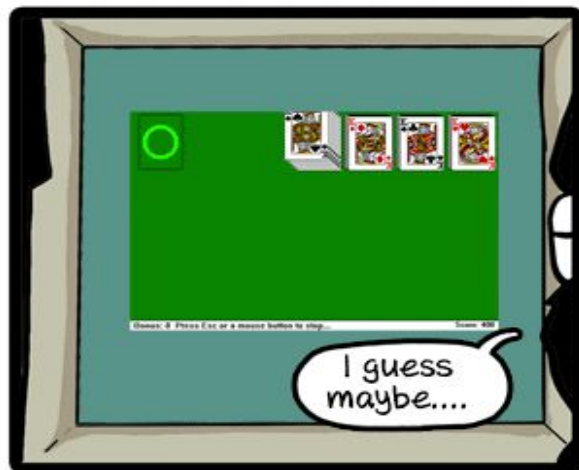
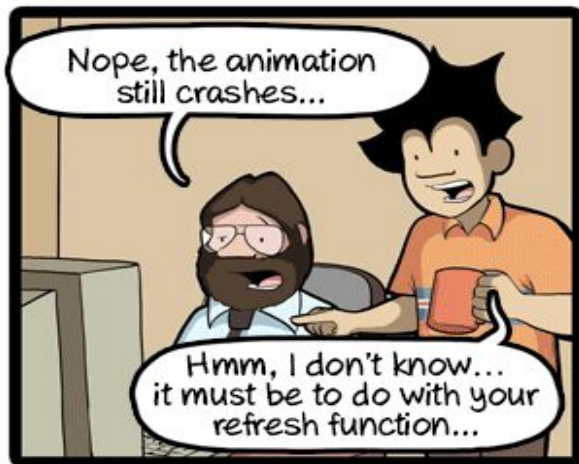
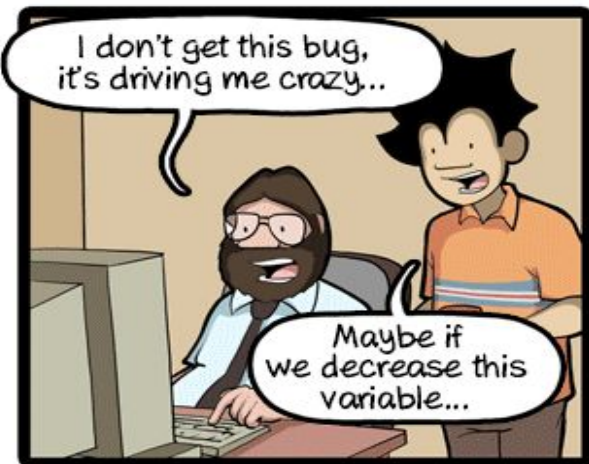


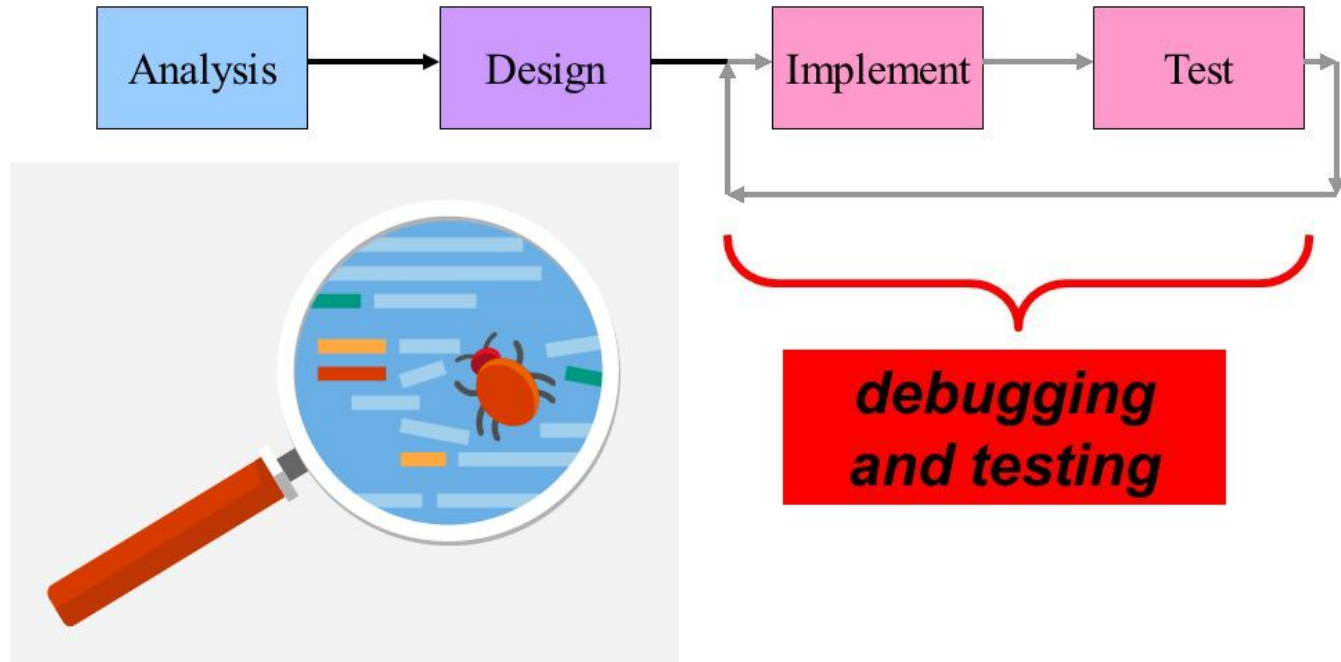
Εκσφαλμάτωση Προγράμματος (Debugging)

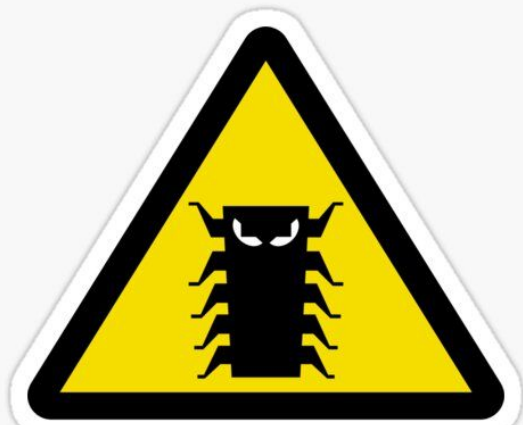
{B1} Κεφάλαιο 13 (Σελ. 232-235)

{B2} Κεφάλαιο 5 (Σελ. 115-142)



Development Cycle





Millennium bug
act now!

REMEMBER

Turn your computer off
before midnight on

31/12/99.



