

Κεφ. 13 - Εκσφαλμάτωση Προγράμματος (Συμπληρωματικό Εκπαιδευτικό Υλικό Ενότητα 5)

Καραμασούνας Πολύκαρπος

5.1 Κατηγορίες Λαθών

5.1.1 Συντακτικά λάθη

```
1  ΠΡΟΓΡΑΜΜΑ Αθροισμα_θετικών_αριθμών
2  ΜΕΤΑΒΛΗΤΕΣ
3      ΑΚΕΡΑΙΕΣ: Σ,Χ,Ι
4      Σ <- 0
5      ΓΙΑ Ι ΑΠΟ 1 ΕΩΣ 10
6          ΓΡΑΨΕ 'Δώσε έναν ακέραιο αριθμό'
7          ΔΙΑΒΑΣΕ Χ
8          ΑΝ Χ > 0
9              Σ <- Σ + Χ
10         ΤΕΛΟΣ_ΑΝ
11     ΓΡΑΨΕ 'Σ=', Σ
12 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```



```
1  ΠΡΟΓΡΑΜΜΑ Αθροισμα_θετικών_αριθμών
2  ΜΕΤΑΒΛΗΤΕΣ
3      ΑΚΕΡΑΙΕΣ: Σ,Χ,Ι
4  ΑΡΧΗ
5      Σ <- 0
6      ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
7          ΓΡΑΨΕ 'Δώσε έναν ακέραιο αριθμό'
8          ΔΙΑΒΑΣΕ Χ
9          ΑΝ Χ > 0 ΤΟΤΕ
10             Σ <- Σ + Χ
11         ΤΕΛΟΣ_ΑΝ
12     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
13     ΓΡΑΨΕ 'Σ=', Σ
14 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

```
1  ΠΡΟΓΡΑΜΜΑ Αθροισμα_θετικών_αριθμών
2  ΜΕΤΑΒΛΗΤΕΣ
3      ΠΡΑΓΜΑΤΙΚΕΣ: Χ
4  ΑΡΧΗ
5      ΓΡΑΨΕ 'Χ=' Προσπαθήστε να εντοπίσετε τα συντακτικά λάθη
6      ΔΙΑΒΑΣΕ Χ
7      Σ <- 0
8      ΟΣΟ Χ > 0
9          Σ <- Σ + Χ
10         ΓΡΑΨΕ 'Χ='
11         ΔΙΑΒΑΣΕ Χ
12     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
13     ΕΑΝ Σ > 0 ΤΟΤΕ
14         ΓΡΑΨΕ 'Σ=', Σ
15     ΑΛΛΙΩΣ
16         ΓΡΑΨΕ 'Δεν δόθηκαν αυστηρά θετικοί αριθμοί'
17     ΤΕΛΟΣ_ΑΝ
18 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

```
ΠΡΟΓΡΑΜΜΑ Αθροισμα_θετικών_αριθμών
ΜΕΤΑΒΛΗΤΕΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: Χ, Σ
ΑΡΧΗ
    ΓΡΑΨΕ 'Χ='
    ΔΙΑΒΑΣΕ Χ
    Σ <- 0
    ΟΣΟ Χ > 0 ΕΠΑΝΑΛΑΒΕ
        Σ <- Σ + Χ
        ΓΡΑΨΕ 'Χ='
        ΔΙΑΒΑΣΕ Χ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΝ Σ > 0 ΤΟΤΕ
        ΓΡΑΨΕ 'Σ=', Σ
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 'Δεν δόθηκαν αυστηρά θετικοί αριθμοί'
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

5.1.2 Λάθη κατά την εκτέλεση – Λάθη που οδηγούν σε αντικανονικό τερματισμό του προγράμματος

π.χ. αν επιχειρήσουμε να διαιρέσουμε με το μηδέν ή αν κατά την ανάγνωση ενός ακεραίου αριθμού εισαχθεί ένα γράμμα



