ζώνη I (1500 m):**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	0.12923		×
650000	0.12923		×
500000	0.12923		×

**Ζώνη II (450 m):**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	0.12923		×
650000	0.12923		×
500000	0.12923		×

**Ζώνη III (170 m):**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	0.12923		×
650000	0.12923		×
500000	0.12923		×

### Ζώνη IV (40 m):

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	0.12923		×
650000	0.12923		×
500000	0.12923		×

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της Πίεση δεξαμενής P (όπου στο πρόγραμμα δεν ορίζεται απ' το χρήστη, ενώ θα πρέπει κανονικά)
- ◆ Υπολογισμός Ποσοστού Εκπεμπόμενης Ακτινοβολίας
- ◆ Εγκυρότητα του τύπου υπολογισμού του Ποσοστού Εκπεμπόμενης Ακτινοβολίας

Ελέγχθηκαν οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο έχουμε να κάνουμε τις εξής παρατηρήσεις:

- Οι προγραμματιστές θα έπρεπε να δημιουργήσουν μια επιλογή εισαγωγής της πίεσης της δεξαμενής για τον υπολογισμό του ποσοστού εκπεμπόμενης ακτινοβολίας, έτσι ώστε να μην θεωρείται σταθερό και να μην προκύπτουν συνέχεια σταθερά αποτελέσματα, όπως προκύπτουν τώρα.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 1 – λάθη 4/ζώνη)

### Έλεγχος συνάρτησης Θερμικής Ακτινοβολίας :

Τύπος :  $I = \left( \frac{\zeta \omega \nu \eta}{T_s} \right)^{0,75}$  , όπου I η θερμική ακτινοβολία, ζώνη η ζώνη που δηλώνουμε  
και T<sub>s</sub> η χρονική διάρκεια BLEVE

Είσοδος : Ακτίνα Ζώνης

Έξοδος : Θερμική Ακτινοβολία I

Σταθερές παράμετροι : T<sub>s</sub> η χρονική διάρκεια BLEVE

Μεταβλητές Παράμετροι : Ακτίνα Ζώνης και χρονική διάρκεια BLEVE T<sub>s</sub>

#### Ζώνη I (1500 m):

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	26.6012	26.6012	✓
650000	19.9858	19.9858	✓
500000	21.0349	21.0349	✓

#### Ζώνη II (450 m):

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	10.7831	10.7831	✓
650000	8.10143	8.10143	✓
500000	8.52669	8.52669	✓

#### Ζώνη III (170 m):

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	5.196	5.196	✓
650000	3.90381	3.90381	✓
500000	4.10873	4.10873	✓

### Ζώνη IV (40 m):

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	1.75541	1.75541	✓
650000	1.31886	1.31886	✓
500000	1.38809	1.38809	✓

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της Ακτίνας ζώνης
- ◆ Υπολογισμός : Θερμικής Ακτινοβολίας I
- ◆ Εγκυρότητα του τύπου υπολογισμού της Θερμικής Ακτινοβολίας I

Ελέγχθηκαν οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο δεν προέκυψαν κάποια λάθη, από την εισαγωγή τιμών, ώστε να οφείλονται σε λανθασμένο υπολογισμό του συγκεκριμένου τύπου.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 0 – λάθη 0)

### Έλεγχος συνάρτησης Απόστασης Παρατηρητή :

$$\text{Τύπος : } L = \sqrt{\frac{F * M_R * E_B * T_A}{4 * \pi * I * 1000 * T_S}}, \text{ όπου } F \text{ το ποσοστό εκπεμπόμενης ακτινοβολίας,}$$

$M_R$  η ολική ποσότητα εκροής,  $E_B$  η ενέργεια καύσης,  $T_A$  ο ατμοσφαιρικός συντελεστής μετάδοσης  $\cong 0,7$ ,  $T_S$  η χρονική διάρκεια BLEVE,  $I$  η θερμική ακτινοβολία και  $L$  η απόσταση του παρατηρητή.

Είσοδος :  $M_R$  η ολική ποσότητα εκροής

Έξοδος : Απόσταση παρατηρητή  $L$

Σταθερές παράμετροι :  $\pi$ ,  $E_B$  η ενέργεια καύσης,  $T_A$  ο ατμοσφαιρικός συντελεστής μετάδοσης,  $F$  το ποσοστό εκπεμπόμενης ακτινοβολίας,  $I$  η θερμική ακτινοβολία,  $T_S$  η χρονική διάρκεια BLEVE

Μεταβλητές Παράμετροι :  $M_R$  η ολική ποσότητα εκροής

#### Ζώνη I (1500 m):

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	310.968	310.968	✓
650000	617.206	617.206	✓
500000	545.961	545.961	✓

#### Ζώνη II (450 m):

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	488.422	488.422	✓
650000	969.415	969.415	✓
500000	857.514	857.514	✓

#### Ζώνη III (170 m):

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	703.609	703.609	✓
650000	1396.52	1396.52	✓
500000	1235.31	1235.31	✓

### Ζώνη IV (40 m):

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
150000	1210.54	1210.54	✓
650000	2402.66	2402.66	✓
500000	2125.31	2125.31	✓

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της  $M_R$  η ολική ποσότητα εκροής
- ◆ Υπολογισμός Απόστασης παρατηρητή L
- ◆ Εγκυρότητα του τύπου υπολογισμού της Απόστασης παρατηρητή L

Ελέγχθηκαν οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο δεν προέκυψαν κάποια λάθη, από την εισαγωγή τιμών, ώστε να οφείλονται σε λανθασμένο υπολογισμό του συγκεκριμένου τύπου.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 0 – λάθη 0)

## Συνάρτηση FlameJet και Default:

Στην συνέχεια, απ' την επιλογή του MainMenu, πληκτρολογώντας 2 για το φαινόμενο FlameJet, ο χρήστης, καλεί στην ουσία την συνάρτηση *void Default()* όπου η συνάρτηση αυτή καλεί την *double FlameJet(double FF)*, όπου η συνάρτηση *FlameJet*, υπολογίζει από προκαθορισμένο τύπο το μήκος της φλόγας για διαρροή λόγω διάβρωσης. Εδώ ο χρήστης έχει να εισάγει το ρυθμό διαρροής. Πληκτρολογώντας και τον επιθυμητό ρυθμό διαρροής, το λογισμικό πρέπει να του εμφανίσει το μήκος της φλόγας που θα προκύψει. Στην ουσία λοιπόν, θα ελέγξουμε την εισαγωγή τιμών μέσω της συνάρτησης *Default* για την εισαγωγή του ρυθμού διάβρωσης και την εγκυρότητα των τιμών που επιστρέφει η συνάρτηση *FlameJet*. Έτσι λοιπόν, οι τιμές που θα εισάγονται δεν υπάρχει περίπτωση να δημιουργήσουν εικασίες λανθασμένων τιμών όπως για παράδειγμα:

- ◆ Καμία τιμή (Enter)
- ◆ Γράμματα ή άλλοι χαρακτήρες (π.χ. @,#,"δ,a,?,B )
- ◆ Κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Μεγάλο πλήθος ακεραίων που δέχεται
- ◆ Σωστή ενημέρωση χρήστη
- ◆ Αρνητική τιμή
- ◆ Τιμές εκτός των ορίων

## Παρατηρήσεις:

Έλεγχος συνάρτησης `int Default(), case(2)` → **Επιλογή ρυθμού διαρροής**

### Ισοδύναμες κλάσεις τιμών :

- ◆  $< 0,17$  (Άκυρες κλάσεις)
- ◆  $> 1,30$  (Άκυρες κλάσεις)
- ◆  $0,17 \leq \text{τιμή} \leq 1,30$  (Έγκυρες κλάσεις)

### Οριακές τιμές :

- ◆ 0.17, 0.16, 0.18 (Οριακές τιμές)
- ◆ 1.30, 1.29, 1.31 (Οριακές τιμές)
- ◆ 1.00, 0.99, 1.01 (Οριακές τιμές)
- ◆ 1.01, 1.00, 1.02 (Οριακές τιμές)

### Εικασίες τιμών :

- ◆ Καμία τιμή (Enter)
- ◆ Μεγάλη τιμή (πχ 10000)
- ◆ Αρνητική τιμή (πχ -254)
- ◆ Μηδενική τιμή (0)
- ◆ Χαρακτήρας (πχ α, s, %)
- ◆ Κενό
- ◆ Τυχαίες τιμές μεταξύ των ορίων

<b>Ισοδύναμες κλάσεις τιμών (τιμές εισόδου)</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0.17(έγκυρη κλάση)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση αποτελεσμάτων	✗
1.30(έγκυρη κλάση)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση αποτελεσμάτων	✗
-5(άκυρη κλάση)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην ίδια συνάρτηση για εισαγωγή νέας τιμής	✗
5(άκυρη κλάση)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην ίδια συνάρτηση για εισαγωγή νέας τιμής	✗
<b>Οριακές τιμές</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
0.17	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση αποτελεσμάτων	✗
0.16	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην ίδια συνάρτηση για εισαγωγή νέας τιμής	✗
0.18	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.30	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση αποτελεσμάτων	✗
1.29	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.31	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην ίδια συνάρτηση για εισαγωγή νέας τιμής	✗
1.00	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
0.99	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.01	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.01	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.00	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1.02	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓

<b>Εικασίες τιμών</b>	<b>Αποτέλεσμα Επεξεργασίας Τιμών</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
Καμία τιμή (Enter)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	✗
1000 (Μεγάλη τιμή)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην ίδια συνάρτηση για εισαγωγή νέας τιμής	✗
- 254 (Αρνητική τιμή)	Εμφάνιση λανθασμένων τιμών και ξαναεισαγωγή κωδικών ασφαλείας	Εμφάνιση μηνύματος λάθους & επιστροφή στην ίδια συνάρτηση για εισαγωγή νέας τιμής	✗
a (χαρακτήρας)	Τερματισμός	Εμφάνιση μηνύματος λάθους & επιστροφή στην ίδια συνάρτηση για εισαγωγή νέας τιμής	✗
Φ (χαρακτήρας)	Τερματισμός	Εμφάνιση μηνύματος λάθους & επιστροφή στην ίδια συνάρτηση για εισαγωγή νέας τιμής	✗
\$ (χαρακτήρας)	Τερματισμός	Εμφάνιση μηνύματος λάθους & επιστροφή στην ίδια συνάρτηση για εισαγωγή νέας τιμής	✗
_ (κενο)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	✗
0,53 (Τυχαίες τιμές)	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
1,19 (Τυχαίες τιμές)	Εμφάνιση Αποτελέσματος	Εμφάνιση Αποτελέσματος	✓
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επιστροφή στην ίδια συνάρτηση για εισαγωγή νέας τιμής	✗

## Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ δεν παίρνει ως τιμή χαρακτήρες
- ◆ δεν παίρνει ως τιμή δεκαδικούς αριθμούς με περισσότερα από 2 δεκαδικά ψηφία
- ◆ δεν παίρνει ως τιμή αρνητικούς αριθμούς
- ◆ παίρνει τιμές από 0,17 έως 1,30 μόνο

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν κάποια λάθη, από την εισαγωγή τιμών εκτός ορίων, μη επιτρεπόμενων τιμών (όπως για παράδειγμα χαρακτήρες, αρνητικοί αριθμοί, δεκαδικά ψηφία κλπ). Υπάρχουν όμως κάποιες παρατηρήσεις που η εφαρμογή τους είναι απαραίτητη σε κάποια σημεία της μεθόδου αυτά είναι:

- Δημιουργήθηκε μια κλάση εξαιρέσεων και όντως εμφανίζει μήνυμα λάθους κατά την εισαγωγή λανθασμένων τιμών, όμως εκεί όπου κατά τον έλεγχο βάλουμε επιτυχημένη και αποτυχημένη ενέργεια ταυτόχρονα, απλά σημαίνει ότι η κλάση εξαιρέσεων εμφανίζει το μήνυμα λανθασμένης εισαγωγής τιμών, όμως επιστρέφει αντί για να δώσεις νέα τιμή στην αρχή όπου βάζεις πάλι username και password. Και επιπλέον, γραφικά, ανακατεύεται αυτό που εμφανίζει και το γράφει πάνω σε άλλες γραμμές που έχουν ήδη φραφτεί. Αυτό μόνο προτείνουμε στους προγραμματιστές να διορθώσουν, ώστε μετά την αποτυχημένη προσπάθεια να μπορεί να γίνει πάλι εισαγωγή τιμών και να μην “βγαίνει” ο χρήστης απ’ την συγκεκριμένη επιλογή.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 1-λάθη 16)