*We've exterminated
the Y2K bug!*



this machine is your

Κατηγορίες Λαθών (Errors - Bugs)

- **Συντακτικά** Λάθη (κατά την Υλοποίηση)
- Λάθη κατά την **Εκτέλεση** (*Run-Time Errors*)
- **Λογικά** Λάθη (λάθος αποτελέσματα)

1. Συντακτικά Λάθη

Ο **Μεταγλωττιστής** (*compiler*) ή ο **Διερμηνευτής** ανιχνεύει λοιπόν τα Συντακτικά Λάθη του προγράμματος και εμφανίζει τα κατάλληλα **διαγνωστικά** μηνύματα.

Αν το πρόγραμμα περιέχει ένα Συντακτικό Λάθος, τότε **δεν επιτρέπεται η εκτέλεσή του**, μέχρι να το διορθώσει ο προγραμματιστής.

- Μη δήλωση Μεταβλητών
- Λάθος Όνομα Μεταβλητών
- Δεσμευμένες Λέξεις
- Ανοιχτές Παρενθέσεις
- Κλείσιμο Εμφωλευμένων ΑΝ
- Λάθος Τύπος Δεδομένων

```
1 ΠΡΟΓΡΑΜΜΑ BUGS
2 ΜΕΤΑΒΛΗΤΕΣ
3 ΑΚΕΡΑΙΕΣ: E, B[10], I
4 ΑΡΧΗ
5   E ← 5
6   I ← 1
7
8   B[I] ← 2 * E
9
10 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

θόνη εκτέλεσης

```
1 Το όνομα «E» δεν είναι έγκυρο
& όνομα μεταβλητής.
2
```

```
1 ΠΡΟΓΡΑΜΜΑ BUGS
2 ΜΕΤΑΒΛΗΤΕΣ
3 ΑΚΕΡΑΙΕΣ: A, B[10], I
4 ΑΡΧΗ
5
6 A ← 'TEST'
7
8 I ← 0
9
10 B[I] ← A
11
12 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Οθόνη εκτέλεσης

```
1 Δεν είναι δυνατόν να γίνει αυτή
& η ανάθεση τιμής γιατί τα
& ορίσματα είναι διαφορετικού
& τύπου δεδομένων.
2 «A»: ακέραια μεταβλητή
3 «'TEST'»: αλφαριθμητική σταθερά
4
```



```
1 ΠΡΟΓΡΑΜΜΑ BUGS
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: A, B[10], I
4 ΑΡΧΗ
5
6   A ← 100
7
8   I ← 5
9
10  B[I] ← 'A'
11
12 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Οθόνη εκτέλεσης

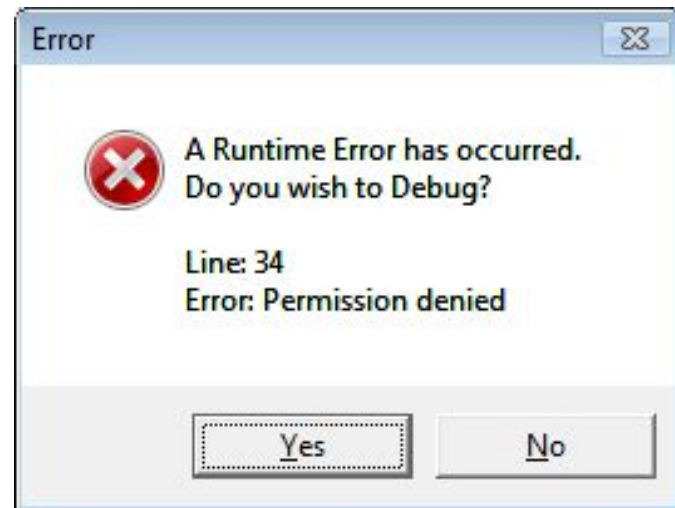
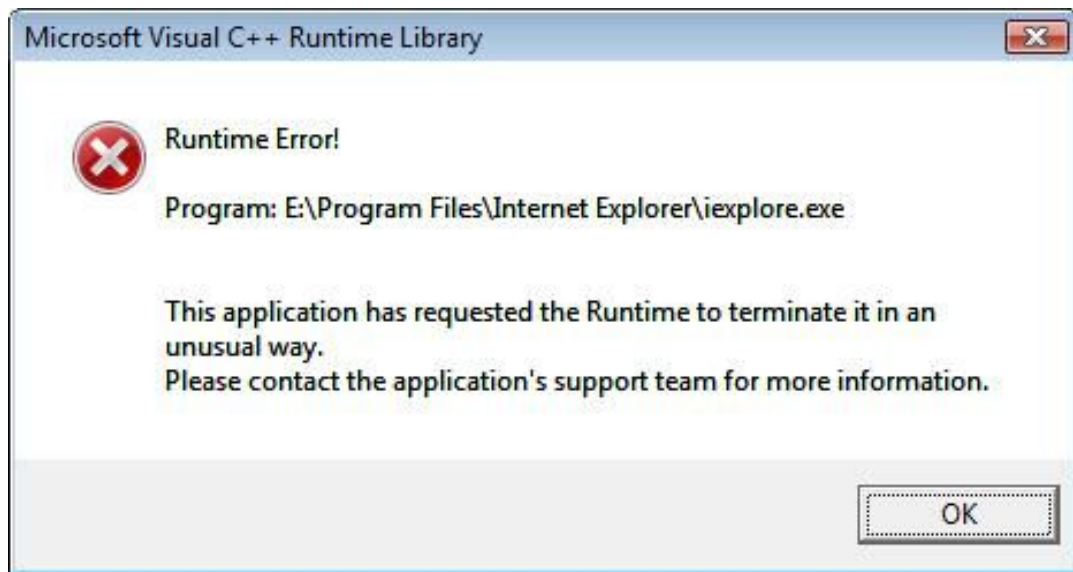
```
1 Δεν είναι δυνατόν να γίνει αυτή
& η ανάθεση τιμής γιατί τα
& ορίσματα είναι διαφορετικού
& τύπου δεδομένων.
2 «B[I]»: στοιχείο πίνακα
& ακεραίων
3 «'A'»: αλφαριθμητική σταθερά
4
```

```
1 ΠΡΟΓΡΑΜΜΑ BUGS
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: A, B[10], I
4 ΑΡΧΗ
5
6   A ← 100
7
8   I ← 5
9
10  B[I] ← 2 * A / 10
11
12 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Οθόνη εκτέλεσης

```
1 Δεν είναι δυνατόν να γίνει αυτή η
& ανάθεση τιμής γιατί τα ορίσματα
& είναι διαφορετικού τύπου
& δεδομένων.
2 «B[I]»: στοιχείο πίνακα ακεραίων
3 «2 * A / 10»: πραγματική έκφραση
4
```