5.1.3 Λογικά λάθη

```
1  ΠΡΟΓΡΑΜΜΑ Μέσος_όρος_ύψους
2  ΜΕΤΑΒΛΗΤΕΣ
3      ΑΚΕΡΑΙΕΣ: πλήθος
4      ΠΡΑΓΜΑΤΙΚΕΣ: ύψος, Σ, ΜΟ
5  ΑΡΧΗ
6      Σ <- 0
7      πλήθος <- 0
8      ΓΡΑΨΕ 'Δώσε ύψος:'
9      ΔΙΑΒΑΣΕ ύψος
10     ΟΣΟ ύψος > 0 ΕΠΑΝΑΛΑΒΕ
11         Σ <- Σ + ύψος
12         πλήθος <- πλήθος + 1
13         ΓΡΑΨΕ 'Δώσε ύψος:'
14         ΔΙΑΒΑΣΕ ύψος
15     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
16     ΜΟ <- Σ/πλήθος
17     ΓΡΑΨΕ 'Μέσος όρος ύψους:', ΜΟ
18 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

```
1  ΠΡΟΓΡΑΜΜΑ Αγορά_τετραδίων
2  ΜΕΤΑΒΛΗΤΕΣ
3      ΠΡΑΓΜΑΤΙΚΕΣ: τιμή, πλήθος, διαθέσιμο_ποσό, οφειλόμενο
4  ΑΡΧΗ
5      ΓΡΑΨΕ 'Δώσε την τιμή του τετραδίου:'
6      ΔΙΑΒΑΣΕ τιμή
7      ΓΡΑΨΕ 'Δώσε το πλήθος των τετραδίων'
8      ΔΙΑΒΑΣΕ πλήθος
9      ΓΡΑΨΕ 'Δώσε το διαθέσιμο ποσό'
10     ΔΙΑΒΑΣΕ διαθέσιμο_ποσό
11     οφειλόμενο_ποσό <- τιμή * πλήθος
12     ΑΝ οφειλόμενο_ποσό <= διαθέσιμο_ποσό ΤΟΤΕ
13         ΓΡΑΨΕ 'Η αγορά είναι εφικτή'
14     ΑΛΛΙΩΣ
15         ΓΡΑΨΕ 'Η αγορά δεν είναι εφικτή'
16     ΤΕΛΟΣ_ΑΝ
17 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

5.2 Εκσφαλμάτωση (debugging)

Η διαδικασία ελέγχου, εντοπισμού και διόρθωσης των σφαλμάτων ενός προγράμματος καλείται εκσφαλμάτωση (debugging). Στόχος της διαδικασίας εκσφαλμάτωσης είναι ο εντοπισμός των σημείων του προγράμματος που προκαλούν προβλήματα στη λειτουργία του.

5.2.1 Εκσφαλμάτωση λογικών λαθών στις δομές επιλογής

Half of programming is coding.
The other 90% is debugging.

«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει την ηλικία ενός επιβάτη αστικού λεωφορείου σε έτη και ανάλογα με την τιμή της ηλικίας του, να υπολογίζει το αντίτιμο του εισιτηρίου που πρέπει να πληρώσει. Εάν έχει συμπληρώσει το 18^ο έτος της ηλικίας του, ο επιβάτης πληρώνει κανονικό εισιτήριο 1€. Διαφορετικά, πληρώνει μειωμένο εισιτήριο που αντιστοιχεί στο 50% του κανονικού εισιτηρίου. Το αντίτιμο του εισιτηρίου να εμφανίζεται στην οθόνη».

Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.7]). Προσπαθήστε να εντοπίσετε τυχόν λογικά λάθη που οδηγούν σε λανθασμένα αποτελέσματα και να προτείνετε διορθώσεις.

Διασταυρώστε την απάντησή σας με αυτή που ακολουθεί.