## Έλεγχος συνάρτησης double FlameJet(double FF)-->FlameJet→Φαινόμενο FlameJet

### Έλεγχος συνάρτησης Μήκους φλόγας για διαρροή :

Τύπος :  $F_F = 0,0006785 * L_F^{2,5}$  , όπου  $F_F$  η ροή του καυσίμου και  $L_F$  το μήκος της

$$\text{φλόγας, οπότε είναι } L_F = 2,5 \sqrt{\frac{F_F}{0,0006785}}$$

Είσοδος :  $F_F$  η ροή του καυσίμου

Εξοδος : Μήκος φλόγας για διαρροή  $L_F$

Σταθερές παράμετροι : -

Μεταβλητές Παράμετροι : Ρυθμός Διαρροής  $F_F$

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
1,19	19,8426	19,8426	✓
0,59	14,9872	14,9872	✓
0,18	9,32157	9,32157	✓
1,29	20,4935	20,4935	✓
1,00	18,5089	18,5089	✓
1,01	18,5827	18,5827	✓

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή του ρυθμού διαρροής
- ◆ Υπολογισμός του μήκους της πόρινης φλόγας
- ◆ Εγκυρότητα του τύπου υπολογισμού του μήκους της πόρινης φλόγας

Ελέγχθηκαν οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο δεν προέκυψαν κάποια λάθη, από την εισαγωγή τιμών, ώστε να οφείλονται σε λανθασμένο υπολογισμό του συγκεκριμένου τύπου.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 0 – λάθη 0)

## Στο αρχείο 16\_running.cpp

### Συνάρτηση pass, ReadFromFile και class BXptn :

Εδώ έχουμε τις 2 συναρτήσεις:

```
bool pass(string username,string password)
```

```
bool ReadFromFile(string user,string password )
```

και την κλάση εξαιρέσεων:

```
class BXptn
```

Θα ελέγξουμε παρακάτω τον τρόπο που λειτουργούν και που αλληλοσυνεργάζονται μεταξύ τους αυτές οι 2 συναρτήσεις και η 1 κλάση. Αρχικά να πούμε ότι κατά την εκκίνηση του συστήματος, ο χρήστης ανάλογα με το αν είναι administrator ή κοινός χρήστης, καλείται να πληκτρολογήσει ένα password και ένα username. Έτσι αρχίζει λοιπόν η κλήση της συνάρτησης *pass*, η οποία για να ελέγξει αν εισήχθη ο σωστός χρήστης, ελέγχει ένα αρχείο στο σκληρό δίσκο, με την συνάρτηση *ReadFromFile* και ανάλογα αν δεν εισήχθη ο σωστός χρήστης ή δεν υπάρχει ο χρήστης στο αρχείο, καλείται η κλάση εξαιρέσεων και εμφανίζει ένα μήνυμα σφάλματος.

Έτσι λοιπόν, για τον έλεγχο αυτών των συναρτήσεων, έχουμε τους παρακάτω περιορισμούς.

- ◆ Να υπάρχει στο αρχείο ο χρήστης
- ◆ Για εισαγωγή administrator, το username πρέπει να ξεκινά με ^, ενώ το password με ~
- ◆ Να μην υπάρχει κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Να μην είναι πολύ μεγάλο το πλήθος των χαρακτήρων που δέχεται (3 έως 10)
- ◆ Να μην γίνεται εισαγωγή χρήστη με ειδικούς χαρακτήρες, εκτός απ' τους administrator, όπου το username ξεκινά με ^ και το password με ~
- ◆ Να μην υπάρχει χρήστης 2<sup>ος</sup> με τα ίδια username και password
- ◆ Ο administrator μπορεί να θεωρείται και χρήστης, ο κοινός χρήστης όμως δε μπορεί να θεωρείται και administrator
- ◆ Σωστή ενημέρωση χρήστη

## Παρατηρήσεις:

### Έλεγχος συναρτήσεων

`bool pass(string username,string password)`

**Εισαγωγή κωδικών ασφαλείας**

`bool ReadFromFile(string user,string password )`

### Εικασίες τιμών :

- ◆ Ανύπαρκτα username και password
- ◆ username και password σωστά
- ◆ εισαγωγή ως administrator χωρίς όμως ^ και ~
- ◆ Ειδικόί Χαρακτήρες (πχ α, s, %)
- ◆ Μόνο αριθμοί
- ◆ Κενό ανάμεσα στις λέξεις

Για να πραγματοποιηθεί ο απαραίτητος έλεγχος, εισάγω στο αρχείο όπου βρίσκονται οι κωδικοί διάφορους κωδικούς όπως:

#### ***username      password***

- ❖ ^K            ~K    (έγκυρο, administrator)
- ❖ !!            !!    (άκυρο, μόνο ειδικόί χαρακτήρες)
- ❖ 12            34    (άκυρο, μόνο αριθμοί)
- ❖ α 0           ρλ    (άκυρο, κενό μέσα στη λέξη)
- ❖ gioufa       yfgefa (έγκυρο, κοινόί χρήστης)
- ❖ makhs       lamprou (έγκυρο, κοινόί χρήστης)
- ❖ asasasdfasdfaadfasdfasdfadfa adsfadsfasdfadfadfdafasdf (άκυρο,κωδικοί άνω 10 χαρακτήρων)

Εικασίες τιμών Username-Password	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Σχόλιο	Επιτυχία / Αποτυχία
^K ~K	Εισαγωγή	Εισαγωγή	administrator	✓
^K ^K	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Δεν υπάρχει στο αρχείο	✓
~K ~K	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Δεν υπάρχει στο αρχείο	✓
~K ^K	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Δεν υπάρχει στο αρχείο	✓
gioufa yfgefa	Εισαγωγή	Εισαγωγή	Απλός χρήστης	✓
!! !!	Εισαγωγή	Εμφάνιση μηνύματος λάθους	Μόνο ειδικοί χαρακτήρες	✗
12 34	Εισαγωγή	Εμφάνιση μηνύματος λάθους	Μόνο αριθμοί	✗
** α 0 ρλ	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Κενό μεταξύ των χαρακτήρων	✗
makhs lamprou	Εισαγωγή	Εισαγωγή	Απλός χρήστης	✓
asasasdfasdfa adsfadfasdfadfadfadfasdf	Εισαγωγή	Εμφάνιση μηνύματος λάθους	Πολύ μεγάλοι κωδικοί	✗
lewnidas pelopidas	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Δεν υπάρχει στο αρχείο	✓
12345 12345	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Δεν υπάρχει στο αρχείο	✓
KWSTAS KWSTAS	Εμφάνιση μηνύματος λάθους και επανάκληση	Εμφάνιση μηνύματος λάθους και επανάκληση	Δεν υπάρχει στο αρχείο	✓

### Παρατηρήσεις.

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν κάποια λάθη που δημιουργήσαμε εμείς για να δούμε κατά πόσο ανταποκρίνεται το πρόγραμμα στα λάθη και έχουμε να τονίσουμε κάποιες παρατηρήσεις:

- \*\* Εδώ κατά την εισαγωγή, εμφανίζει μόν λάθος, αλλά θεωρώντας τον χαρακτήρα μετά το κενό ως password και όχι ως ενιαίο username
- Μετά την εισαγωγή 3 αποτυχημένων προσπαθειών, τερματίζεται το πρόγραμμα
- Για να ξεκινήσει το πρόγραμμα, θα πρέπει να έχει κάνει εισαγωγή πρώτα ο administrator έναν χρήστη. Δεν μπορεία από μόνος του να δηλώσει νέους κωδικούς και να εισαχθεί μόνος του.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 3-λάθη 4)

## Συνάρτηση changeAdmin :

Εδώ έχουμε την συνάρτηση:

```
void changeAdmin(string NAdmiUser,string NAdmipassword)
```

Αρχικά να πούμε ότι η συνάρτηση αυτή καλείται και εμφανίζεται ως επιλογή μόνο εάν έχει γίνει εισαγωγή ως administrator. Έτσι λοιπόν όταν κληθεί η συγκεκριμένη συνάρτηση, ο administrator έχει δικαίωμα ν'αλλάξει το username και το password και ν'αποθηκευτούν στο αρχείο.βασική προϋπόθεση είναι να ελεγχθεί αν υπάρχουν στο αρχείο ίδιοι κωδικοί και ΜΟΝΟ αν δεν υπάρχουν να γίνει η αλλαγή.

Έτσι λοιπόν, για τον έλεγχο της συνάρτησης,έχουμε τους παρακάτω περιορισμούς.

- ◆ Να μην υπάρχουν ίδιοι κωδικοί στο αρχείο
- ◆ Για εισαγωγή administrator,το username πρέπει να ξεκινά με ^, ενώ το password με ~
- ◆ Να μην υπάρχει κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- ◆ Να μην είναι πολύ μεγάλο το πλήθος των χαρακτήρων που δέχεται
- ◆ Να μην υπάρχει χρήστης 2<sup>ος</sup> με τα ίδια username και password
- ◆ Σωστή ενημέρωση χρήστη

Έλεγχος συνάρτησης `void changeAdmin(string NAdmiUser,string NAdmipassword)`  
→ **changeAdmin → Αλλαγή κωδικών πρόσβασης Administrator**

### Εικασίες τιμών :

- ◆ Εισαγωγή ήδη υπάρχοντων username και password
- ◆ Εισαγωγή κωδικών χωρίς ^ και ~ στην αρχή του κάθε κωδικού
- ◆ Εισαγωγή πολύ μεγάλου πλήθους χαρακτήρων κωδικού (άνω 10 χαρακτήρων)
- ◆ Εισαγωγή μόνο αριθμών

\* Στο αρχείο ήδη υπάρχει καταχωρημένος administrator με username : ^K και password : ~K.

Εικασίες τιμών Username-Password	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Σχόλιο	Επιτυχία / Αποτυχία
^K ~K	Καταχώρηση αλλαγής, Εφάνιση Μηνύματος μη καταχωρημένου χρήστη και εισαγωγή στο πρόγραμμα με username και password	Εμφάνιση μηνύματος λάθους και επανάκληση	Πρέπει χωρίς ειδικούς χαρακτήρες και >3 χαρακτήρων	✗
HHHH HHHH	Καταχώρηση αλλαγής και συνέχεια στην επόμενη διεργασία	Καταχώρηση αλλαγής και συνέχεια στην επόμενη διεργασία	Έγκυρη τιμή	✓
K K	Καταχώρηση αλλαγής και συνέχεια στην επόμενη διεργασία	Εμφάνιση μηνύματος λάθους και επανάκληση	Υπάρχουν ΉΔΗ αυτοί οι κωδικοί πρόσβασης και >3 χαρακτήρων	✗
gioufa yfgefa	Καταχώρηση αλλαγής και τερματισμός του προγράμματος	Εμφάνιση μηνύματος λάθους και επανάκληση	Υπάρχουν ΉΔΗ αυτοί οι κωδικοί πρόσβασης	✗
1111 1111	Καταχώρηση αλλαγής, Εφάνιση Μηνύματος μη καταχωρημένου χρήστη και εισαγωγή στο πρόγραμμα με username και password	Εμφάνιση μηνύματος λάθους και επανάκληση	Πρέπει να μη περιέχει μόνο αριθμούς	✗
manos kwtsos	Καταχώρηση αλλαγής και τερματισμός του προγράμματος	Καταχώρηση αλλαγής και συνέχεια στην επόμενη διεργασία	Έγκυρη τιμή	✗
- ξκα	Καταχώρηση αλλαγής, Εφάνιση Μηνύματος μη καταχωρημένου χρήστη και εισαγωγή στο πρόγραμμα με username και password	Εμφάνιση μηνύματος λάθους και επανάκληση	Πρέπει χωρίς ειδικούς χαρακτήρες και >3 χαρακτήρων	✗
Aa Bb	Καταχώρηση αλλαγής, Εφάνιση Μηνύματος μη καταχωρημένου χρήστη και εισαγωγή στο πρόγραμμα με username και password	Εμφάνιση μηνύματος λάθους και επανάκληση	Πρέπει από 3 έως 10 χαρακτήρες	✗
Saasfasfsssfdf asfasfasfshgfh	Καταχώρηση αλλαγής και τερματισμός του προγράμματος	Εμφάνιση μηνύματος λάθους και επανάκληση	Πρέπει έως 10 χαρακτήρες	✗

### Παρατηρήσεις.

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν κάποια λάθη που δημιουργήσαμε εμείς για να δούμε κατά πόσο ανταποκρίνεται το πρόγραμμα στα λάθη και έχουμε να τονίσουμε κάποιες παρατηρήσεις:

- Κατά τον έλεγχο προέκυψαν κάποια λάθη που προκάλεσαν μη ανταπόκριση του συστήματος και τερματισμό του προγράμματος. Προφανώς θα πρέπει να συμπεριληφθούν τα λάθη στην κλάση εξαιρέσεων και να διορθωθεί η συνάρτηση ώστε να μπορεί να γίνει καταχώρηση των σωστών ενεργειών

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 1-λάθη 8)

## Συνάρτηση WriteToFile:

Εδώ έχουμε την συνάρτηση:  
`void WriteToFile ()`

Και η συνάρτηση αυτή καλείται μόνο όταν γίνει εισαγωγή ως administrator και καλείται για καταχώρηση στοιχείων νέων χρηστών και έλεγχο αν υπάρχουν σ'αρχείο τα στοιχεία αυτά που θα εισαχθούν. Έτσι λοιπόν όταν κληθεί η συγκεκριμένη συνάρτηση, ο administrator έχει δικαίωμα ν'αλλάξει το username και το password και ν'αποθηκευτούν στο αρχείο.βασική προϋπόθεση είναι να ελεγχθεί αν υπάρχουν στο αρχείο ίδιοι κωδικοί και ΜΟΝΟ αν δεν υπάρχουν να γίνει η αλλαγή.