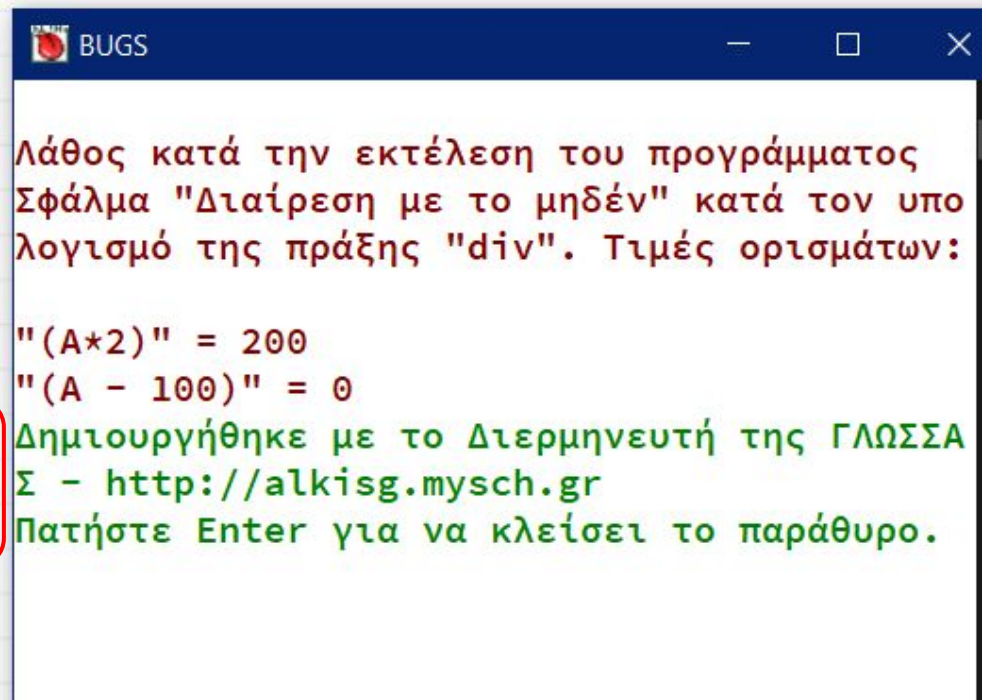


2. Λάθη Εκτέλεσης (Καθοριστικότητα)

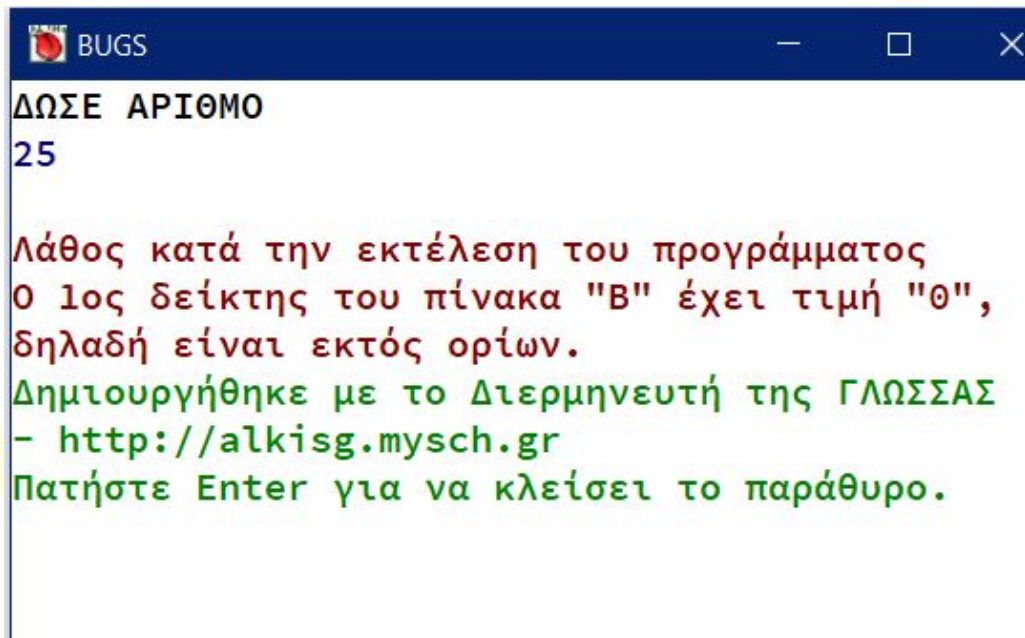
Τα λάθη που προκαλούνται κατά το **χρόνο εκτέλεσης** του προγράμματος, είναι πιο επώδυνα γιατί εμφανίζονται σε πραγματικό περιβάλλον εκτέλεσης και συνήθως προκαλούν **τον αντικανονικό τερματισμό** της εφαρμογής και το κρέμασμα (**crash**) του συστήματος.

Για παράδειγμα, τέτοια λάθη προκύπτουν όταν επιχειρήσουμε να **διαιρέσουμε με το μηδέν** ή αν κατά την ανάγνωση ενός ακεραίου αριθμού **εισαχθεί ένα γράμμα**.


```
1 ΠΡΟΓΡΑΜΜΑ BUGS
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: A, B[10], I
4 ΑΡΧΗ
5
6   A <- 100
7
8   I <- 5
9
10  B[I] <- (A*2) div (A-100)
11
12 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```



```
1 ΠΡΟΓΡΑΜΜΑ BUGS
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: A, B[10], I
4 ΑΡΧΗ
5
6   ΓΡΑΨΕ 'ΔΩΣΕ ΑΡΙΘΜΟ'
7   ΔΙΑΒΑΣΕ A
8
9   I ← 0
10
11  B[I] ← 2 * A
12
13 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```



BUGS

ΔΩΣΕ ΑΡΙΘΜΟ
25

Λάθος κατά την εκτέλεση του προγράμματος
Ο 1ος δείκτης του πίνακα "B" έχει τιμή "0",
δηλαδή είναι εκτός ορίων.

Δημιουργήθηκε με το Διερμηνευτή της ΓΛΩΣΣΑΣ
- <http://alkisg.mysch.gr>
Πατήστε Enter για να κλείσει το παράθυρο.

```
1 ΠΡΟΓΡΑΜΜΑ BUGS
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: A, B[10], I
4 ΑΡΧΗ
5
6   I ← 5
7
8   ΓΡΑΨΕ 'ΔΩΣΕ ΑΡΙΘΜΟ'
9   ΔΙΑΒΑΣΕ A
10
11  B[I] ← 2 * A
12
13 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

BUGS

ΔΩΣΕ ΑΡΙΘΜΟ
AB

Λάθος κατά την εκτέλεση του προγράμματος
Δεν είναι δυνατή η ανάθεση της τιμής "AB"
στο "A": ακέραια μεταβλητή.

Δημιουργήθηκε με το Διερμηνευτή της ΓΛΩΣΣΑ
Σ - <http://alkisg.mysch.gr>
Πατήστε Enter για να κλείσει το παράθυρο.