When you get 5 additional bugs after fixing 1:



```
1  ΠΡΟΓΡΑΜΜΑ Αντίτιμο_εισιτηρίου
2  ΣΤΑΘΕΡΕΣ
3      Κανονικό_εισιτήριο = 1
4  ΜΕΤΑΒΛΗΤΕΣ
5      ΑΚΕΡΑΙΕΣ: ηλικία
6      ΠΡΑΓΜΑΤΙΚΕΣ: Αντίτιμο
7  ΑΡΧΗ
8      ΓΡΑΨΕ 'Δώσε ηλικία'
9      ΔΙΑΒΑΣΕ ηλικία
10     ΑΝ ηλικία > 18 ΤΟΤΕ
11         Αντίτιμο <- Κανονικό_εισιτήριο
12     ΑΛΛΙΩΣ
13         Αντίτιμο <- 0.5*Κανονικό_εισιτήριο
14     ΤΕΛΟΣ_ΑΝ
15     ΓΡΑΨΕ 'Το αντίτιμο του εισιτηρίου είναι:', Αντίτιμο
16 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

5.2 Εκσφαλμάτωση (debugging)

5.2.1 Εκσφαλμάτωση λογικών λαθών στις δομές επιλογής

«Ένας οργανισμός ύδρευσης υπολογίζει την οφειλή ενός καταναλωτή για ένα τετράμηνο ως εξής: Χρεώνει το πάγιο ποσό 8€, τα πρώτα δέκα κυβικά μέτρα χρεώνονται με 0.4€/κυβικό. Τα δέκα επόμενα κυβικά επιπλέον των δέκα πρώτων χρεώνονται με 0.5€/κυβικό. Κάθε κυβικό επιπλέον των είκοσι πρώτων χρεώνεται με 0.6€/κυβικό. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει τα κυβικά που κατανάλωσε ένα νοικοκυριό σε ένα τετράμηνο και να υπολογίζει και να εμφανίζει το ποσό της οφειλής του νοικοκυριού».

```
1  ΠΡΟΓΡΑΜΜΑ Χρέωση_ύδρευσης
2  ΣΤΑΘΕΡΕΣ
3      πάγιο = 8
4  ΜΕΤΑΒΛΗΤΕΣ
5      ΠΡΑΓΜΑΤΙΚΕΣ: κυβικά, οφειλή
6  ΑΡΧΗ
7      18      οφειλή <- 8 + 10*0.4 + 10*0.5 + (κυβικά - 20)*0.6
8      ! Ανάγνωση δεδομένων
9      ΓΡΑΨΕ 'Δώσε τα κυβικά που καταναλώθηκαν'
10     ΔΙΑΒΑΣΕ κυβικά
11     ! Υπολογισμός κλιμακωτής χρέωσης
12     ΑΝ κυβικά < 0 ΤΟΤΕ
13         ΓΡΑΨΕ 'Μη αποδεκτή τιμή.'
14     ΑΛΛΙΩΣ_ΑΝ κυβικά < 10 ΤΟΤΕ
15         οφειλή <- 8 + κυβικά*0.4
16     ΑΛΛΙΩΣ_ΑΝ κυβικά <= 20 ΤΟΤΕ
17         οφειλή <- 8 + 10*0.4 + (κυβικά - 10)*0.5
18     ΑΛΛΙΩΣ
19         οφειλή <- 8 + 10*0.4 + 10*0.5 + (κυβικά - 20)*0.6
20     ΤΕΛΟΣ_ΑΝ
21     ! Εμφάνιση αποτελέσματος
22     ΓΡΑΨΕ 'Οφειλή=', οφειλή
23 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Διορθώσεις

```
20     ! Εμφάνιση αποτελέσματος
21     ΓΡΑΨΕ 'Οφειλή=', οφειλή
```

Να μεταφερθούν μετά τις γραμμές:

14, 16, 18

ή

Αν κυβικά >= 0 τότε

ΓΡΑΨΕ 'Οφειλή=', οφειλή

Τέλος_Αν



5.2 Εκσφαλμάτωση (debugging)