Έτσι λοιπόν, για τον έλεγχο της συνάρτησης,έχουμε τους παρακάτω περιορισμούς.

- Να μην υπάρχουν ίδιοι κωδικοί στο αρχείο
- Να μην υπάρχει κενό στην αρχή ή ενδιάμεσα στους χαρακτήρες
- Το πλήθος των χαρακτήρων που δέχεται να είναι Όνομα<10, Επώνυμο<15, 3<Username<10, 3<Password<7
- Να μην υπάρχει χρήστης 2<sup>ος</sup> με τα ίδια username και password
- Πληκτρολόγηση STOP για τερματισμό καταχώρησης στοιχείων
- Σωστή ενημέρωση χρήστη
- Δεν υπάρχει περιορισμός για κεφαλαία ή πεζούς χαρακτήρες

Έλεγχος συνάρτησης void WriteToFile () --> **WriteToFile→Καταχώρηση Νέου Χρήστη**

### Εικασίες τιμών :

- ◆ Εισαγωγή ήδη υπάρχοντων username και password
- ◆ Εισαγωγή στοιχείων που ξεκινούν με κενό ή έχουν ενδιάμεσα στους χαρακτήρες
- ◆ Λανθασμένη πληκτρολόγηση stop για τερματισμό
- ◆ Εισαγωγή πολύ μεγάλου πλήθους χαρακτήρων κωδικού
- ◆ Εισαγωγή μόνο αριθμών

\* Για να πραγματοποιηθεί ο απαραίτητος έλεγχος, υπάρχουν στο αρχείο όπου βρίσκονται οι κωδικοί τα εξής στοιχεία:

<b>όνομα</b>	<b>επώνυμο</b>	<b>username</b>	<b>password</b>
George	Yfanths	gioyfa	yfgefa
Nikolaos	Bogiatzhs	nibogiat	mplampla
Marianna	Polymerou	mapolyme	mania
Babis	Yfanths	layf14	oxbow

Εικασίες τιμών Όνομα-Επώνυμο- Username-Password	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Σχόλιο	Επιτυχία / Αποτυχία
John Petrou john25 johnie	Καταχώρηση στοιχειων στο αρχείο	Καταχώρηση στοιχειων στο αρχείο	Έγκυρα στοιχεία	✓
George Yfanths gioyfa yfgefa	Καταχώρηση στοιχειων στο αρχείο	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρα στοιχεία (ήδη υπάρχουν)	✗
Manos Kwstas john25 johnie	Καταχώρηση στοιχειων στο αρχείο	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρα στοιχεία (ήδη υπάρχουν το username και το password)	✗
Alekos Alexiou ela_aleko yfgefa	Καταχώρηση στοιχειων στο αρχείο	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρα στοιχεία (ήδη υπάρχει το password)	✗
Mar8a Mar8akiou layf14 crazygirl	Καταχώρηση στοιχειων στο αρχείο	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρα στοιχεία (ήδη υπάρχει το username)	✗
Geo rge Karamhtros ougka ougka	Καταχώρηση στοιχειων στο αρχείο	Εμφάνιση μηνύματος λάθους και επανάκληση	Άκυρα στοιχεία (εχει κενό στο όνομα και το παίρνει ως όνομα: Geo, επώνυμο: rge, user: karamhtros και pass: ougka)	✗
STOP	Έξοδος διεργασίας	Έξοδος διεργασίας	(για έξοδο)	✓
Stop	Καταχώρηση στοιχειων στο αρχείο ως Όνομα	Έξοδος	(για έξοδο)	✗
stop	Έξοδος διεργασίας	Έξοδος διεργασίας	(για έξοδο)	✓
George Yfanths gioyfa yfgefayfgefa	Εμφάνιση μηνύματος λάθους και συνέχεια στο Password και το Username και εμφάνιση γραμμής πάνω σε άλλη	Εμφάνιση μηνύματος λάθους και συνέχεια στην ίδια διεργασία	Άκυρα στοιχεία (μεγάλο password)	✗

### Παρατηρήσεις.

Ελέγχθηκαν αρκετές περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα και κατά τον έλεγχο προέκυψαν κάποια λάθη που δημιουργήσαμε εμείς για να δούμε κατά πόσο ανταποκρίνεται το πρόγραμμα στα λάθη και έχουμε να τονίσουμε κάποιες παρατηρήσεις:

- Κατά τον έλεγχο συμπεράναμε ότι το πρόγραμμα δεν ελέγχει στο αρχείο για ύπαρξη ιδίων στοιχείων και έτσι μπορεί να κάνει εισαγωγή τιμών που ήδη υπάρχουν, ενώ δε θα πρέπει.
- Επίσης, όταν έχει λάθος σε κάποια επιλογή, ζητά πάλι κωδικούς και τα εμφανίζει πάνω σε άλλες γραμμές.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2-λάθη 7)

## Συνάρτηση DeleteUser :

Εδώ έχουμε την συνάρτηση:

```
void DeleteUser(string user,string password)
```

Αρχικά να πούμε ότι η συνάρτηση αυτή καλείται και εμφανίζεται ως επιλογή μόνο εάν έχει γίνει εισαγωγή ως administrator. Έτσι λοιπόν όταν κληθεί η συγκεκριμένη συνάρτηση, ο administrator έχει δικαίωμα να διαγράψει τα στοιχεία κάποιου χρήστη απ' το αρχείο. βασική προϋπόθεση είναι να ελεγχθεί αν υπάρχουν στο αρχείο ίδια στοιχεία και ΜΟΝΟ αν υπάρχουν να γίνει η διαγραφή.

Έτσι λοιπόν, για τον έλεγχο της συνάρτησης,έχουμε τους παρακάτω περιορισμούς.

- ◆ Να υπάρχουν ίδια στοιχεία στο αρχείο
- ◆ Να ΜΗΝ επιτρέπεται διαγραφή χρηστώ όπου το Username ν' αρχίζει με ^ και το Password με ~ (γιατί στη περίπτωση αυτή διαγράφεται ο Administrator και δε μπορεί να ξαναγίνει εισαγωγή στο πρόγραμμα ως Administrator!)
- ◆ Σωστή ενημέρωση χρήστη
- ◆ Να πληρούνται οι προϋποθέσεις που ισχύουν και κατά την εισαγωγή χρήστη (δηλαδή για παράδειγμα, να εμφανίζει μήνυμα λάθους αν πάει ο administrator να διαγράψει κάποιο κωδικό που δεν υπάρχει)

Έλεγχος συνάρτησης `void DeleteUser(string user,string password)` → **DeleteUser → Διαγραφή Χρήστη**

### Εικασίες τιμών :

- ◆ Διαγραφή μη υπάρχόντων username και password
- ◆ Διαγραφή κωδικών χωρίς ^ και ~ στην αρχή του κάθε κωδικού

\* Για να πραγματοποιηθεί ο απαραίτητος έλεγχος, υπάρχουν στο αρχείο όπου βρίσκονται οι κωδικοί τα εξής στοιχεία:

<b>όνομα</b>	<b>επώνυμο</b>	<b>username</b>	<b>password</b>	
kavala	oil	^q	~q	(Administrator)
George	Yfanths	gioufa	yfgefa	
Nikolaos	Bogiatzhs	nibogiat	mplampla	
Marianna	Polymerou	mapolyme	mania	
Babis	Yfanths	layf14	oxbow	

Εικασίες τιμών Username-Password	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Σχόλιο	Επιτυχία / Αποτυχία
^q ~q	Καταχώρηση αλλαγής και τερματισμός του προγράμματος	Εμφάνιση μηνύματος ΑΠΑΓΟΡΕΥΣΗΣ της συγκεκριμένης διαγραφής και επανάκληση	Απαγορεύεται να γίνει διαγραφή κωδικών που ξεκινούν με ^ και ~, γιατί διαγράφεται ο administrator	✗
HHHH HHHH	Εμφάνιση μηνύματος Επιτυχούς Διαγραφής και συνέχεια στην επόμενη διεργασία	Εμφάνιση μηνύματος Μη έγκυρων κωδικών και επανάκληση	Μη Έγκυρη τιμή (δεν υπάρχουν οι συγκεκριμένοι κωδικοί)	✗
K K	Εμφάνιση μηνύματος Επιτυχούς Διαγραφής και συνέχεια στην επόμενη διεργασία	Εμφάνιση μηνύματος Μη έγκυρων κωδικών και επανάκληση	Δεν Υπάρχουν αυτοί οι κωδικοί πρόσβασης και <3 χαρακτήρων	✗
gioufa yfgefa	Εμφάνιση μηνύματος Επιτυχούς Διαγραφής και συνέχεια στην επόμενη διεργασία	Εμφάνιση μηνύματος Επιτυχούς Διαγραφής και συνέχεια στην επόμενη διεργασία	Έγκυρη τιμή (υπάρχουν οι συγκεκριμένοι κωδικοί)	✓ ✗ *
^layf14 ~oxbow	Εμφάνιση μηνύματος Επιτυχούς Διαγραφής και συνέχεια στην επόμενη διεργασία	Εμφάνιση μηνύματος Μη έγκυρων κωδικών και επανάκληση	Μη Έγκυρη τιμή (δεν υπάρχουν οι συγκεκριμένοι κωδικοί, αλλά και να υπήρχαν, δε θα πρεπε να διαγραφεί, με ^ και ~)	✗
manos kwtsos	Εμφάνιση μηνύματος Επιτυχούς Διαγραφής και συνέχεια στην επόμενη διεργασία	Εμφάνιση μηνύματος Μη έγκυρων κωδικών και επανάκληση	Μη Έγκυρη τιμή (δεν υπάρχουν οι συγκεκριμένοι κωδικοί)	✗
layf14 oxbow	Εμφάνιση μηνύματος Επιτυχούς Διαγραφής και συνέχεια στην επόμενη διεργασία	Εμφάνιση μηνύματος Επιτυχούς Διαγραφής και συνέχεια στην επόμενη διεργασία	Έγκυρη τιμή (υπάρχουν οι συγκεκριμένοι κωδικοί)	✓ ✗ *

## Παρατηρήσεις.

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν κάποια λάθη που δημιουργήσαμε εμείς για να δούμε κατά πόσο ανταποκρίνεται το πρόγραμμα στα λάθη και έχουμε να τονίσουμε κάποιες παρατηρήσεις:

- \* Στις περιπτώσεις αυτές, φαινομενικά, εμφανίζεται σα να διαγράφει τα στοιχεία απ' το αρχείο, αλλά στην ουσία, στο αρχείο διαγράφονται ΜΟΝΟ κάποιες λέξεις ή κάποια γράμματα των συγκεκριμένων στοιχείων και άλλα γράφονται στο τέλος του αρχείου. Οπότε, ΕΙΝΑΙ δυνατόν, ενώ έχει γίνει διαγραφή, να μπορεί να ξαναεισέλθει κάποιος με τους συγκεκριμένους κωδικούς που διεγράφησαν. Και είναι ΠΟΛΥ επικίνδυνο, γιατί καταστρέφεται όλο το αρχείο έτσι και δεν μπορούν να χρησιμοποιηθούν οι κωδικοί και των άλλων χρηστών.
- Κατά τον έλεγχο επίσης προέκυψαν κάποια λάθη που προκάλεσαν μη ανταπόκριση του συστήματος και τερματισμό του προγράμματος. Προφανώς θα πρέπει να συμπεριληφθούν τα λάθη στην κλάση εξαιρέσεων και να διορθωθεί η συνάρτηση ώστε να μπορεί να γίνει καταχώρηση των σωστών ενεργειών
- Τέλος, πρέπει να τονίσουμε ότι κατά τον έλεγχο, μπορούσε δε μπορούσε να γίνει διαγραφή, ΠΑΝΤΑ εμφανιζόταν μήνυμα Επιτυχούς Εκτέλεσης. Θα πρέπει οι προγραμματιστές, να συμπεριλάβουν και ελέγχους για τις περιπτώσεις όπου δε επιτρέπεται να γίνει διαγραφή

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 3-λάθη 7)

## Συνάρτηση PoolFire:

Εδώ έχουμε την συνάρτηση:

```
void PoolFire(const double MR,double &RP,double &HF,double &Q,double  
&EP,  
              double &l,double &TDU_I1,double &TDU_I2,double  
&TDU_I3,double &TDU_I4,double  
&TDU_I5,double &X1,double &X2,double &X3,double  
&X4,double &X5)
```

Η συνάρτηση αυτή καλείται μέσω της συνάρτησης default και ανάλογα με την επιλογή που δίνει ο χρήστης, αν κατά την κλήση της επιλογής φαινομένου επιλέξει 3, τότε ο χρήστης εισάγει την ποσότητα διαρροής. Έτσι λοιπόν κατά την κλήση της συνάρτησης PoolFire, γίνεται ο υπολογισμός α) της ακτίνας λίμης υγρού, β) του ύψους της φλόγας, γ) του εμβαδού της λίμνης του υγρού, δ) της ολικής θερμότητας εκπομπής, ε) της έντασης της θερμικής ακτινοβολίας και τέλος για κάθε μια απ' τις 5 προκαθορισμένες ζώνες της έντασης της θερμικής ακτινοβολίας ανά ζώνη και της απόστασης απ' το κέντρο της φλόγας.