```
1 ΠΡΟΓΡΑΜΜΑ BUGS
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: A, B[10], I
4 ΑΡΧΗ
5
6   ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10
7     B[I] ← 2 * A
8 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
9
10 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```



BUGS

Το πρόγραμμα προσπάθησε να χρησιμοποιήσει την τιμή της μεταβλητής "A" ενώ αυτή δεν έχει αρχικοποιηθεί.

Δημιουργήθηκε με το Διερμηνευτή της ΓΛΩΣΣΑΣ - <http://alkisg.mysch.gr>

Πατήστε Enter για να κλείσει το παράθυρο.

3. Λογικά Λάθη (Αποτελεσματικότητα)

Τα λογικά λάθη είναι συνήθως **λάθη σχεδιασμού** και δεν προκαλούν τη διακοπή της εκτέλεσης του προγράμματος.

Για να εντοπίσουμε τα λογικά λάθη μπορούμε να κάνουμε **δοκιμαστικές εκτελέσεις** του προγράμματός μας και να ελέγξουμε αν για συγκεκριμένες **τιμές εισόδου**, το πρόγραμμά μας εξάγει **σωστά αποτελέσματα**.

```
1 ΠΡΟΓΡΑΜΜΑ BUGS
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: A, B[10], I
4 ΑΡΧΗ
5   A ← 5
6   I ← 1
7   ΟΣΟ I <= 10 ΕΠΑΝΑΛΑΒΕ
8     B[I] ← 2 * A
9     A ← A + 1
10  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
11
12 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Προειδοποίηση



Αυτός ο αλγόριθμος έχει ήδη εκτελέσει 100.000 εντολές. Αν νομίζετε ότι βρίσκεται σε ατέρμονα βρόχο, μετά το κλείσιμο του διαλόγου πατήστε «Παύση εκτέλεσης» (Pause) ή «Αργή εκτέλεση» (Shift + Pause) για να εντοπίσετε το πρόβλημα.

Εντάξει

ΑΤΕΡΜΩΝ ΒΡΟΧΟΣ - INFINITE LOOP


```
1 ΠΡΟΓΡΑΜΜΑ BUGS
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: Σ, Χ, Ι, ΠΛ
4 ΑΡΧΗ
5   ΠΛ ← 0
6   ΓΙΑ Ι ΑΠΟ 10 ΜΕΧΡΙ 1
7     ΔΙΑΒΑΣΕ Χ
8     ΑΝ Χ > 0 ΤΟΤΕ
9       Σ ← Σ + Χ
10      ΠΛ ← ΠΛ + 1
11 ΤΕΛΟΣ_ΑΝ
12 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
13
14 ΓΡΑΨΕ 'ΜΟ: ', Σ/ΠΛ
15 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

→ ΜΕ_ΒΗΜΑ -1 ! ΛΟΓΙΚΟ ΛΑΘΟΣ

→ Σ ← 0 ! ΑΡΧΙΚΟΠΟΙΗΣΗ

→ ΑΝ ΠΛ <> 0 ΤΟΤΕ ! ΔΙΑΙΡΕΣΗ ΜΕ ΜΗΔΕΝ

WHAT IS BETA TESTING?



13.2 | Εκσφαλμάτωση

Τα λάθη που κυρίως μας απασχολούν στη φάση της εκσφαλμάτωσης είναι τα **λογικά** λάθη και τα λάθη που παρουσιάζονται κατά το χρόνο **εκτέλεσης** του προγράμματος.

Για τον εντοπισμό ενός λάθους δεν υπάρχουν ιδιαίτερα **μυστικά και κόλπα**. Η εκσφαλμάτωση είναι ένα **πρόβλημα λογικής** και όσο πιο καλά αντιλαμβάνεται ο προγραμματιστής τον τρόπο που εργάζεται το πρόγραμμα, τόσο πιο εύκολα και σύντομα θα εντοπίσει λάθη που προκαλούν δυσλειτουργίες.

13.2 | Εκσφαλμάτωση



Η εισαγωγή γραμμών με σχόλια σε ένα πρόγραμμα υποβοηθά σημαντικά την εκσφαλμάτωση.



Τα ονόματα των μεταβλητών πρέπει να ανάγουν στο περιεχόμενό τους. Έτσι διευκολύνεται η εκσφαλμάτωση.

Εκσφαλμάτωση Λογικών Λαθών στις Δομές Επιλογής

Τι θα εμφανιστεί για ένα μαθητή που πήρε βαθμό 16;

```
ΔΙΑΒΑΣΕ β
ΑΝ β >= 18 ΤΟΤΕ
    ΓΡΑΨΕ 'ΑΡΙΣΤΑ'
ΑΛΛΙΩΣ_ΑΝ β >= 10 ΤΟΤΕ
    ΓΡΑΨΕ 'ΚΑΛΩΣ'
ΑΛΛΙΩΣ_ΑΝ β >= 14 ΤΟΤΕ
    ΓΡΑΨΕ 'ΛΙΑΝ ΚΑΛΩΣ'
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΡΑΝΤΕΒΟΥ ΤΟ ΣΕΠΤΕΜΒΡΗ'
ΤΕΛΟΣ_ΑΝ
```



! $\beta \in (0, 20]$

ΔΙΑΒΑΣΕ β

ΑΝ $\beta > 0$ ΚΑΙ $\beta < 9.5$ ΤΟΤΕ

ΓΡΑΨΕ 'ΡΑΝΤΕΒΟΥ ΤΟ ΣΕΠΤΕΜΒΡΗ'

ΑΛΛΙΩΣ_ΑΝ $\beta < 14$ ΤΟΤΕ

ΓΡΑΨΕ 'ΚΑΛΩΣ'

ΑΛΛΙΩΣ_ΑΝ $\beta < 18$ ΤΟΤΕ

ΓΡΑΨΕ 'ΛΙΑΝ ΚΑΛΩΣ'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'ΑΡΙΣΤΑ'

ΤΕΛΟΣ_ΑΝ

ΔΙΑΒΑΣΕ β

ΑΝ $\beta > 0$ ΚΑΙ $\beta < 9.5$ ΤΟΤΕ

ΓΡΑΨΕ 'ΡΑΝΤΕΒΟΥ ΤΟ ΣΕΠΤΕΜΒΡΗ'

ΑΛΛΙΩΣ_ΑΝ $\beta \geq 9.5$ ΚΑΙ $\beta < 14$ ΤΟΤΕ

ΓΡΑΨΕ 'ΚΑΛΩΣ'