5.2.1 Εκσφαλμάτωση λογικών λαθών στις δομές επιλογής

```
1  ΠΡΟΓΡΑΜΜΑ Υπολογισμός_αντίτιμου_εισιτηρίου
2  ΣΤΑΘΕΡΕΣ
3      Κανονικό_εισιτήριο = 1
4      Μειωμένο_εισιτήριο = 0.5
5  ΜΕΤΑΒΛΗΤΕΣ
6      ΑΚΕΡΑΙΕΣ: κατηγορία, πλήθος
7      ΠΡΑΓΜΑΤΙΚΕΣ: Αντίτιμο
8  ΑΡΧΗ
9      ΓΡΑΨΕ '1. Κανονικό εισιτήριο (1 ευρώ) '
10     ΓΡΑΨΕ '2. Μειωμένο εισιτήριο (0.5 ευρώ) '
11     ΓΡΑΨΕ 'Δώσε τον αριθμό της κατηγορίας του εισιτηρίου:'
12     ΔΙΑΒΑΣΕ κατηγορία
13     ΓΡΑΨΕ 'Δώσε το πλήθος των εισιτηρίων'
14     ΔΙΑΒΑΣΕ πλήθος
15     ΑΝ κατηγορία = 1 ΤΟΤΕ
16         Αντίτιμο <- πλήθος*Κανονικό_εισιτήριο
17         ΓΡΑΨΕ 'Αντίτιμο=', Αντίτιμο
18     ΑΛΛΙΩΣ_ΑΝ κατηγορία = 2 ΤΟΤΕ
19         Αντίτιμο <- Μειωμένο_εισιτήριο
20         ΓΡΑΨΕ 'Αντίτιμο=', Αντίτιμο
21     ΑΛΛΙΩΣ
22         ΓΡΑΨΕ 'Μη αποδεκτή κατηγορία εισιτηρίου'
23     ΤΕΛΟΣ_ΑΝ
24 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει το πλήθος των εισιτηρίων που θέλουμε να εκδώσουμε και την κατηγορία τους. Στη συνέχεια να υπολογίζει και να εμφανίζει το κόστος των εισιτηρίων. Υπάρχουν οι εξής κατηγορίες εισιτηρίων:

1. Κανονικό εισιτήριο (1€)
2. Μειωμένο εισιτήριο (0,5€)

Σε μία εκτέλεση του προγράμματος μπορούν να ληφθούν υπόψη μόνο εισιτήρια μίας κατηγορίας. Θεωρείστε ότι ο χρήστης δίνει μη αρνητικές τιμές για το πλήθος των εισιτηρίων».



5.2 Εκσφαλμάτωση (debugging)

5.2.2 Εκσφαλμάτωση λογικών λαθών στις δομές επανάληψης

Σε μια δομή επανάληψης μπορεί να εμφανιστούν λογικά λάθη που σχετίζονται με:

- τη συνθήκη επανάληψης ή τερματισμού,
- την αρχικοποίηση της συνθήκης,
- την ενημέρωση της συνθήκης εντός του βρόχου επανάληψης,
- τις εντολές που περιλαμβάνονται εντός του βρόχου.

«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει έναν βαθμό τετραμήνου στην εικοσάβαθμη κλίμακα. Να εκτελείται έλεγχος αποδεκτής τιμής.»



```
1  ΠΡΟΓΡΑΜΜΑ Ανάγνωση_βαθμολογίας
2  ΜΕΤΑΒΛΗΤΕΣ
3    ΑΚΕΡΑΙΕΣ: Βαθμός
4  ΑΡΧΗ
5    ΓΡΑΨΕ "Δώσε βαθμό στην εικοσάβαθμη κλίμακα"
6    ΔΙΑΒΑΣΕ Βαθμός
7    ΟΣΟ Βαθμός < 0 ΚΑΙ Βαθμός > 20 ΕΠΑΝΑΛΑΒΕ
8      ΓΡΑΨΕ "Μη αποδεκτή τιμή"
9      ΓΡΑΨΕ "Δώσε βαθμό στην εικοσάβαθμη κλίμακα"
10     ΔΙΑΒΑΣΕ Βαθμός
11   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
12   ΓΡΑΨΕ "Καταχωρήθηκε ο βαθμός της εικοσάβαθμης κλίμακας ", Βαθμός
13 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

5.2 Εκσφαλμάτωση (debugging)