Έτσι λοιπόν, θα ελέγξουμε για εικασίες τιμών για την ποσότητα διαρροής που δίνει ο χρήστης, μιας και τα επιτρεπτά όρια τιμών είναι από 150.000 έως 650.000.

Έλεγχος συνάρτησης void PoolFire(const double MR,double  
&RP,double &HF,double &Q,double &EP,double &l,double  
&TDU\_I1,double &TDU\_I2,double &TDU\_I3,double  
&TDU\_I4,double &TDU\_I5,double &X1,double &X2,double  
&X3,double &X4,double &X5)

--> PoolFire → Φαινόμενο PoolFire

### Ισοδύναμες κλάσεις τιμών :

- ◆ < 150000 (Άκυρες κλάσεις)
- ◆ > 650000 (Άκυρες κλάσεις)
- ◆ 150000 ≤ τιμή ≤ 650000 (Έγκυρες κλάσεις)

### Οριακές τιμές :

- ◆ 149999, 150001, 150000 (Οριακές τιμές)
- ◆ 649999, 650001, 650000 (Οριακές τιμές)

### Εικασίες τιμών :

- ◆ Καμία τιμή (Enter)
- ◆ Μεγάλη τιμή (πχ 1000000)
- ◆ Αρνητική τιμή (πχ -254)
- ◆ Χαρακτήρας (πχ α, s, %)
- ◆ Μηδενική τιμή (0)
- ◆ Κενό

Ισοδύναμες κλάσεις τιμών (τιμές εισόδου)	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000 (έγκυρη κλάση)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	✓
150000 (έγκυρη κλάση)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	✓
650000 (έγκυρη κλάση)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	✓
500000 (έγκυρη κλάση)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	✓
-5(άκυρη κλάση)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	✓
1000000 (άκυρη κλάση)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	✓
Οριακές τιμές	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
149999	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
150001	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	✓
150000	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	✓
649999	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	Συνέχεια στην επόμενη διεργασία (εμφάνιση αποτελεσμάτων)	✓
650001	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
650000	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
Εικασίες τιμών	Αποτέλεσμα Επεξεργασίας Τιμών	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
Καμία τιμή (Enter)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	✗
1000000 (Μεγάλη τιμή)	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
- 254 (Αρνητική τιμή)	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✗

\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & συνέχεια στην ίδια διεργασία	✗
_ (κενο)	Καμία ενέργεια	Εμφάνιση μηνύματος εισαγωγής τιμής	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους	✗
Μηδενική τιμή (0)	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ δεν παίρνει ως τιμή χαρακτήρες
- ◆ δεν παίρνει ως τιμή δεκαδικούς αριθμούς
- ◆ δεν παίρνει ως τιμή αρνητικούς αριθμούς
- ◆ παίρνει τιμές από 150000 έως 650000 μόνο

Ελέγχθηκαν όλες οι περιπτώσεις που θα μπορούσαν να δημιουργήσουν πρόβλημα στο συγκεκριμένο κομμάτι κώδικα. Κατά τον έλεγχο προέκυψαν ελάχιστα λάθη και σχετικά ασήμαντα σε σχέση με προηγούμενους ελέγχους. Υπάρχουν όμως κάποιες παρατηρήσεις που η εφαρμογή τους είναι απαραίτητη σε κάποια σημεία της μεθόδου αυτά είναι:

- Είναι προτεινόμενο να δημιουργηθούν κάποιοι επιπλέον έλεγχοι για όλες αυτές τις περιπτώσεις σφαλμάτων που προκύπτουν.

### **ΕΛΕΓΧΟΣ ΑΝΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 1-λάθη 6)

### Έλεγχος συνάρτησης Ακτίνας Λίμνης υγρού :

Τύπος :  $R_p = \sqrt{\frac{M_R}{\pi * H_L * D_L}}$  , όπου  $R_p$  η ακτίνα λίμνης του υγρού,  $M_R$  η ολική ποσότητα εκροής,  $H_L$  το βάθος της λίμνης υγρού και  $D_L$  η πυκνότητα του υγρού, οπότε είναι

Είσοδος :  $M_R$  η ολική ποσότητα εκροής

Έξοδος :  $R_p$  η ακτίνα λίμνης του υγρού

Σταθερές παράμετροι :  $\pi$ ,  $H_L$  το βάθος της λίμνης υγρού και  $D_L$  η πυκνότητα του υγρού

Μεταβλητές Παράμετροι :  $M_R$  η ολική ποσότητα εκροής

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	44.7784	44.7784	✓
150000	38.7792	38.7792	✓
500000	70.8008	70.8008	✓
1000000	100.127	100.127	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

#### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της ολικής ποσότητας εκροής
- ◆ Υπολογισμός της ακτίνας λίμνης του υγρού
- ◆ Εγκυρότητα του τύπου υπολογισμού της ακτίνας λίμνης του υγρού
- Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν επιτυχής, αφού στον παραπάνω τύπο το  $M_R$  δίνεται απ' τον χρήστη.
- Επίσης, θα πρέπει οι προγραμματιστές επιτρέπουν στον χρήστη εισαγωγή τιμών εντός των ορίων, τα οποία και ελέγχονται, όμως θα πρέπει να ελέγχονται και περιπτώσεις εισαγωγής χαρακτήρων και συνδυασμό αριθμών-χαρακτήρων που δημιουργούν προβλήματα.

#### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη 4)

### Έλεγχος συνάρτησης Ύψους φλόγας :

$$\text{Τύπος : } H_F = 2 * R_p * K_1 * \left( \frac{M_{BR}}{D_A * \sqrt{2 * R_p * G}} \right)^{0.6}, \text{ όπου } R_p \text{ η ακτίνα λίμνης του υγρού,}$$

$H_F$  το ύψος της φλόγας,  $K_1$  σταθερά (=42),  $M_{BR}$  ειδικός ρυθμός καύσης,  $D_A$  η πυκνότητα του ατμοσφαιρικού αέρα,  $R_p$  η ακτίνα λίμνης υγρού και  $G$  η επιτάχυνση της βαρύτητας

Είσοδος :  $R_p$  η ακτίνα λίμνης του υγρού (που υπολογίζεται με το προηγούμενο τύπο)

Έξοδος :  $H_F$  το ύψος της φλόγας

Σταθερές παράμετροι :  $K_1$  σταθερά (=42),  $M_{BR}$  ειδικός ρυθμός καύσης,  $D_A$  η πυκνότητα του ατμοσφαιρικού αέρα,  $R_p$  η ακτίνα λίμνης υγρού και  $G$  η επιτάχυνση της βαρύτητας

Μεταβλητές Παράμετροι :  $R_p$  η ακτίνα λίμνης του υγρού

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	81.5829	81.5829	✓
150000	73.7685	73.7685	✓
500000	112.429	112.429	✓
1000000	143.298	143.298	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της ακτίνα λίμνης του υγρού (που υπολογίζεται με το προηγούμενο τύπο)
- ◆ Υπολογισμός του ύψους της φλόγας
- ◆ Εγκυρότητα του τύπου υπολογισμού του ύψους της φλόγας
- Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν επιτυχής, αφού στον παραπάνω τύπο το  $M_R$  δίνεται απ' τον χρήστη.
- Επίσης, θα πρέπει οι προγραμματιστές επιτρέπουν στον χρήστη εισαγωγή τιμών εντός των ορίων, τα οποία και ελέγχονται, όμως θα πρέπει να ελέγχονται και περιπτώσεις εισαγωγής χαρακτήρων και συνδυασμό αριθμών-χαρακτήρων που δημιουργούν προβλήματα.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη 4)

### Έλεγχος συνάρτησης Εμβαδού λίμνης υγρού :

Τύπος :  $E_p = 2 * \pi * R_p^2$  , όπου  $R_p$  η ακτίνα λίμνης του υγρού και  $E_p$  το εμβαδό λίμνης υγρού

Είσοδος :  $R_p$  η ακτίνα λίμνης του υγρού

Έξοδος :  $E_p$  το εμβαδό λίμνης υγρού

Σταθερές παράμετροι :  $\pi$

Μεταβλητές Παράμετροι :  $R_p$  η ακτίνα λίμνης του υγρού

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	12598.4	12598.4	✓
150000	9448.82	9448.82	✓
500000	31496.1	31496.1	✓
1000000	62992.1	62992.1	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- Εισαγωγή της ακτίνα λίμνης του υγρού (που υπολογίζεται με το προηγούμενο τύπο)
- Υπολογισμός του εμβαδού λίμνης υγρού
- Εγκυρότητα του τύπου υπολογισμού του εμβαδού λίμνης υγρού
- Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν επιτυχής, αφού στον παραπάνω τύπο το  $M_R$  δίνεται απ' τον χρήστη.
- Επίσης, θα πρέπει οι προγραμματιστές επιτρέπουν στον χρήστη εισαγωγή τιμών εντός των ορίων, τα οποία και ελέγχονται, όμως θα πρέπει να ελέγχονται και περιπτώσεις εισαγωγής χαρακτήρων και συνδυασμό αριθμών-χαρακτήρων που δημιουργούν προβλήματα.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη 4)

### Έλεγχος συνάρτησης Ολικής Θερμότητας Εκπομπής :

Τύπος :  $Q = K_2 * E_p * D_L * C_L$  , όπου Q η ολική θερμότητα εκπομπής,  $K_2$  σταθερα ( $=5,1*10^{-5}$ ),  $E_p$  το εμβαδό της λίμνης,  $D_L$  η πυκνότητα του υγρού και  $C_L$  η καθαρή θερμογόνος δύναμη του υγρού.

Είσοδος :  $E_p$  το εμβαδό της λίμνης (όπου υπολογίζεται απ' την προηγούμενη συνάρτηση)

Έξοδος : Q η ολική θερμότητα εκπομπής

Σταθερές παράμετροι :  $K_2$  σταθερα ( $=5,1*10^{-5}$ ),  $E_p$  το εμβαδό της λίμνης,  $D_L$  η πυκνότητα του υγρού και  $C_L$  η καθαρή θερμογόνος δύναμη του υγρού

Μεταβλητές Παράμετροι :  $E_p$  το εμβαδό της λίμνης

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	1.836e+07	1.836e+07	✓
150000	1.377e+07	1.377e+07	✓
500000	4.59e+07	4.59e+07	✓
1000000	9.18e+07	9.18e+07	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

#### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- Εισαγωγή του εμβαδού λίμνης υγρού
- Υπολογισμός της ολικής θερμότητας εκπομπής
- Εγκυρότητα του τύπου υπολογισμού της ολικής θερμότητας εκπομπής
- Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν επιτυχής, αφού στον παραπάνω τύπο το  $M_R$  δίνεται απ' τον χρήστη.
- Επίσης, θα πρέπει οι προγραμματιστές επιτρέπουν στον χρήστη εισαγωγή τιμών εντός των ορίων, τα οποία και ελέγχονται, όμως θα πρέπει να ελέγχονται και περιπτώσεις εισαγωγής χαρακτήρων και συνδυασμό αριθμών-χαρακτήρων που δημιουργούν προβλήματα.

#### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη 4)

## Έλεγχος συνάρτησης Έντασης Θερμικής Ακτινοβολίας :

Τύπος :  $I = K_3 * \frac{E_p * D_L * C_L}{\pi * R_p * H_F + E_p}$  , όπου I είναι η ένταση της θερμικής ακτινοβολίας,

$K_3$  είναι σταθερά ( $=2,5*10^{-5}$ ),  $E_p$  είναι το εμβαδό της λίμνης υγρού,  $D_L$  η πυκνότητα του υγρού,  $C_L$  η καθαρή θερμογόνος δύναμη του υγρού,  $R_p$  η ακτίνα λίμνης του υγρού,  $H_F$  το ύψος της φλόγας

Είσοδος :  $E_p$  το εμβαδό της λίμνης υγρού και  $R_p$  η ακτίνα λίμνης του υγρού

Έξοδος : I η ένταση της θερμικής ακτινοβολίας

Σταθερές παράμετροι :  $\pi$ ,  $K_3$ ,  $D_L$  η πυκνότητα του υγρού,  $C_L$  η καθαρή θερμογόνος δύναμη του υγρού,  $H_F$  το ύψος της φλόγας

Μεταβλητές Παράμετροι :  $E_p$  το εμβαδό της λίμνης υγρού και  $R_p$  η ακτίνα λίμνης του υγρού

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	1.836e+07	1.836e+07	✓
150000	1.377e+07	1.377e+07	✓
500000	4.59e+07	4.59e+07	✓
1000000	9.18e+07	9.18e+07	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- Εισαγωγή του εμβαδού της λίμνης υγρού και την ακτίνα λίμνης του υγρού
- Υπολογισμός της έντασης της θερμικής ακτινοβολίας
- Εγκυρότητα του τύπου υπολογισμού της έντασης της θερμικής ακτινοβολίας
- Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν επιτυχής, αφού στον παραπάνω τύπο το  $M_R$  δίνεται απ' τον χρήστη.
- Επίσης, θα πρέπει οι προγραμματιστές επιτρέπουν στον χρήστη εισαγωγή τιμών εντός των ορίων, τα οποία και ελέγχονται, όμως θα πρέπει να ελέγχονται και περιπτώσεις εισαγωγής χαρακτήρων και συνδυασμό αριθμών-χαρακτήρων που δημιουργούν προβλήματα.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη 4)

### Έλεγχος συνάρτησης Δόσης Θερμικής Ακτινοβολίας (TDU) ανά Ζώνη :

Τύπος :  $TDU - I(x) = \left( \frac{Z\Omega NH}{T} \right)^{3/4}$  , όπου TDU\_I(x) η δόση θερμικής ακτινοβολίας και x η κάθε μια περίπτωση ζώνης, ZΩNH η κάθε μια απ' τις 4 ζώνες και T σταθερά (=40)

Είσοδος : ZΩNH η κάθε μια απ' τις 4 ζώνες

Έξοδος : TDU\_I(x) η δόση θερμικής ακτινοβολίας και x η κάθε μια περίπτωση ζώνης

Σταθερές παράμετροι : T

Μεταβλητές Παράμετροι : ZΩNH η κάθε μια απ' τις 4 ζώνες

#### Ζώνη I (1500 m) :

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	15,1539	15,1539	✓
150000	15,1539	15,1539	✓
500000	15,1539	15,1539	✓
1000000	15,1539	15,1539	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

**Ζώνη II (450 m) :**

<b>Είσοδος</b>	<b>Έξοδος</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
200000	6.14277	6.14277	✓
150000	6.14277	6.14277	✓
500000	6.14277	6.14277	✓
1000000	6.14277	6.14277	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

**Ζώνη III (170 m) :**

<b>Είσοδος</b>	<b>Έξοδος</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
200000	2.96	2.96	✓
150000	2.96	2.96	✓
500000	2.96	2.96	✓
1000000	2.96	2.96	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

### Ζώνη IV (40 m) :

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	1	1	✓
150000	1	1	✓
500000	1	1	✓
1000000	1	1	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή του ρυθμού διαρροής ανά ζώνη
- ◆ Υπολογισμός της δόσης θερμικής ακτινοβολίας ανά ζώνη
- ◆ Εγκυρότητα του τύπου υπολογισμού της δόσης θερμικής ακτινοβολίας ανά ζώνη
- Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν επιτυχής, αφού στον παραπάνω τύπο το  $M_R$  δίνεται απ' τον χρήστη.
- Επίσης, θα πρέπει οι προγραμματιστές επιτρέπουν στον χρήστη εισαγωγή τιμών εντός των ορίων, τα οποία και ελέγχονται, όμως θα πρέπει να ελέγχονται και περιπτώσεις εισαγωγής χαρακτήρων και συνδυασμό αριθμών-χαρακτήρων που δημιουργούν προβλήματα.

### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 2 – λάθη 4/περίπτωση)

**Έλεγχος συνάρτησης Απόστασης απ' το κέντρο της Φλόγας (χ) ανά Ζώνη :**

Τύπος :  $X = \sqrt{\frac{Q * \cos \theta}{4 * \pi * TDU\_I}}$  , όπου X η απόσταση απ' το κέντρο της φλόγας, Q η

ολική θερμότητα εκπομπής,  $\cos \theta$  το συνημίτονο της γωνίας θέασης και TDU\_I η ένταση της θερμικής ακτινοβολίας ανάλογα με την ζώνη

Είσοδος : TDU\_I η ένταση της θερμικής ακτινοβολίας ανάλογα με την ζώνη (στην ουσία η ζώνη αλλάζει δηλαδή)

Εξοδος : X η απόσταση απ' το κέντρο της φλόγας

Σταθερές παράμετροι : Q η ολική θερμότητα εκπομπής,  $\cos \theta$  το συνημίτονο της γωνίας θέασης,  $\pi$

Μεταβλητές Παράμετροι : : TDU\_I η ένταση της θερμικής ακτινοβολίας ανάλογα με την ζώνη

**Ζώνη I (1500 m) :**

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	288.958	288.958	✓
150000	250.245	250.245	✓
500000	456.883	456.883	✓
1000000	646.13	646.13	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

**Ζώνη II (450 m) :**

<b>Είσοδος</b>	<b>Έξοδος</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
200000	453.852	453.852	✓
150000	393.048	393.048	✓
500000	717.603	717.603	✓
1000000	1014.84	1014.84	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

**Ζώνη III (170 m) :**

<b>Είσοδος</b>	<b>Έξοδος</b>	<b>Αναμενόμενο Αποτέλεσμα</b>	<b>Επιτυχία / Αποτυχία</b>
200000	653.809	653.809	✓
150000	566.215	566.215	✓
500000	1033.76	1033.76	✓
1000000	1461.96	1461.96	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

### Ζώνη IV (40 m) :

Είσοδος	Έξοδος	Αναμενόμενο Αποτέλεσμα	Επιτυχία / Αποτυχία
200000	1124.86	1124.86	✓
150000	974.153	974.153	✓
500000	1778.55	1778.55	✓
1000000	2515.25	2515.25	✓
0	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
-100	Εμφάνιση μηνύματος λάθους & επανάκληση	Εμφάνιση μηνύματος λάθους & επανάκληση	✓
a (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
Φ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
\$ (χαρακτήρας)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗
4λ(αριθμός-γράμμα)	Ατέρμονη επανάληψη	Εμφάνιση μηνύματος λάθους & επανάκληση	✗

#### Παρατηρήσεις.

Η συνάρτηση αυτή πρέπει να έχει τους εξής περιορισμούς:

- ◆ Εισαγωγή της έντασης θερμικής ακτινοβολίας ανά ζώνη
- ◆ Υπολογισμός της απόστασης απ' το κέντρο της φλόγας
- ◆ Εγκυρότητα του τύπου υπολογισμού της απόστασης απ' το κέντρο της φλόγας
- Στη συγκεκριμένη συνάρτηση, ο έλεγχος ήταν επιτυχής, αφού στον παραπάνω τύπο το  $M_R$  δίνεται απ' τον χρήστη.
- Επίσης, θα πρέπει οι προγραμματιστές επιτρέπουν στον χρήστη εισαγωγή τιμών εντός των ορίων, τα οποία και ελέγχονται, όμως θα πρέπει να ελέγχονται και περιπτώσεις εισαγωγής χαρακτήρων και συνδυασμό αριθμών-χαρακτήρων που δημιουργούν προβλήματα.

#### **ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

(παρατηρήσεις 1 – λάθη 4/περίπτωση)

Γ' Έλεγχος  
Ανοιχτού Κουτιού



Ελεγκτής Α' : Υφαντής Γεώργιος  
Ελεγκτής Β' : Αναστασιάδου Έλενα

**ΗΜΕΡΟΛΟΓΙΟ ΕΛΕΓΚΤΗ: 20 Μαΐου 2004**

**Ver 1.0**

**Ελέγχθηκαν τα αρχεία :**

● ScNGL.cpp

**και οι συναρτήσεις :**

- **BXptn**
- **ChangeAdmin**
- DeleteUser
- **Pass**
- **ReadFromFile**
- **WriteToFile**
- **Bleve**
- **PoolFire**
- **FlameJet**
- **MainMenu**
- **ZoneSelect**
- **Default**
- main

Θα τις εξετάσουμε λοιπόν μια προς μια:

## 1. Έλεγχος της κλάσης εξαιρέσεων ΒΧρtn

```
class ΒΧρtn {
private:
    string          synarthsh;
    unsigned int    eidos;
    int             timh1;
public:
    enum {ektosoriwn,maxstring,lathosuser,arxeioporwn};
    ΒΧρtn(string afethria, unsigned int problhma, int t1=0)
    {
        synarthsh=afethria;
        eidos=problhma;
        timh1=t1;
    }
    string getOnoma() const {return synarthsh;}
    unsigned int getEidos() const {return eidos;}
    int getTimh1() const {return timh1;}
};
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στον κώδικα αυτό δεν έχουμε κάποια από τις παραπάνω περιπτώσεις, παρά μόνο τη δήλωση της κλάσης εξαιρέσεων.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 2. Έλεγχος της συνάρτησης changeAdmin(string NAdmiUser,string NAdmipassword)

```
void changeAdmin(string NAdmiUser,string NAdmipassword)
{
    fstream o;
    o.open("Arxeio.txt",3);
    string User>Password>Name>Last;
    string temp="",temp1,temp2,temp3,temp4,stohos1,stohos2;

    o>>Name>>Last>>User>>Password;
    while(!o.eof())
    {
        if(User[0]=='^' && Password[0]=='~')
        {
            stohos1=User;
            stohos2>Password;
            goto label2;
        }
    }
    label2:
    o.close();
    o.open("Arxeio.txt",3);

    o>>temp1>>temp2>>temp3>>temp4;
    while(!o.eof())
    {
        temp=temp + temp1;
        temp=temp + " ";
        temp=temp +temp2;
        temp=temp + " ";
        temp=temp +temp3;
        temp=temp + " ";
        temp=temp +temp4;
        temp=temp +"\n";
        o>>temp1>>temp2>>temp3>>temp4;
    }

    temp.replace(temp.find(stohos1),stohos1.length(),'^'+NAdmiUser);
    temp.replace(temp.find(stohos2),stohos2.length(),'~'+NAdmipassword);

    o.close();
    o.open("Arxeio.txt",3);

    o<<temp;
    o.close();
}
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στην συνάρτηση αυτή έχουμε αλλαγή username και password.

- Έχουμε μια while η οποία διαβάζει το αρχείο "Arχειο.txt" και βάζει σε ένα string το username και το password του χρηστη.
- Έχουμε μια while
- Τέλος με την replace αντικαθιστούμε τα στοιχεία

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

### 3. Έλεγχος της συνάρτησης DeleteUser(string user,string password)

```
void DeleteUser(string user,string password) //
{
    fstream o;
    o.open("Arxeio.txt",3);
    string User,Password,Name,Last;
    string
temp="",temp1,temp2,temp3,temp4,stohos1,stohos2,stohos3,stohos4,ttemp=""
;

    o>>Name>>Last>>User>>Password;
    while(!o.eof())
    {
        if(User==user && Password==password)
        {
            stohos1=Name;
            stohos2=Last;
            stohos3=User;
            stohos4=Password;
            goto label2;
        }
        o>>Name>>Last>>User>>Password;
    }
    label2:
    o.close();
    o.open("Arxeio.txt",3);

    o>>temp1>>temp2>>temp3>>temp4;
    while(!o.eof())
    {
        temp=temp + temp1;
        temp=temp + " ";
        temp=temp +temp2;
        temp=temp + " ";
        temp=temp +temp3;
        temp=temp + " ";
        temp=temp +temp4;
        temp=temp +"\n";
        o>>temp1>>temp2>>temp3>>temp4;
    }

    temp.replace(temp.find(stohos1),stohos1.length()," ");
    temp.replace(temp.find(stohos2),stohos2.length()," ");
    temp.replace(temp.find(stohos3),stohos3.length()," ");
    temp.replace(temp.find(stohos4),stohos4.length()," ");
```

```
o.close();
o.open("Arxeio.txt",2);

o<<temp;
o.close();
}
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στην συνάρτηση αυτή έχουμε αλλαγή username και password.

- Έχουμε μια while η οποία διαβάζει το αρχείο "Arxeio.txt" και βάζει σε string το το username το password, το όνομα και το επώνυμο του χρήστη.
- Έχουμε μια while
- Τέλος με την replace διαγράφουμε τα στοιχεία

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

#### 4. Έλεγχος της συνάρτησης pass(string username, string password)

```
bool pass(string username,string password){ //1

    bool a=false;
    ifstream o;
    o.open("Arxeio.txt");
    if(!o)throw BΧrptn("pass", BΧrptn::arxeioporwn);
    string User,Password,Name,Last;
    o>>Name>>Last>>User>>Password;
    while(!o.eof())&& a==false)
    {
        if(username[0]=='^' && password[0]=='~' && username==User &&
password==Password)
        { //cout<<"\n          CONNECTION SUCCEEDED"<<endl;
            a=true;
            goto label1;
        }
        o>>Name>>Last>>User>>Password;

    }

}

label1:

return a;
}
```

#### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

#### Παρατηρήσεις:

Στην παραπάνω συνάρτηση έχουμε :

- Αρχικά μια if η οποία ελέγχει αν άνοιξε το αρχείο και σε περίπτωση που δεν άνοιξε βγάζει το κατάλληλο μήνυμα.
- Επίσης έχουμε μια while(!o.eof())&& a==false) η οποία διαβάζει ένα αρχείο και με την if ελέγχει αν το username και το password έχουν στην αρχή το συνθηματικό και αν το έχουν επιστρέφει true.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 5. Έλεγχος της συνάρτησης ReadFromFile(string user,string password )