ΑΛΛΙΩΣ_ΑΝ $\beta \geq 14$ ΚΑΙ $\beta < 18$ ΤΟΤΕ

ΓΡΑΨΕ 'ΛΙΑΝ ΚΑΛΩΣ'

ΑΛΛΙΩΣ_ΑΝ $\beta \geq 18$ ΚΑΙ $\beta \leq 20$ ΤΟΤΕ

ΓΡΑΨΕ 'ΑΡΙΣΤΑ'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Λάθος βαθμός'

ΤΕΛΟΣ_ΑΝ

Βοηθάει συχνά να ξεκινάμε από τις λάθος τιμές

```
ΔΙΑΒΑΣΕ β
ΑΝ β <= 0 Η β > 20 ΤΟΤΕ
    ΓΡΑΨΕ 'Λάθος βαθμός'
ΑΛΛΙΩΣ_ΑΝ β < 9.5 ΤΟΤΕ
    ΓΡΑΨΕ 'ΡΑΝΤΕΒΟΥ ΤΟ ΣΕΠΤΕΜΒΡΗ'
ΑΛΛΙΩΣ_ΑΝ β < 14 ΤΟΤΕ
    ΓΡΑΨΕ 'ΚΑΛΩΣ'
ΑΛΛΙΩΣ_ΑΝ β < 18 ΤΟΤΕ
    ΓΡΑΨΕ 'ΛΙΑΝ ΚΑΛΩΣ'
ΑΛΛΙΩΣ !_ΑΝ β <= 20 ΤΟΤΕ
    ΓΡΑΨΕ 'ΑΡΙΣΤΑ'
ΤΕΛΟΣ_ΑΝ
```

```
ΔΙΑΒΑΣΕ β
ΑΝ β > 0 ΚΑΙ β < 9.5 ΤΟΤΕ
    ΓΡΑΨΕ 'ΡΑΝΤΕΒΟΥ ΤΟ ΣΕΠΤΕΜΒΡΗ'
ΑΛΛΙΩΣ_ΑΝ β >= 9.5 ΚΑΙ β < 14 ΤΟΤΕ
    ΓΡΑΨΕ 'ΚΑΛΩΣ'
ΑΛΛΙΩΣ_ΑΝ β >= 14 ΚΑΙ β < 18 ΤΟΤΕ
    ΓΡΑΨΕ 'ΛΙΑΝ ΚΑΛΩΣ'
ΑΛΛΙΩΣ_ΑΝ β >= 18 ΚΑΙ β <= 20 ΤΟΤΕ
    ΓΡΑΨΕ 'ΑΡΙΣΤΑ'
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Λάθος βαθμός'
ΤΕΛΟΣ_ΑΝ
```

! Ισοδύναμα Προγράμματα ???

```
ΑΝ Χ > 0 ΤΟΤΕ  
  Μ <- 'ΘΕΤΙΚΟΣ'  
  ΓΡΑΨΕ Μ  
ΑΛΛΙΩΣ_ΑΝ Χ < 0 ΤΟΤΕ  
  Μ <- 'ΑΡΝΗΤΙΚΟΣ'  
  ΓΡΑΨΕ Μ  
ΤΕΛΟΣ_ΑΝ
```

ΑΝ Χ = 0 ?

```
ΑΝ Χ > 0 ΤΟΤΕ  
  Μ <- 'ΘΕΤΙΚΟΣ'  
ΑΛΛΙΩΣ_ΑΝ Χ < 0 ΤΟΤΕ  
  Μ <- 'ΑΡΝΗΤΙΚΟΣ'  
ΤΕΛΟΣ_ΑΝ  
ΓΡΑΨΕ Μ
```

Στην περίπτωση που δοθεί αριθμός αρνητικός ή με περισσότερα από 3 ψηφία ο αλγόριθμος πρέπει να εμφανίζει το μήνυμα «Λάθος Δεδομένα».

Αλγόριθμος Ψηφία

Διάβασε x

Αν $x \geq 0$ και $x < 10$ τότε

εμφάνισε 'Μονοψήφιος'

Αλλιώς_αν $x < 100$ τότε

εμφάνισε 'Διψήφιος'

Αλλιώς_αν $x < 1000$ τότε

εμφάνισε 'Τριψήφιος'

Αλλιώς

εμφάνισε 'Λάθος Δεδομένα'

Τέλος_αν

Τέλος Ψηφία

2012 Επαν.
(Α3)

Λογικό Λάθος για $X < 0$



Αν $X > 0$ ΚΑΙ $X < 100$

...

Κατά την εκσφαλμάτωση των **δομών επανάληψης** προσοχή στα εξής:

- στους **Συγκριτικούς** και τους **Λογικούς** τελεστές των **Συνθηκών** επανάληψης ή τερματισμού.
- στην **Αρχικοποίηση** της Συνθήκης.
- στην **Ενημέρωση** της συνθήκης εντός του βρόχου.
- στο Κριτήριο της **Περατότητας**.
- στην **Τελευταία Επανάληψη**.
- στην **Πρώτη Επανάληψη** και στην περίπτωση που ο βρόχος επανάληψης δεν πρέπει να εκτελεστεί **ούτε μία φορά**.

1. Λάθη **Υλοποίησης** $\Rightarrow$ Συντακτικά (*Compiler*)
2. Λάθη **Εκτέλεσης** $\Rightarrow$ Runtime Errors (*User*)
3. Λάθη **Αποτελεσμάτων** $\Rightarrow$ Λογικά Λάθη (*Tester*)

Εκσφαλμάτωση Λογικών Λαθών στις Δομές Επανάληψης

```
1  ΠΡΟΓΡΑΜΜΑ Γινόμενο_μη_μηδενικών_αριθμών
2  ΜΕΤΑΒΛΗΤΕΣ
3      ΠΡΑΓΜΑΤΙΚΕΣ: X, Γινόμενο
4  ΑΡΧΗ
5      Γινόμενο <- 1
6      ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
7          ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"
8          ΔΙΑΒΑΣΕ X
9          Γινόμενο <- X*Γινόμενο
10     ΜΕΧΡΙΣ_ΟΤΟΥ X = 0
11     ΓΡΑΨΕ "Γινόμενο=", Γινόμενο
12 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```


ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"

ΔΙΑΒΑΣΕ X

ΑΝ X $\neq$ 0 ΤΟΤΕ

Γινόμενο $\leftarrow$ X*Γινόμενο

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ X = 0

ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"

ΔΙΑΒΑΣΕ X

ΟΣΟ X $\neq$ 0 ΕΠΑΝΑΛΑΒΕ

Γινόμενο $\leftarrow$ X*Γινόμενο

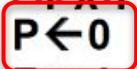
ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"

ΔΙΑΒΑΣΕ X

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

B2. Το ακόλουθο πρόγραμμα έχει σκοπό να διαβάσει 10 θετικούς αριθμούς και να υπολογίζει και να εμφανίζει το γινόμενο όσων από αυτούς είναι πολλαπλάσιοι και του 3 και του 5 (συγχρόνως). Στο πρόγραμμα, όμως, υπάρχουν λάθη.

2017 (B2)

1. Πρόγραμμα Αριθμοί
2. Μεταβλητές
3. Πραγματικές: X   ! Συντακτικό Λάθος !
4. Ακέραιες: P, I
5. Αρχή
6.   ! Λογικό Λάθος !
7. Για I από 1 μέχρι 10
8. Διάβασε X
9. Αν $X \text{ MOD } 3 = 0$  $\text{H } X \text{ MOD } 5 = 0$ τότε
10. $P \leftarrow P * X$  ! Λογικό Λάθος !
11. 
12. Τέλος_επανάληψης  ! Συντακτικό Λάθος !
13. Γράψε P
14. Τέλος_προγράμματος

A4. Δίνεται το παρακάτω πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο διαβάζει ένα σύνολο θετικών αριθμών και υπολογίζει και τυπώνει το γινόμενό τους και τον μέσο όρο τους:

```
1  ΠΡΟΓΡΑΜΜΑ Α4
2  ΜΕΤΑΒΛΗΤΕΣ
3    ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ
4    ΑΚΕΡΑΙΕΣ: ΓΙΝ, ΑΘΡ, ΠΛ
5  ΑΡΧΗ
6    ΠΛ ← 0
7    ΓΙΝ ← 0
8    ΑΘΡ ← '0'
9    ΔΙΑΒΑΣΕ x
10   ΟΣΟ x > 0 ΕΠΑΝΑΛΑΒΕ
11     ΓΙΝ ← ΓΙΝ * x
12     ΑΘΡ ← ΑΘΡ + x
13     ΠΛ ← ΠΛ + 1
14     ΔΙΑΒΑΣΕ x
15   ΤΕΛΟΣ_ΑΝ
16   ΜΟ ← ΑΘΡ / ΠΛ
17   ΓΡΑΨΕ ΜΟ, ΓΙΝ
18 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

2022 (A4)

Συντακτικό (, X)

Λογικό Λάθος (ΓΙΝ ← 1)

Συντακτικό (ΑΘΡ ← 0)

Συντακτικό (~~ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ~~)

Runtime Error (ΑΝ ΠΛ <> 0)

Εκσφαλμάτωση Λογικών Λαθών στους Πίνακες

Προσοχή στα εξής:

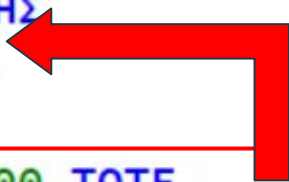
- στο **Μέγεθος** των πινάκων κατά τη **δήλωσή** τους.
- στους **Δείκτες** των πινάκων κατά την **προσπέλασή** τους.
- στη μη **υπέρβαση των ορίων** του πίνακα.
- στην **Αρχικοποίηση** των δεδομένων του πίνακα.

```
1 ΠΡΟΓΡΑΜΜΑ BUGS
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: ΣΥΝΟΛΟ [20], ΑΠΟΥΣΙΕΣ, Ι, Κ
4 ΑΡΧΗ
5   ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 20
6     ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 9
7     ΔΙΑΒΑΣΕ ΑΠΟΥΣΙΕΣ
8     ΣΥΝΟΛΟ[Ι] ← ΣΥΝΟΛΟ[Ι] + ΑΠΟΥΣΙΕΣ
9     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
10    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
11
12    ΑΝ ΣΥΝΟΛΟ[Ι] > 100 ΤΟΤΕ
13      ΓΡΑΨΕ 'ΕΜΕΙΝΕ Ο ΜΑΘΗΤΗΣ: ', Ι
14    ΤΕΛΟΣ_ΑΝ
15
16 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

! ΣΥΝΟΛΟ ΑΠΟΥΣΙΩΝ 9 ΜΗΝΩΝ

! ΓΙΑ 20 ΜΑΘΗΤΕΣ ΜΕ ΟΡΙΟ ΤΙΣ 100

ΣΥΝΟΛΟ[Ι] ← 0 ! ΑΡΧΙΚΟΠΟΙΗΣΗ



Εκσφαλμάτωση Λογικών Λαθών στα Υποπρογράμματα

Προσοχή στα εξής:

- την **Κλήση** των Διαδικασιών και Συναρτήσεων.
- το πέραςμα και την επιστροφή των **Παραμέτρων**.
- την **Δήλωση** των Υποπρογραμμάτων

B2. Δίνονται οι παρακάτω δηλώσεις υποπρογραμμάτων και των παραμέτρων τους:

ΣΥΝΑΡΤΗΣΗ $A(\chi, \psi)$: ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: χ

ΧΑΡΑΚΤΗΡΕΣ: $\psi[10]$

ΔΙΑΔΙΚΑΣΙΑ $B(\chi, \psi, \zeta)$
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: ψ

ΠΡΑΓΜΑΤΙΚΕΣ: χ, ζ

2019 (B2)

Επίσης δίνεται το τμήμα δηλώσεων κύριου προγράμματος:

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: $\kappa, \lambda[10], \mu$

ΠΡΑΓΜΑΤΙΚΕΣ: $\pi, \rho[10], \gamma$

ΧΑΡΑΚΤΗΡΕΣ: $\theta[10], \upsilon$

1. $\pi \leftarrow A(\kappa, \upsilon)$ → ! Πίνακας Χαρακτήρων, ΟΧΙ απλή τιμή!
2. ΚΑΛΕΣΕ $A(\mu, \theta)$ → ! Συνάρτηση, ΟΧΙ Διαδικασία!
3. ΚΑΛΕΣΕ $B(\pi, \mu)$ → ! Τρεις Παράμετροι, ΟΧΙ Δύο!
4. $\upsilon \leftarrow A(\mu, \theta)$ → ! Πραγματική Συνάρτηση, ΟΧΙ Χαρακτήρα!
5. $\kappa \leftarrow B(\pi, \mu, \rho[1])$ → ! Διαδικασία, ΟΧΙ Συνάρτηση!

B2. Δίνονται οι παρακάτω δηλώσεις υποπρογραμμάτων και των παραμέτρων τους:

ΣΥΝΑΡΤΗΣΗ A(χ , ψ): ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: χ

ΧΑΡΑΚΤΗΡΕΣ: $\psi[10]$

ΔΙΑΔΙΚΑΣΙΑ B(χ , ψ , ζ)
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: ψ

ΠΡΑΓΜΑΤΙΚΕΣ: χ , ζ

Επίσης δίνεται το τμήμα δηλώσεων κύριου προγράμματος:

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: κ , $\lambda[10]$, μ

ΠΡΑΓΜΑΤΙΚΕΣ: π , $\rho[10]$, γ

ΧΑΡΑΚΤΗΡΕΣ: $\theta[10]$, υ

1. $\pi \leftarrow A(\kappa, \upsilon)$
2. ΚΑΛΕΣΕ A(μ , θ)
3. ΚΑΛΕΣΕ B(π , μ)
4. $\upsilon \leftarrow A(\mu, \theta)$
5. $\kappa \leftarrow B(\pi, \mu, \rho[1])$

$\pi \leftarrow A(\kappa, \theta)$

$\pi \leftarrow A(\mu, \theta)$

ΚΑΛΕΣΕ B(π, μ, γ)

$\gamma \leftarrow A(\mu, \theta)$

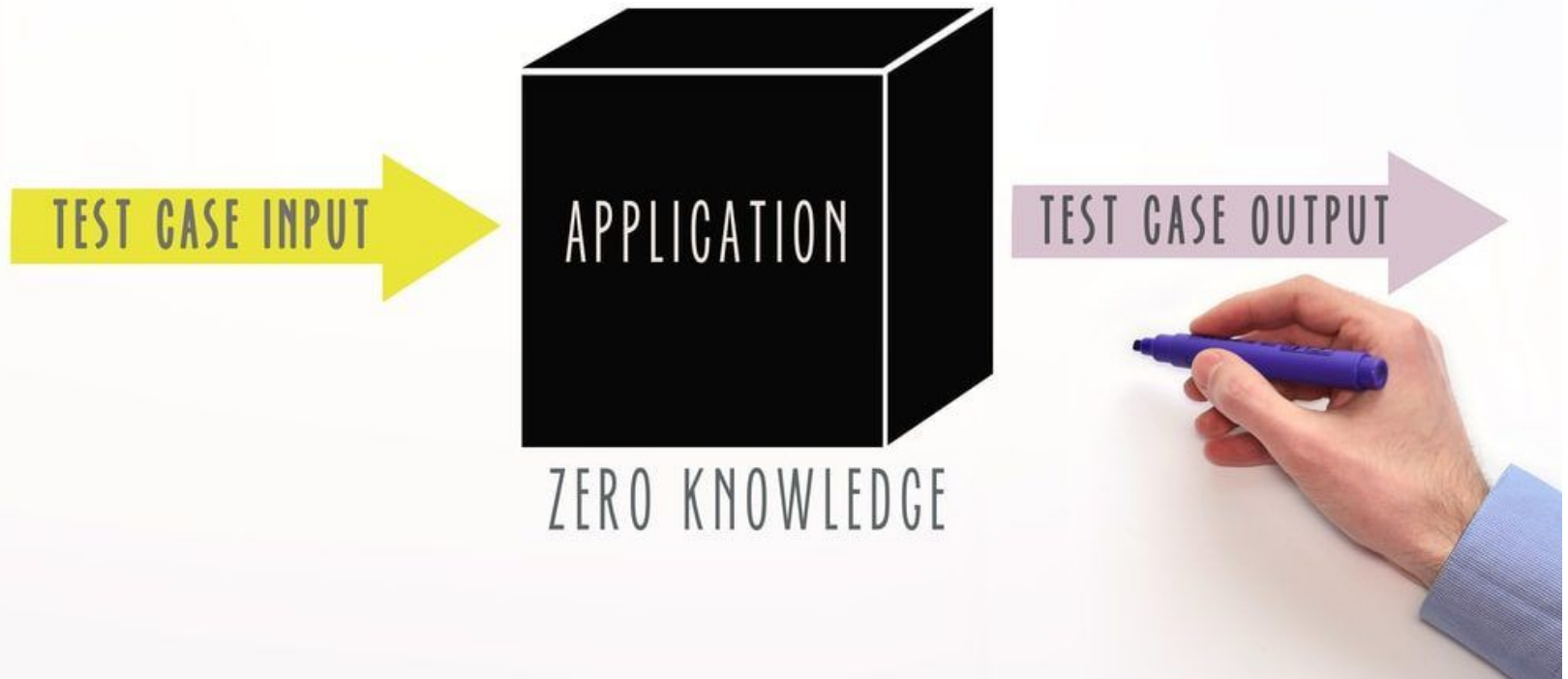
ΚΑΛΕΣΕ B($\pi, \mu, \rho[1]$)

Κατά το στάδιο της υλοποίησης πολύ συχνά πραγματοποιούνται **ανασκοπήσεις κώδικα** από ομάδες προγραμματιστών με τη χρήση **σεναρίων ελέγχου**.

Ένα σενάριο ελέγχου (**test case**) περιγράφει τα **δεδομένα εισόδου** ολόκληρου του προγράμματος ή τμήματος του (υποπρόγραμμα) και τα **αναμενόμενα αποτελέσματα**.

Τα σενάρια ελέγχου εκτελούνται, είτε σε **πραγματικό περιβάλλον** προγραμματισμού είτε εικονικά με δημιουργία **πίνακα τιμών** των μεταβλητών.

BLACK-BOX TESTING



Μια δημοφιλή τεχνική ελέγχου είναι ο **έλεγχος μαύρου κουτιού** (black-box testing).

Ονομάζεται έτσι επειδή τα δεδομένα εισόδου στα σενάρια ελέγχου προκύπτουν από τις **προδιαγραφές** του προγράμματος, **αγνοώντας** εντελώς τον **κώδικα**.

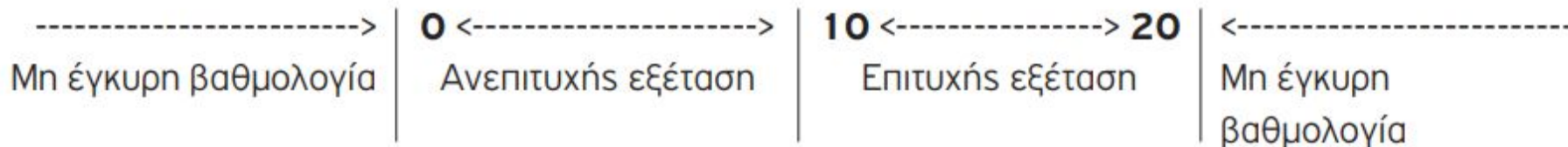
Ιδανικά θα θέλαμε να ελέγξουμε **όλες τις τιμές** εισόδου και όλα τα πιθανά αποτελέσματα. Επειδή όμως αυτό είναι **αδύνατο**, προσπαθούμε να βρούμε **αντιπροσωπευτικές τιμές** για τα δεδομένα εισόδου που θα παράγουν αντιπροσωπευτικά αποτελέσματα.



Παράδειγμα 10 – Βαθμολογία γραπτής εξέτασης