5.2.2 Εκσφαλμάτωση λογικών λαθών στις δομές επανάληψης

«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει από το πληκτρολόγιο αριθμούς διάφορους του μηδενός, να υπολογίζει το γινόμενό τους και στο τέλος να το εμφανίζει. Αν δε δοθούν αριθμοί, να εμφανίζει τον αριθμό 1».

```
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
```

```
6  ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"
7  ΔΙΑΒΑΣΕ X
8  ΟΣΟ X <> 0 ΕΠΑΝΑΛΑΒΕ
9  Γινόμενο <- X*Γινόμενο
10 ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"
11 ΔΙΑΒΑΣΕ X
12 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ή

```
6  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
7  ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"
8  ΔΙΑΒΑΣΕ X
9  ΑΝ X <> 0 ΤΟΤΕ
10 Γινόμενο <- X*Γινόμενο
11 ΤΕΛΟΣ_ΑΝ
12 ΜΕΧΡΙΣ_ΟΤΟΥ X = 0
```



5.2 Εκσφαλμάτωση (debugging)

5.2.2 Εκσφαλμάτωση λογικών λαθών στις δομές επανάληψης

«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάσει βαθμούς μαθητών, να υπολογίζει τον μέσο όρο τους και στο τέλος να τον εμφανίζει. Το πρόγραμμα να αποδέχεται μόνο τιμές μεγαλύτερες ή ίσες του μηδενός για τους βαθμούς. Μόλις διαβάσει κάποιον αριθμό μικρότερο του μηδενός, να σταματήσει την ανάγνωση των βαθμών. Θεωρούμε ότι από το πληκτρολόγιο δε δίνονται τιμές μεγαλύτερες από τον μέγιστο επιτρεπτό βαθμό».

```
1  ΠΡΟΓΡΑΜΜΑ ΜΟ_Βαθμολογίας
2  ΜΕΤΑΒΛΗΤΕΣ
3      ΑΚΕΡΑΙΕΣ: Βαθμός, Πλήθος, Άθροισμα
4      ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ
5  ΑΡΧΗ
6      Άθροισμα <- 0
7      Πλήθος <- 0
8      ΓΡΑΨΕ "Βαθμός:"
9      ΔΙΑΒΑΣΕ Βαθμός
10     ΟΣΟ Βαθμός >= 0 ΕΠΑΝΑΛΑΒΕ
11         ΓΡΑΨΕ "Βαθμός:"
12         ΔΙΑΒΑΣΕ Βαθμός
13         Άθροισμα <- Άθροισμα + Βαθμός
14     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
15     ΜΟ <- Άθροισμα/Πλήθος
16     ΓΡΑΨΕ "Μέσος όρος βαθμών=", ΜΟ
17 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```



Οι γραμμές 11,12 να μπουν μετά τη 13
Οι γραμμές 15,16 να μπουν σε Αν (Πλήθος <> 0)
Πλήθος ← Πλήθος + 1 (μέσα στην Όσο)

5.2 Εκσφαλμάτωση (debugging)