```
bool ReadFromFile(string user,string password ) //1
{
    bool a=false;
    ifstream o;
    o.open("Arxeio.txt");
    if(!o)throw BΧptn("ReadFromFile", BΧptn::arxeioporwn);
    string User,Password,Name,Last;
    o>>Name>>Last>>User>>Password;
    while(!o.eof()&& a==false)
    {

    if(user==User && password==Password)
        {
            a=true;
            goto label1;
        }
    o>>Name>>Last>>User>>Password;

    }//while

label1:

return a;
}
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στην παραπάνω συνάρτηση έχουμε :

- Αρχικά μια if η οποία ελέγχει αν άνοιξε το αρχείο και σε περίπτωση που δεν άνοιξε βγάζει το κατάλληλο μήνυμα.
- Επίσης έχουμε μια while(!o.eof()&& a==false) η οποία διαβάζει ένα αρχείο και με την if ελέγχει αν το username και το password είναι το ίδιο με αυτό στο αρχείο και αν είναι επιστρέφει true.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 6. Έλεγχος της συνάρτησης WriteToFile()

```
void WriteToFile() //1
{
    unsigned int n1,l1,u1,p1;
    string Name,Last,User>Password,keno=" ",g="\n";
    char ch;
    ifstream i;
    i.open("Arxeio.txt",3);

    while(!i.eof())
    {
        i.get(ch);

    }

    do
    {
        cout<<"\n\n          STOIXEIA PROS KATAXWRHSH";
        cout<<"\n\n ONOMA (STOP GIA TELOS) ( <10 )      :";
        cin>>Name;

        if(Name=="stop" || Name=="STOP")break;

        cout<<"\n EPWNYMO ( <15 )          :";
        cin>>Last;
        cout<<"\n USERNAME ( <10 )          :";
        cin>>User;
        cout<<"\n PASSWORD ( <7 )          :";
        cin>>Password;

        n1=Name.length();
        l1=Last.length();
        u1=User.length();
        p1>Password.length();
        if( n1>10 || l1>15 || u1>10 || p1>7 )
        {
            throw BXptn("WriteToFile", BXptn::maxstring);
        }
        break;
    }
    i<<(Name+keno)<<"          "<<(Last+keno)<<"          "<<(User+keno)<<"
    "<<(Password+keno+g);
    }while( Name !="stop");
}
```

## Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

## Παρατηρήσεις:

Στην συνάρτηση WriteToFile γράφουμε τα στοιχεία μας

- Έχουμε μια while που διαβάζει το αρχείο
- Μια do while στην οποία όσο ο χρήστης όσο δεν δίνει stop ζητάει όνομα επώνυμο username και password.
- Με μια if ελέγχετε το μήκος των στοιχείων και σε περίπτωση λάθους βγάζει το κατάλληλο μήνυμα αλλιώς το καταχωρεί στο αρχείο. Εκτός από το μήκος δεν ελέγχετε τη δίνει ο χρήστης .
- Θα έπρεπε να ελέγχονται τα εξής:
  - Για το όνομα να δίνονται μόνο χαρακτήρες (όχι αριθμοί ή σύμβολα), να μην υπάρχουν κενά μεταξύ των χαρακτήρων και το μήκος να είναι τουλάχιστον 3.
  - Για το επώνυμο να δίνονται μόνο χαρακτήρες (όχι αριθμοί ή σύμβολα), να μην υπάρχουν κενά μεταξύ των χαρακτήρων και το μήκος να είναι τουλάχιστον 3.
  - Για το username να δίνονται μόνο χαρακτήρες (όχι αριθμοί ή σύμβολα), να μην υπάρχουν κενά μεταξύ των χαρακτήρων και το μήκος να είναι τουλάχιστον 3.
  - Για το password να δίνονται από 1 έως 6 χαρακτήρες και από 1 έως 6 αριθμούς (όχι σύμβολα), να μην υπάρχουν κενά μεταξύ τους και το μήκος να είναι τουλάχιστον 3

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 7. Έλεγχος της συνάρτησης Blevé(int zone, double &Rs, double &Ts, double &F, double &I, double &L)

```
void Blevé(double MR,int zone, double &Rs, double &Ts, double &F, double
&I, double &L) //2
{
    bool f;
    double t1,t2,t3;

    t3= pow(MR,0.325);
    Rs=((0.5*6.48)*t3);

    Ts=k2*(pow(MR,0.26));

    F=k3*(pow(P,0.32));

    I=pow((zone/Ts),0.75);

    t1= F*MR*Eb*Ta;
    t2=4*M_PI*I*1000*Ts;
    L=sqrt((t1/t2));
}
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στον κώδικα αυτό δεν έχουμε κάποια από τις παραπάνω περιπτώσεις, παρά μόνο τύπους οι οποίοι ελέγχθηκαν ένας προς ένας και δεν βρέθηκαν λάθη.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 8. Έλεγχος της συνάρτησης PoolFire()

```
void PoolFire(const double MR,double &RP,double &HF,double &Q,double
&EP,
                double &I,double &TDU_I1,double
&TDU_I2,double &TDU_I3,double &TDU_I4,
                double &TDU_I5,double &X1,double &X2,double &X3,double
&X4,double &X5)
{

    RP=sqrt(MR/(M_PI*HL*DL));

    HF=(2*RP*K1)*(pow((MBR/(DA*(sqrt(2*RP*G)))) , 0.6 ) );

    EP=2*M_PI*pow(RP,2);

    Q=K2*EP*DL*CL;

    I=K3*((EP*DL*CL)/((M_PI*RP*HF)+EP));

    TDU_I1=pow((zone1/T),(3./4));

    X1=sqrt( (Q*( cos(DEG*(M_PI/180)) ))/(4*M_PI*TDU_I1));

    TDU_I2=pow((zone2/T),0.75);

    X2=sqrt( (Q*cos(DEG*(M_PI/180)))/(4*M_PI*TDU_I2));

    TDU_I3=pow((zone3/T),0.75);

    X3=sqrt( (Q*cos(DEG*(M_PI/180)))/(4*M_PI*TDU_I3));

    TDU_I4=pow((zone4/T),0.75);

    X4=sqrt( (Q*cos(DEG*(M_PI/180)))/(4*M_PI*TDU_I4));

    TDU_I5=37.5;

    X5=sqrt( (Q*cos(DEG*(M_PI/180)))/(4*M_PI*TDU_I5));

}
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στον κώδικα αυτό δεν έχουμε κάποια από τις παραπάνω περιπτώσεις, παρά μόνο τύπους οι οποίοι ελέγχθηκαν ένας προς ένας και δεν παρατηρήθηκε κανένα λάθος.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 9. Έλεγχος της συνάρτησης FlameJet(double FF)

```
double FlameJet(double FF)
{
    double LF;
    LF=pow((FF/0.0006785),(1/2.5));

    return LF;
}
```

### Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στον κώδικα αυτό δεν έχουμε κάποια από τις παραπάνω περιπτώσεις, παρά μόνο ένας τύπος ο οποίος ελέγχθηκε και δεν παρατηρήθηκε κανένα λάθος.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 10. Έλεγχος της συνάρτησης MainMenu(string user,string pas)

```
int MainMenu(string user,string pas)
{
    int choice;

    bool a,b;

    a=pass(user,pas);
    b=ReadFromFile(user,pas);
    if(a)
    {
        do{
            cout<<"\nEPELEXE FAINOMRNO"<<endl;
            cout<<"1.GIA FAINOMENO BLEVE"<<endl;
            cout<<"2.GIA FAINOMENO FLAME JET / FLARE"<<endl;
            cout<<"3.GIA FAINOMENO POOL FIRE"<<endl;
            cout<<"4.KATAXWRHSH STOIXEIWN "<<endl;
            cout<<"5.ALAGH      PASSWORD      KAI      USERNAME
ADMINISTRATOR"<<endl;
            cout<<"0.EXIT"<<endl;
            cout<<"DOSE EPILOGI 0..5 : ";
            cin>>choice;
        } while (choice<0 || choice>5);

        return choice;
    }
    else if(b)
    {
        do{
            cout<<"\nEPELEXE FAINOMENO"<<endl;
            cout<<"1.GIA FAINOMENO BLEVE"<<endl;
            cout<<"2.GIA FAINOMENO FLAME JET / FLARE"<<endl;
            cout<<"3.GIA FAINOMENO POOL FIRE"<<endl;
            cout<<"0.EXIT"<<endl;
            cout<<"DOSE EPILOGI 0..3 : ";
            cin>>choice;
        } while (choice<0 || choice>3);

        return choice;
    }
    else throw BXptn("MAIN MENU", BXptn::lathosuser);
}
```

### Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

Στην συνάρτηση αυτή παίρνω το username και το password από το χρήστη και με τις δυο συναρτήσεις ελέγχω αν είναι σωστά .

➤ Με τις δυο if ελέγχω αν το username και το password υπάρχουν στο αρχείο μου.

- Για την πρώτη περίπτωση που ο χρήστης δεν δίνει το σωστό username και το password ή δεν το δίνει καθολου παρα μονο το σωστο συνθηματικο υπάρχει μια συνθήκη **do while** η οποία δίνει τη δυνατότητα στο χρήστη να επιλέξει μια από της 6 επιλογές. Η συνθήκη υλοποιήθηκε σωστά και δεν υπάρχει πρόβλημα στην εκτέλεση της.

Όμως όταν ο χρήστης δώσει οτιδήποτε άλλο εκτός από αριθμό( γράμμα ή σύμβολο).

- Για την δεύτερη περίπτωση που ο χρήστης δίνει το σωστό username και το password και το σωστό συνθηματικό υπάρχει μια συνθήκη **do while** η οποία δίνει τη δυνατότητα στο χρήστη να επιλέξει μια από της 4 επιλογές. Η συνθήκη υλοποιήθηκε σωστά και δεν υπάρχει πρόβλημα στην εκτέλεση της Η συνθήκη υλοποιήθηκε σωστά και δεν υπάρχει πρόβλημα στην εκτέλεση της.

Όμως όταν ο χρήστης δώσει οτιδήποτε άλλο εκτός από αριθμό( γράμμα ή σύμβολο).

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 11. Έλεγχος της συνάρτησης ZoneSelect()

```
int ZoneSelect()
{
  int choice;
  do{
    cout<<"\nEPELEXE ZWNH "<<endl;
    cout<<"1.ZWNH I (1500)"<<endl;
    cout<<"2.ZWNH II (450)"<<endl;
    cout<<"3.ZWNH III (170)"<<endl;
    cout<<"4.ZWNH IV (40)"<<endl;
    cout<<"0.EXIT"<<endl;
    cout<<"DOSE EPILOGH (0..4) : ";
    cin>>choice;
  } while(choice<0 || choice>4);

  switch(choice)
  {
    case(0):{ return 0; }
    case(1):{ return 1500;}
    case(2):{ return 450;}
    case(3):{ return 170;}
    case(4):{ return 40;}
  }
  //return;
}
```

### Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

- Στον κώδικα αυτό υπάρχει μια συνθήκη **do while με την switch(choice)** η οποία δίνει τη δυνατότητα στο χρήστη να επιλέξει μια από της 5 επιλογές.

Η συνθήκη υλοποιήθηκε σωστά και δεν υπάρχει πρόβλημα στην εκτέλεση της Όμως όταν ο χρήστης δώσει οτιδήποτε άλλο εκτός από αριθμό( γράμμα ή σύμβολο).