Η βαθμολογία στις γραπτές δοκιμασίες τετραμήνου στο Λύκειο δίνεται με ακέραιους αριθμούς στην κλίμακα από 0 έως και 20. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάσει τη βαθμολογία σε μια γραπτή δοκιμασία και στη συνέχεια να εμφανίζει μήνυμα «Επιτυχής εξέταση», αν η βαθμολογία είναι τουλάχιστον 10, και μήνυμα «Ανεπιτυχής εξέταση» αν η βαθμολογία είναι μικρότερη από 10. Σε περίπτωση που δοθεί τιμή εκτός του διαστήματος 0-20 να εμφανίζεται μήνυμα λάθους «Μη έγκυρη βαθμολογία».

1ο Βήμα: Δημιουργία Ισοδύναμων Διαστημάτων τιμών (*equivalence partitioning*)





Παράδειγμα 10 – Βαθμολογία γραπτής εξέτασης

Η βαθμολογία στις γραπτές δοκιμασίες τετραμήνου στο Λύκειο δίνεται με ακέραιους αριθμούς στην κλίμακα από 0 έως και 20. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάσει τη βαθμολογία σε μια γραπτή δοκιμασία και στη συνέχεια να εμφανίζει μήνυμα «Επιτυχής εξέταση», αν η βαθμολογία είναι τουλάχιστον 10, και μήνυμα «Ανεπιτυχής εξέταση» αν η βαθμολογία είναι μικρότερη από 10. Σε περίπτωση που δοθεί τιμή εκτός του διαστήματος 0-20 να εμφανίζεται μήνυμα λάθους «Μη έγκυρη βαθμολογία».

2ο Βήμα: Έλεγχος των Ακραίων Τιμών κάθε διαστήματος (*boundary value analysis*)



3ο Βήμα: Δημιουργία Σεναρίων Ελέγχου

A/A	Είσοδος	Αναμενόμενο αποτέλεσμα	Περίπτωση που ελέγχεται
1	-1	Μη έγκυρη βαθμολογία	Άνω άκρο διαστήματος βαθμός < 0
2	0	Ανεπιτυχής εξέταση	Κάτω άκρο διαστήματος $0 \leq \text{βαθμός} < 10$
3	9	Ανεπιτυχής εξέταση	Άνω άκρο διαστήματος $0 \leq \text{βαθμός} < 10$
4	10	Επιτυχής εξέταση	Κάτω άκρο διαστήματος $10 \leq \text{βαθμός} \leq 20$
5	20	Επιτυχής εξέταση	Άνω άκρο διαστήματος $10 \leq \text{βαθμός} \leq 20$
6	21	Μη έγκυρη βαθμολογία	Κάτω άκρο διαστήματος βαθμός > 20

Ο έλεγχος ενός προγράμματος είναι διαδικασία που απαιτεί **χρόνο**, τόσο για τον **σχεδιασμό** όσο και για την **υλοποίηση** και εκτέλεση των **σεναρίων ελέγχου**, ενώ αν προκύψουν λάθη θα απαιτηθεί επιπλέον χρόνος για τον εντοπισμό και τη διόρθωση τους.

Γι' αυτό και η φάση του ελέγχου και της **εκσφαλμάτωσης** αποτελούν **σημαντικό ποσοστό** του συνολικού **χρόνου ανάπτυξης** ενός προγράμματος.

Στις μεγάλες εταιρείες λογισμικού υπάρχουν **εξειδικευμένες ομάδες** προγραμματιστών που ασχολούνται αποκλειστικά με αυτό.

Ε.2: Χαρακτηρίστε τις παρακάτω προτάσεις ως Σωστές ή Λάθος. Στην περίπτωση που πιστεύετε ότι είναι λανθασμένες δικαιολογήστε την επιλογή σας και σκεφτείτε ποια θα μπορούσε να είναι η αντίστοιχη σωστή πρόταση.

α/α	Προτάσεις	Σ	Λ
1	Στον έλεγχο «μαύρου κουτιού» τα σενάρια ελέγχου βασίζονται στον κώδικα του προγράμματος.	☐	☒
2	Τα σενάρια ελέγχου περιλαμβάνουν και μη έγκυρες τιμές εισόδου.	☒	☐
3	Κατά τον έλεγχο ακραίων τιμών ελέγχονται τυχαίες τιμές από κάθε διάστημα της εισόδου.	☐	☒
4	Ο έλεγχος «μαύρου κουτιού» μπορεί να εφαρμοστεί και σε υποπρογράμματα.	☒	☐

<https://forms.gle/JrBrCdePBFuKFt8U6>

E.3: Στο Λύκειο, για την ετήσια επίδοση των μαθητών και μαθητριών χρησιμοποιείται ο γενικός μέσος όρος (Γ.Μ.Ο.) που είναι πραγματικός αριθμός από 0 μέχρι και 20 με ακρίβεια ενός δεκαδικού ψηφίου. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο να διαβάζει έναν πραγματικό αριθμό που να αντιστοιχεί στον Γ.Μ.Ο. ενός μαθητή ή μιας μαθήτριας. Αν ο Γ.Μ.Ο. είναι τουλάχιστον 9,5 να εμφανίζεται μήνυμα «Προάγεται», διαφορετικά να εμφανίζεται μήνυμα «Παραπέμπεται σε επανεξέταση». Αν δοθεί τιμή εκτός του διαστήματος 0-20, να εμφανίζεται μήνυμα «Μη έγκυρος Γ.Μ.Ο.».

Σύμφωνα με τις παραπάνω προδιαγραφές, να πραγματοποιήσετε έλεγχο ακραίων τιμών δημιουργώντας τα κατάλληλα σενάρια.

E.4: Μια τουριστική επιχείρηση διαθέτει διαμερίσματα για βραχυχρόνια μίσθωση σύμφωνα με την ακόλουθη τιμολογιακή πολιτική: για διαμονή έως και 3 ημέρες 50€/ημέρα, για διαμονή έως και 7 ημέρες 47€/ημέρα, για διαμονή έως και 20 ημέρες 42€/ημέρα. Ο μέγιστος χρόνος μίσθωσης κάθε διαμερίσματος είναι 20 ημέρες. Να αναπτύξετε συνάρτηση σε ΓΛΩΣΣΑ, η οποία να δέχεται ως είσοδο το πλήθος των ημερών διαμονής και να επιστρέφει τη συνολική χρέωση. Σε περίπτωση που δοθεί είσοδος εκτός του διαστήματος 1-20 η συνάρτηση να επιστρέφει την τιμή - 1.

Να δημιουργήσετε κατάλληλα σενάρια με βάση τις παραπάνω προδιαγραφές, για να πραγματοποιήσετε έλεγχο ακραίων τιμών.