5.2.3 Εκσφαλμάτωση λογικών λαθών σε πίνακες

«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει από το πληκτρολόγιο τα μηνιαία έσοδα ενός καταστήματος για το πρώτο εξάμηνο και να τα καταχωρεί σε πίνακα. Στη συνέχεια για τους μήνες Φεβρουάριο, Μάρτιο, Απρίλιο, Μάιο και Ιούνιο να ελέγχει αν είχαν περισσότερα έσοδα από τον ακριβώς προηγούμενο μήνα και να εμφανίζει κατάλληλο μήνυμα που θα δηλώνει την ύπαρξη αύξησης».

```
1  ΠΡΟΓΡΑΜΜΑ ΕΣΟΔΑ_ΚΑΤΑΣΤΗΜΑΤΟΣ
2  ΜΕΤΑΒΛΗΤΕΣ
3      ΠΡΑΓΜΑΤΙΚΕΣ : ΕΣΟΔΑ[6]
4      ΑΚΕΡΑΙΕΣ : Ι
5  ΑΡΧΗ
6      ! Επανάληψη ανάγνωσης εσόδων
7      ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6
8          ΓΡΑΨΕ "Δώσε τα έσοδα του ", Ι, "ου μήνα"
9          ΔΙΑΒΑΣΕ ΕΣΟΔΑ[Ι]
10     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
11     ! Επανάληψη ελέγχου αυξητικής τάσης
12     ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6
13         ΑΝ ΕΣΟΔΑ[Ι] <= ΕΣΟΔΑ[Ι + 1] ΤΟΤΕ
14             ΓΡΑΨΕ "Ο ", Ι, "ος μήνας ΑΥΞΗΣΗ"
15         ΤΕΛΟΣ_ΑΝ
16     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
17 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```



Γραμμή 12: ΜΕΧΡΙ 5

Γραμμή 13: <

Γραμμή 14: Ι+1

5.2 Εκσφαλμάτωση (debugging)

5.2.3 Εκσφαλμάτωση λογικών λαθών σε πίνακες

«Μια επιχείρηση έχει τρία υποκαταστήματα. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει από το πληκτρολόγιο τα έσοδα κάθε υποκαταστήματος ανά τρίμηνο ενός έτους και να τα καταχωρεί σε πίνακα. Για κάθε υποκατάστημα να υπολογίζει και να εμφανίζει τα ετήσια έσοδα».

```
1  ΠΡΟΓΡΑΜΜΑ Έσοδα_υποκαταστημάτων
2  ΜΕΤΑΒΛΗΤΕΣ
3  ΠΡΑΓΜΑΤΙΚΕΣ: ΕΣΟΔΑ[3, 4], Αθροισμα
4  ΑΚΕΡΑΙΕΣ: Ι, Κ
5  ΑΡΧΗ
6  Αθροισμα <- 0
7  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 3
8  ΓΡΑΨΕ 'Δώσε τα έσοδα των τριμήνων του ', Ι, 'ου υποκαταστήματος:'
9  ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 4
10  ΔΙΑΒΑΣΕ ΕΣΟΔΑ[Ι, Κ]
11  Αθροισμα <- Αθροισμα + ΕΣΟΔΑ[Ι, Κ]
12  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
13  ΓΡΑΨΕ 'Ετήσια έσοδα:', Αθροισμα
14  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
15  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Η γραμμή 6 να μεταφερθεί μετά την 7



5.2 Εκσφαλμάτωση (debugging)

5.2.4 Εκσφαλμάτωση λογικών λαθών στα υποπρογράμματα

«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάσει τα μηνιαία έσοδα και τα μηνιαία έξοδα μιας επιχείρησης για το πρώτο εξάμηνο του έτους και να τα καταχωρεί σε πίνακες. Μέσω συνάρτησης να υπολογίζει το πλήθος των μηνών που είχαν ζημία, δηλαδή τα έσοδα ήταν λιγότερα από τα έξοδα. Τέλος, να εμφανίζει το πλήθος των μηνών που είχαν ζημία».

```
1  ΠΡΟΓΡΑΜΜΑ ΕΛΕΓΧΟΣ_ΖΗΜΙΑΣ
2  ΜΕΤΑΒΛΗΤΕΣ
3  ΑΚΕΡΑΙΕΣ: Ι, Πλήθος, Μήνες_με_ζημία
4  ΠΡΑΓΜΑΤΙΚΕΣ: ΕΣΟΔΑ[6], ΕΞΟΔΑ[6]
5  ΑΡΧΗ
6  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6
7  ΓΡΑΨΕ 'Δώσε τα έσοδα του ', Ι, 'ου μήνα:'
8  ΔΙΑΒΑΣΕ ΕΣΟΔΑ[Ι]
9  ΓΡΑΨΕ 'Δώσε τα έξοδα του ', Ι, 'ου μήνα:'
10 ΔΙΑΒΑΣΕ ΕΞΟΔΑ[Ι]
11 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
12 Μήνες_με_ζημία <- Υπολογισμός(ΕΣΟΔΑ, ΕΞΟΔΑ)
13 ΓΡΑΨΕ 'Μήνες που είχαν ζημία:', Μήνες_με_ζημία
14 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
15
```

```
«Μήνες_με_ζημία <- Υπολογισμός(ΕΣΟΔΑ, ΕΞΟΔΑ)»
```