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

## 12. Έλεγχος της συνάρτησης Default(string user,string password)

```
void Default(string user,string password){

int zone;
double Rs,Ts,F,I,L;
double FF,LF;
int choice;
double MR;

double RP,HF,Q,EP,TDU_I1,TDU_I2,TDU_I3,TDU_I4,TDU_I5;
double X1,X2,X3,X4,X5;

string Npass,Nuser;
do{

choice=MainMenu(user,password);

switch(choice)
{
case(1):{

cout<<"\n DOSE POSOTHTA DIAROHS (MR>150000 MR<650000) :
";
cin>>MR;
while(MR<150000 || MR>650000)
{
cout<<"\n POSOTHTA EKTOS WRION";
cout<<"\n DOSE POSOTHTA DIAROHS ( MR>150000
MR<650000 ) : ";
cin>>MR;
}
zone=ZoneSelect();
if (zone!=0)
{
Bleve(MR,zone,Rs,Ts,F,I,L);
cout<<"\n\n APOTELESMATA EPEKSERGASIAS"<<endl;
cout<<"\n FAINOMENO BLEVE "<<endl;
cout<<" AKTINA PURINHS SFAIRAS Rs (m) : "<<Rs<<"
m"<<endl;
cout<<" XRONIKI DIARKEIA BLEVE Ts (sec) : "<<Ts<<"
sec"<<endl;
cout<<" POSOSTO EKPEMPOMENHS AKTINOVOLIAS F (%) :
"<<F<<" %"<<endl;
cout<<" THERMIKI AKTINOVOLIA I (W/m2) : "<<I<<"
W/m2"<<endl;
cout<<" APOSTASH PARATHRHRH L (m) : "<<L<<"
m"<<endl<<endl;

```

```

    }
    break;
}
case(2):{

    cout<<"\n          FAINOMENO FLAME JET / FLARE"<<endl;
    cout<<"          DIAROH LOGO DIABROSHS          O,17 - 1,00
kgr/sec"<<endl;
    cout<<"          DIATRSHSH H MHXANIKH ASTOXIA 1,01 - 1,30
kgr/sec"<<endl;
    cout<<"          EISAGWGH RYTHMOY DIAROHS SE (kgr/sec):
";
        cin>>FF;

    cout<<"\n  APOTELESMATA EPEKSERGASIAS"<<endl;
    if(FF>0.17 && FF<=1.0)
    {
        LF=FlameJet(FF);
        cout<<endl;
        cout<<"\n  MHKOS  FLOGAS  (m)  GIA  DIAROH  LOGO
DIABROSHS:"<<LF<<endl<<endl;
    }
    else if(FF>=1.01 && FF<1.30)
    {
        LF=FlameJet(FF);
        cout<<"\n  MHKOS FLOGAS (m) GIA DIATRSHSH H MHXANIKH
ASTOXIA:"<<LF<<endl<<endl;
    }
    else throw BXptn("FlameJet",BXptn::ektosoriwn);

    break;
}
case(3):{

        cout<<"\n\nDWSE  POSOTHTA  DIARROHS
(Kgr): ";
        cin>>MR;

        while(MR<150000 || MR>650000)
        {
            cout<<"\n  POSOTHTA EKTOS ORIWN";
            cout<<"\n  DWSE POSOTHTA DIARROHS ( MR>150000
MR<650000 ) : ";
            cin>>MR;
        }

        PoolFire(MR,RP,HF,Q,EP,I,TDU_I1,TDU_I2,TDU_I3,TDU_I4,TDU_I5,X
1,X2,X3,X4,X5);

```

```

        cout<<"\n\n      APOTELESMATA EPEKSERGASIAS"<<endl;

cout<<"\nAKTINA      LIMNHS      YGROU      NGL      RP(m)
:_____ "<<<RP<<" m"<<endl;
cout<<"YPSOS              FLOGAS              HF(m)
:_____ "<<<HF<<" m"<<endl;
cout<<"EMBADON      LIMNHS      YGROU      NGL      EP(m^2)
:_____ "<<<EP<<" m^2"<<endl;
cout<<"OLIKH      THERMOTHTA      EKPOMPHS      Q(KW)
:_____ "<<<Q<<" KW"<<endl;
cout<<"ENTASH      THERMIKHS      AKTINOBOLIAS      I(KW/m^2)
:_____ "<<<I<<" KW/m^2"<<endl<<endl;

        cout<<"ZONE      I      II      III      IV      V "<<endl;

cout<<"_____
_____ "<<endl;
        cout<<"TDU_I"<<TDU_I1<<"      "<<TDU_I2<<"
"<<TDU_I3<<"      "<<TDU_I4<<"      "<<TDU_I5<<endl;

cout<<"_____
_____ "<<endl;

        cout<<"      X      "<<<X1<<"      "<<<X2<<"
"<<<X3<<"      "<<<X4<<"      "<<<X5<<endl<<endl;

        break;
    }
    case(4):{

                WriteToFile();
        cout<<endl<<"H KATAXWRHSH PRAGMATOPOIHTHIKE"<<endl;
        break;
    }
    case(5):{

        cout<<"\n\nNEA STOIXEIA ADMINISTRATOR ";
        cout<<"\n\nNEO USERNAME:";
        cin>>Nuser;
        cout<<"\n NEO PASSWORD:";
        cin>>Npass;
                changeAdmin(Nuser,Npass);
        cout<<"OI ALLAGES KATAXWRITHIKAN "<<endl<<endl;
        break;
    }
    case(6):{

        cout<<"\n\nSTOIXEIA PROS DIAGRAFH";
        cout<<"\n\n USERNAME:";
        cin>>Nuser;

```

```

    cout<<"\n PASSWORD:";
    cin>>Npass;
    DeleteUser(Nuser,Npass);
    cout<<endl<<"DIAGRAFH XRHSTH OLOKLHROTHIKE "<<endl;
    break;
    }
} //switch

} while(choice!=0);

    switch(choice)
    {
        case(0):exit(1);
    }

}

```

#### Κάλυψη των διαδρομών:

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

#### Παρατηρήσεις:

Στον κώδικα αυτό υπάρχει μια συνθήκη **do while με την switch(choice)** η οποία δίνει τις πέντε περιπτώσεις

- 1) Η οποία κάνει exit
- 2) Ζητάει από το χρήστη ποσότητα διαρροής και ελέγχει τα όρια με μια while και σε περίπτωση που δοθούν λάθος όρια βγάζει το κατάλληλο μήνυμα Με την συνάρτηση ZoneSelect() διαλέγουμε ζώνη ελέγχοντας με μια if δόθηκε ζώνη 0 για να κάνει exit και με την συνάρτηση bleve υπολογίζονται τα αποτελέσματα.
- 3) Ζητάει από το χρήστη ποσότητα διαρροής και με μια if else ελέγχει τα όρια και σε περίπτωση λάθος τιμής βγάζει το κατάλληλο μήνυμα και με την συνάρτηση flame jet υπολογίζει τα αποτελέσματα .
- 4) Ζητάει από το χρήστη ποσότητα διαρροής(χωρίς όμως να διευκρινίζει τα όρια των τιμών) και ελέγχει τα όρια με μια while και σε περίπτωση που δοθούν λάθος όρια βγάζει το κατάλληλο μήνυμα και με την συνάρτηση pool fire υπολογίζει τα αποτελέσματα.
- 5) Καλεί την συνάρτηση writeToFile για να γίνει καταχώρηση των στοιχείων.

- 6) Ζητάει νέο username και password και με την συνάρτηση `changeAdmin` τα αλλάζει. Αλλά δεν ελέγχονται καθόλου τα στοιχεία που δίνει ο χρήστης. Θα έπρεπε να ελέγχονται τα εξής:
- Για το username να δίνονται μόνο χαρακτήρες (όχι αριθμοί ή σύμβολα), να μην υπάρχουν κενά μεταξύ των χαρακτήρων και το μήκος να είναι από 3 έως 10.
  - Για το password να δίνονται από 1 έως 6 χαρακτήρες και από 1 έως 6 αριθμούς (όχι σύμβολα), να μην υπάρχουν κενά μεταξύ τους και το μήκος να είναι από 3 έως 7
- 7) Ζητάει το username και password του χρηστή που θέλουμε να διαγράψουμε και με την συνάρτηση `deleteuser` το διαγραφει.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

### 13. Έλεγχος της συνάρτησης main()

```
main() {

    clrscr();

    string user,pas;
    int c=0;
    int f;

    do
    {
    try{

    printf("\nUSERNAME: ");
    cin>>user;
    printf("\nPASSWORD: ");
    cin>>pas;

    Default(user,pas);
    }
    catch(BXptn x)
    {
        switch(x.getEidos())
        {
            case (BXptn::ektosoriwn):
            {
                cout<<"<<x.getOnoma()<<":  TIMH  EKTOS  ORIWN  STO
                FAINOMENO FLAME JET / FLARE"<<endl;
                break;
            }
            case (BXptn::maxstring):
            {
                cout<<"<<x.getOnoma()<<": POLY MEGALO ONOMA, EPWNYMO,
                USERNAME H PASSWORD"<<endl;
                break;
            }
            case (BXptn::lathosuser):
            {
                c++;
                cout<<"<<x.getOnoma()<<":  CONNECTION  FAILED:  MH
                KATAXWRHMENOS XRHSTHS"<<endl;
                break;
            }
            case(BXptn::arxeioporwn):
            {
```

```
        cout<<" "<<x.getOnoma()<<":      ARXEIO      PORWN      DEN
BRETHIKE"<<endl;
        break;
    }//case
    }//switch
} //catch
}while(c<3);

}
```

### Κάλυψη των διαδρομών

- Εντολές
- Αποφάσεις
- Συνθήκες
- Αποφάσεις , συνθήκες
- Πολλαπλές συνθήκες

### Παρατηρήσεις:

- ◆ Στην main έχουμε μια do while η οποία δίνει τη δυνατότητα στο χρήστη να δώσει μέχρι τρεις φορές username και password μέχρι να δώσει το σωστό.
- ◆ Μετά με στη try παίρνει το username και password και καλεί την συνάρτηση default .
- ◆ Και τέλος ελέγχει τις περιπτώσεις λάθους
- ◆ Για την περίπτωση που είναι εκτός ορίων
- ◆ Για την περίπτωση μεγάλου string(Έπρεπε να γίνετε ξεχωριστός έλεγχος για μεγάλο όνομα, επώνυμο,username, password)
- ◆ Για την περίπτωση λάθους user.

**ΕΛΕΓΧΟΣ ΕΠΙΤΥΧΗΣ**

**ΤΕΚΜΗΡΙΩΣΗ ΕΛΕΓΧΟΥ ΛΟΓΙΣΜΙΚΟΥ**  
(ΗΜΕΡΟΛΟΓΙΟ  
ΛΑΘΗ  
ΜΕΤΡΗΣΕΙΣ)



**Ελεγκτής Α' : Υφαντής Γεώργιος**  
**Ελεγκτής Β' : Αναστασιάδου Έλενα**

### Γενικό σχέδιο Test

Στις 07/05/2003, άρχισε η κατασκευή του γενικού σχεδίου test. Το γενικό σχέδιο test περιέχει περιπτώσεις καθώς και διαδικασίες αντιμετώπισης πιθανών προβλημάτων που μπορούν να δημιουργηθούν κατά την διάρκεια της υλοποίησης του λογισμικού. Όλο το γενικό σχέδιο test είναι βασισμένο στις απαιτήσεις του πελάτη και τις προδιαγραφές.

### Περιπτώσεις Test

Οι περιπτώσεις test περιέχουν τα πιθανά λάθη που μπορεί να παρουσιάσει το λογισμικό (αλγόριθμο), οι τιμές που μπορεί να δεχθεί (τιμές εισαγωγής), να απορρίψει, να επεξεργάζεται και να εμφανίζει στην έξοδο. Για την εύρεση των πιθανών λαθών χρειάστηκε αρκετή μελέτη των απαιτήσεων και των προδιαγραφών.

### Διαδικασίες test

Στις διαδικασίες test προσπαθήσαμε μέσα από τα λάθη που είναι πολύ πιθανά να γίνουν, να ανακαλύψουμε τρόπους αντιμετώπισης αυτών, προλαμβάνοντας έτσι την κατάρρευση του ίδιου του λογισμικού την ασφάλεια του χρήστη που το χρησιμοποιεί καθώς και όσους συνδέονται άμεσα και έμμεσα με αυτό.

### Τερματισμός test

Ο τερματισμός του test έγινε όταν καλύφθηκαν οι προδιαγραφές που συμφώνησαν ο πελάτης με τον project manager της ομάδας μας. Όταν σε αυτές δεν μπορούσαμε να βρούμε άλλα λάθη, μετά από την διεκπεραίωση του ελέγχου συγκεντρώσαμε όλα τα μετρήσιμα στοιχεία για να δημιουργήσουμε, στατιστικά στοιχεία που είναι χρήσιμα, γιατί μέσα από αυτά ανακαλύψαμε βασικές πληροφορίες, χρήσιμες για επόμενα project.