ή

```
«ΣΥΝΑΡΤΗΣΗ Υπολογισμός(Πίνακας2, Πίνακας1): ΑΚΕΡΑΙΑ»
```

ή

```
«ΑΝ Πίνακας1[Ι] < Πίνακας2[Ι] ΤΟΤΕ»
```

```
16 ΣΥΝΑΡΤΗΣΗ Υπολογισμός(Πίνακας1, Πίνακας2): ΑΚΕΡΑΙΑ
17 ΜΕΤΑΒΛΗΤΕΣ
18 ΠΡΑΓΜΑΤΙΚΕΣ: Πίνακας1[6], Πίνακας2[6]
19 ΑΚΕΡΑΙΕΣ: Ι, Πλήθος
20 ΑΡΧΗ
21 Πλήθος <- 0
22 ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6
23 ΑΝ Πίνακας2[Ι] < Πίνακας1[Ι] ΤΟΤΕ
24 Πλήθος <- Πλήθος + 1
25 ΤΕΛΟΣ_ΑΝ
26 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
27 Υπολογισμός <- Πλήθος
28 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```



Προαιρετικά: καλύτερη ονοματολογία των Πίνακας1, Πίνακας2 σε ΕΞΟΔΑ, ΕΣΟΔΑ

5.2 Εκσφαλμάτωση (debugging)

5.2.5 Μέθοδος ελέγχου «Μαύρο Κουτί»



Ένα **σενάριο ελέγχου (test case)** περιγράφει τα δεδομένα εισόδου ολόκληρου του προγράμματος ή τμήματος του προγράμματος (διαδικασία, συνάρτηση) και τα αναμενόμενα αποτελέσματα. Τα σενάρια ελέγχου εκτελούνται, είτε σε πραγματικό περιβάλλον προγραμματισμού είτε εικονικά με δημιουργία πίνακα τιμών των μεταβλητών. Σε περίπτωση αποκλίσεων μεταξύ των αναμενόμενων και των πραγματικών αποτελεσμάτων, υπάρχει λάθος το οποίο πρέπει να εντοπιστεί και να διορθωθεί.

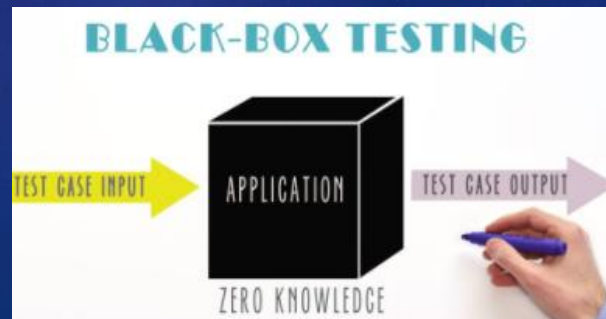


Μια δημοφιλής τεχνική ελέγχου είναι ο **έλεγχος μαύρου κουτιού (black-box testing)**. Ονομάζεται έτσι επειδή τα δεδομένα εισόδου στα σενάρια ελέγχου προκύπτουν από τις προδιαγραφές του προγράμματος, αγνοώντας εντελώς τον κώδικα. Δηλαδή το πρόγραμμα μοιάζει σαν να βρίσκεται μέσα σε ένα μαύρο κουτί που κρύβει το περιεχόμενό του.

Είναι αδύνατο να ελέγξουμε όλες τις τιμές εισόδου.

Βρίσκουμε αντιπροσωπευτικές τιμές εισόδου για τη δημιουργία ισοδύναμων διαστημάτων τιμών (equivalence partitioning).

Καλύτερη στρατηγική: έλεγχος των ακραίων τιμών κάθε διαστήματος (boundary value analysis)



5.2.5 Μέθοδος ελέγχου «Μαύρο Κουτί»

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 διαβάζει 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».

Βήμα 1ο: Δημιουργία ισοδύναμων διαστημάτων

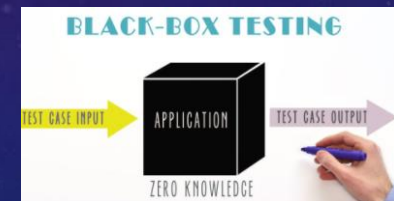
----->	0 <----->	10 <-----> 20	<----->