### Ακριβές Χρονοδιάγραμμα Ελέγχων

## Ημερολόγιο TEST

Ημερ/νίες:	Έλεγχου:	Ημερ/νίες:	Έλεγχου:	Ημερ/νίες:	Έλεγχου:
7/5/2004	Γενικό Σχέδιο Τέστ	17/5/2004	Συνάρτηση PoolFire (A.K.)	21/5/2004	Έναρξη Γ' Τέστ Κλειστού κουτιού
9/5/2004	Έναρξη Α' Τέστ Κλειστού κουτιού	17/5/2004	Συνάρτηση FlameJet (A.K.)	21/5/2004	Συνάρτηση MainMenu (K.K.)
9/5/2004	Συνάρτηση MainMenu (K.K.)	17/5/2004	Συνάρτηση MainMenu (A.K.)	21/5/2004	Συνάρτηση Blevé (K.K.)
10/5/2004	Έναρξη Α' Τέστ Ανοιχτού κουτιού	18/5/2004	Συνάρτηση FlameJet (K.K.)	21/5/2004	Συνάρτηση ZoneSelect (K.K.)
10/5/2004	Συνάρτηση Blevé (A.K.)	18/5/2004	Λήξη Α' Τέστ Κλειστού κουτιού	21/5/2004	Συνάρτηση PoolFire (K.K.)
11/5/2004	Συνάρτηση PoolFire (A.K.)	18/5/2004	Συνάρτηση ZoneSelect (A.K.)	22/5/2004	Συνάρτηση Default (K.K.)
12/5/2004	Συνάρτηση Blevé (K.K.)	18/5/2004	Συνάρτηση Default (A.K.)	22/5/2004	Συνάρτηση FlameJet (K.K.)
12/5/2004	Συνάρτηση FlameJet (A.K.)	19/5/2004	Συνάρτηση Default (A.K.)	22/5/2004	Συνάρτηση pass (K.K.)
13/5/2004	Συνάρτηση ZoneSelect (K.K.)	19/5/2004	Λήξη Α' Τέστ Ανοιχτού κουτιού	22/5/2004	Συνάρτηση Blevé (A.K.)
14/5/2004	Συνάρτηση Default (K.K.)	19/5/2004	Συνάρτηση pass (K.K.)	22/5/2004	Συνάρτηση PoolFire (A.K.)
15/5/2004	Συνάρτηση MainMenu (A.K.)	19/5/2004	Συνάρτηση ReadFromFile(K.K.)	22/5/2004	Συνάρτηση FlameJet (A.K.)
15/5/2004	Έναρξη Β' Τέστ Ανοιχτού κουτιού	19/5/2004	Συνάρτηση ChangeAdmin(K.K.)	22/5/2004	Συνάρτηση MainMenu (A.K.)
15/5/2004	Class ΒΧρptn (A.K.)	19/5/2004	Συνάρτηση ReadFromFile(K.K.)	22/5/2004	Συνάρτηση ChangeAdmin(KK)
15/5/2004	Συνάρτηση ChangeAdmin(A.K.)	19/5/2004	Συνάρτηση PoolFire (K.K.)	22/5/2004	Συνάρτηση WriteToFile (K.K.)
15/5/2004	Συνάρτηση pass (A.K.)	19/5/2004	Class ΒΧρptn (K.K.)	23/5/2004	Συνάρτηση ZoneSelect (A.K.)
16/5/2004	Έναρξη Β' Τέστ Κλειστού κουτιού	19/5/2004	Συνάρτηση main (A.K.)	23/5/2004	Συνάρτηση Default (A.K.)
16/5/2004	Συνάρτηση MainMenu (K.K.)	19/5/2004	Λήξη Β' Τέστ Ανοιχτού κουτιού	23/5/2004	Συνάρτηση main (A.K.)
16/5/2004	Συνάρτηση Blevé (K.K.)	19/5/2004	Λήξη Β' Τέστ Κλειστού κουτιού	23/5/2004	Συνάρτηση ReadFromFile(K.K)
16/5/2004	Συνάρτηση ReadFromFile(A.K.)	20/5/2004	Έναρξη Γ' Τέστ Ανοιχτού κουτιού	23/5/2004	Συνάρτηση DeleteUser (K.K.)
16/5/2004	Συνάρτηση WriteToFile (A.K.)	20/5/2004	Class ΒΧρptn (A.K.)	23/5/2004	Class ΒΧρptn (K.K.)
16/5/2004	Συνάρτηση Blevé (A.K.)	20/5/2004	Συνάρτηση ChangeAdmin(A.K.)	23/5/2004	Λήξη Γ' Τέστ Ανοιχτού κουτιού
17/5/2004	Συνάρτηση ZoneSelect (A.K.)	20/5/2004	Συνάρτηση DeleteUser (A.K.)	23/5/2004	Λήξη Γ' Τέστ Κλειστού κουτιού
17/5/2004	Συνάρτηση ZoneSelect (K.K.)	21/5/2004	Συνάρτηση pass (A.K.)	23/5/2004	Τεκμηρίωση Ελέγχου Λογισμικού
17/5/2004	Συνάρτηση Default (K.K.)	21/5/2004	Συνάρτηση ReadFromFile(A.K.)		
17/5/2004	Συνάρτηση FlameJet (K.K.)	21/5/2004	Συνάρτηση WriteToFile(A.K.)		

✚ Ποιοτικά Στατιστικά στοιχεία που προκύπτουν απ' τον Έλεγχο για κάθε σφάλμα σε κάθε συνάρτηση που ελέγχθηκε :

➤ Συγκεντρωτικός Πίνακας όλων των σφαλμάτων κατά την διάρκεια όλων των ελέγχων, σε όλες τις συναρτήσεις :

Συναρτήσεις	Α' Έλεγχος				Β' Έλεγχος				Γ' Έλεγχος			
	Κρίσιμα	Σημαντικά	Μη Σημαντικά	Σύνολο Σφαλμάτων	Κρίσιμα	Σημαντικά	Μη Σημαντικά	Σύνολο Σφαλμάτων	Κρίσιμα	Σημαντικά	Μη Σημαντικά	Σύνολο Σφ/των
Bleve	20	0	0	20/20	20	0	0	20/20	12	0	0	12/60
FlameJet	0	0	0	0/6	0	0	0	0/6	4	10	2	16/32
MainMenu	11	7	3	21/25	11	8	2	21/26	9	10	2	21/26
ZoneSelect	17	4	2	23/29	18	4	2	24/30	18	4	2	24/30
Default	3	10	2	15/25	4	10	2	16/26	4	0	2	6/21
ChangeAdmin					9	0	0	9/9	0	8	0	8/9
pass					0	3	0	3/13	0	4	0	4/13
ReadFromFile					0	3	0	3/13	0	4	0	4/13
WriteToFile					2	5	0	7/9	1	6	0	7/10
PoolFire					103	1	0	104/139	56	0	2	58/151
DeleteUser									3	4	0	7/7
class BXptn	← Η συμπεριφορά της, ελέγχεται μέσα απ' τις άλλες συναρτήσεις, γιατί είναι κλάση Εξαιρέσεων →											

➤ Συγκεντρωτικός Πίνακας συνολικών σφαλμάτων ανά κατηγορία (κρίσιμα, σημαντικά ή μη σημαντικά), ανά διαδικασία Ελέγχου :

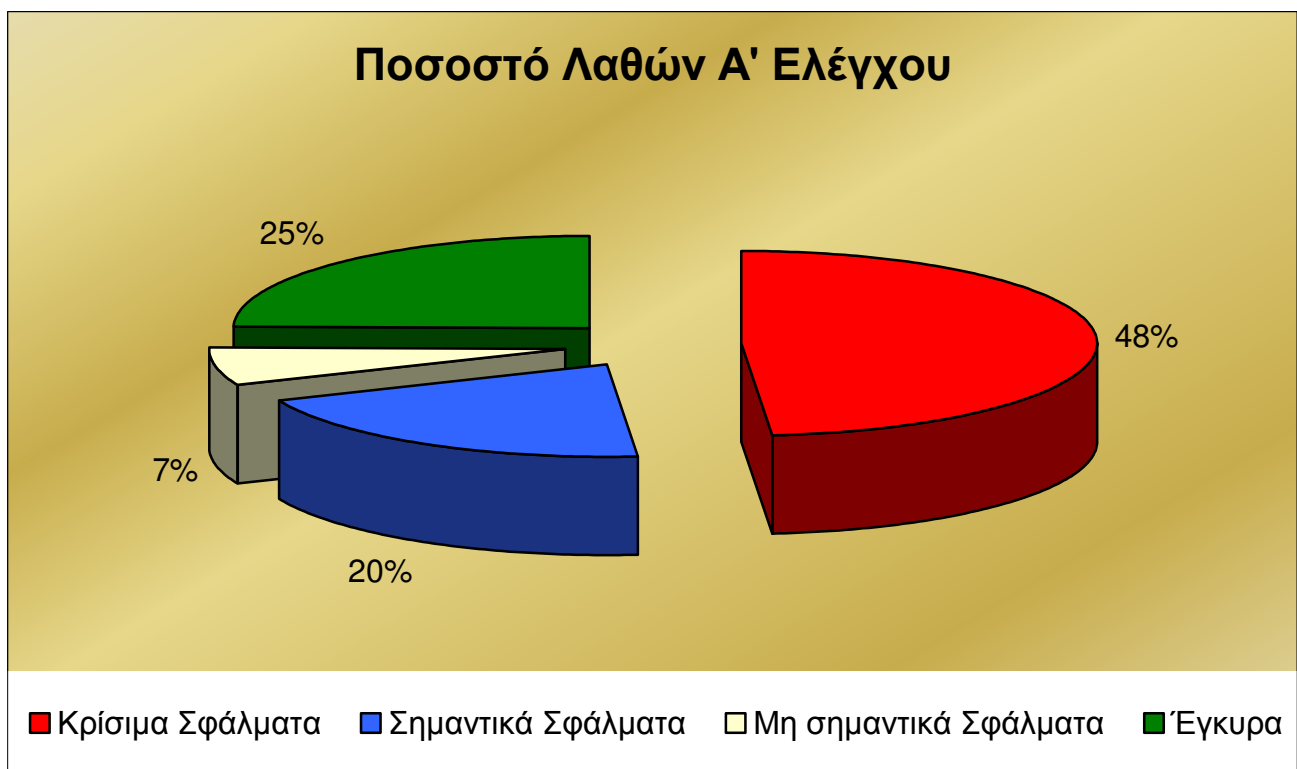
<b>A' Έλεγχος</b>			<b>B' Έλεγχος</b>			<b>Γ' Έλεγχος</b>		
<i>Κρίσιμα</i>	<i>Σημαντικά</i>	<i>Μη Σημαντικά</i>	<i>Κρίσιμα</i>	<i>Σημαντικά</i>	<i>Μη Σημαντικά</i>	<i>Κρίσιμα</i>	<i>Σημαντικά</i>	<i>Μη Σημαντικά</i>
51	21	7	167	34	6	107	50	10
<b>Σύνολο Σφαλμάτων A' Ελέγχου</b>			<b>Σύνολο Σφαλμάτων B' Ελέγχου</b>			<b>Σύνολο Σφαλμάτων Γ' Ελέγχου</b>		
79 / 105 ή 75,23 %			207 / 291 ή 71,13 %			167 / 372 ή 44,89 %		

➤ Διαγράμματα Αναπαράστασης στοιχείων της διαδικασίας Ελέγχου :

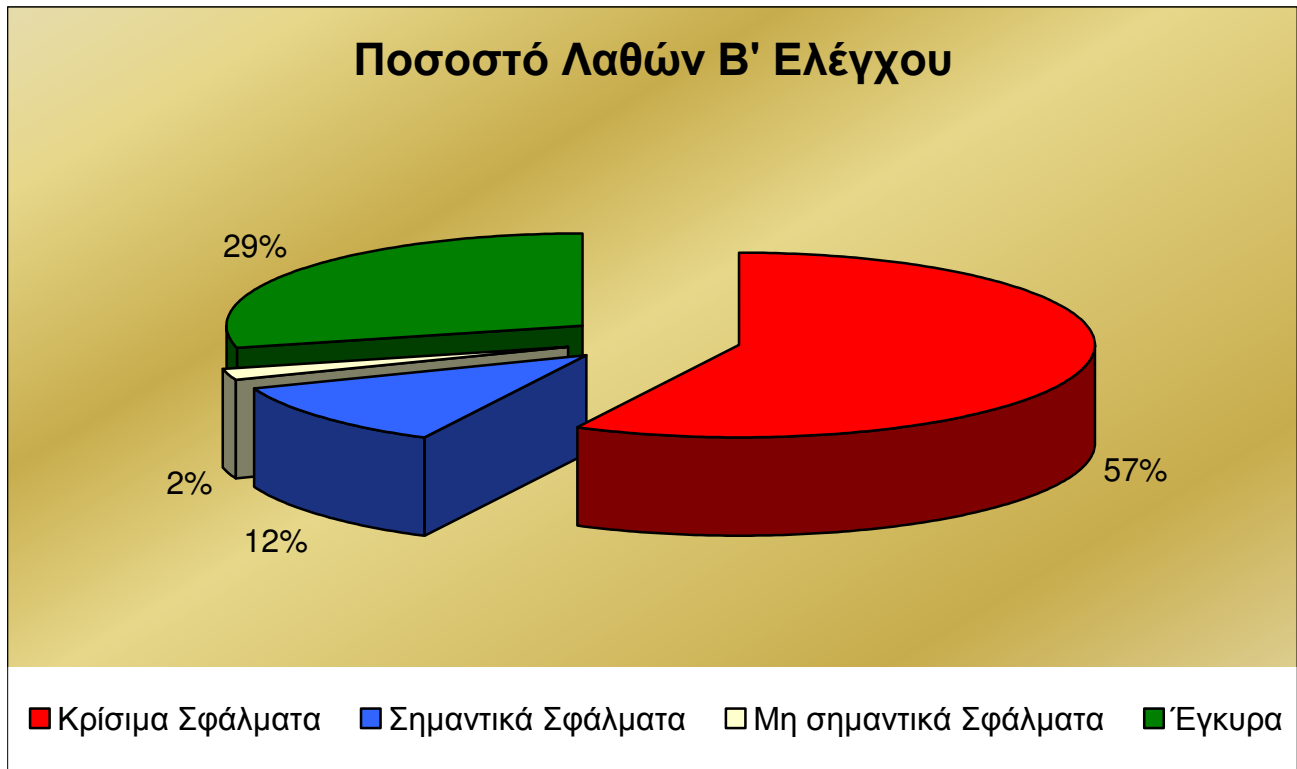
■ Γράφημα Πορείας Λαθών



■ Ποσοστό Σφαλμάτων Α' Ελέγχου :



■ Ποσοστό Σφαλμάτων Β' Ελέγχου :



■ Ποσοστό Σφαλμάτων Γ' Ελέγχου :