Μη έγκυρη βαθμολογία	Ανεπιτυχής εξέταση	Επιτυχής εξέταση	Μη έγκυρη βαθμολογία

Βήμα 2ο: Καθορισμός ακραίων τιμών διαστημάτων

-----> -1	0 <-----> 9	10 <-----> 20	21 <----->
Μη έγκυρη βαθμολογία	Ανεπιτυχής εξέταση	Επιτυχής εξέταση	Μη έγκυρη βαθμολογία

Βήμα 3ο: 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 <= βαθμός < 10
3	9	Ανεπιτυχής εξέταση	Άνω άκρο διαστήματος 0 <= βαθμός < 10
4	10	Επιτυχής εξέταση	Κάτω άκρο διαστήματος 10 <= βαθμός <= 20
5	20	Επιτυχής εξέταση	Άνω άκρο διαστήματος 10 <= βαθμός <= 20
6	21	Μη έγκυρη βαθμολογία	Κάτω άκρο διαστήματος βαθμός > 20



5.2.5 Μέθοδος ελέγχου «Μαύρο Κουτί»

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

Απάντηση

1. **Λάθος** – Στον έλεγχο «μαύρου κουτιού» τα σενάρια ελέγχου προκύπτουν από τις προδιαγραφές του προγράμματος, αγνοώντας εντελώς τον κώδικα. Δηλαδή, το πρόγραμμα μοιάζει σα να βρίσκεται μέσα σε ένα μαύρο κουτί που κρύβει το περιεχόμενό του.

Σωστή πρόταση: Στον έλεγχο «μαύρου κουτιού» τα σενάρια ελέγχου βασίζονται στις προδιαγραφές του προγράμματος.

2. **Σωστό.**

3. **Λάθος** – Καλή στρατηγική θεωρείται να γίνεται ο έλεγχος των ακραίων τιμών κάθε διαστήματος της εισόδου (boundary value analysis).

Σωστή πρόταση: Κατά τον έλεγχο ακραίων τιμών ελέγχονται οι ακραίες τιμές από κάθε διάστημα της εισόδου.

4. **Σωστό.**

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.

Σενάρια ελέγχου

A/A	Είσοδος	Αναμενόμενο αποτέλεσμα	Περίπτωση που ελέγχεται
1	-0,1	Μη έγκυρος Γ.Μ.Ο.	Άνω άκρο διαστήματος Γ.Μ.Ο. < 0
2	0	Παραπέμπεται σε επανεξέταση	Κάτω άκρο διαστήματος $0 \leq \text{Γ.Μ.Ο.} < 9,5$
3	9,4	Παραπέμπεται σε επανεξέταση	Άνω άκρο διαστήματος $0 \leq \text{Γ.Μ.Ο.} < 9,5$
4	9,5	Προάγεται	Κάτω άκρο διαστήματος $9,5 \leq \text{Γ.Μ.Ο.} \leq 20$
5	20	Προάγεται	Άνω άκρο διαστήματος $9,5 \leq \text{Γ.Μ.Ο.} \leq 20$
6	20,1	Μη έγκυρος Γ.Μ.Ο.	Κάτω άκρο διαστήματος Γ.Μ.Ο. > 20



Ε.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.

Σενάρια ελέγχου

A/A	Είσοδος	Αναμενόμενο αποτέλεσμα	Περίπτωση που ελέγχεται
1	0	-1	Άνω άκρο διαστήματος ημέρες < 1
2	1	50	Κάτω άκρο διαστήματος $1 \leq$ ημέρες \leq 3
3	3	150	Άνω άκρο διαστήματος $1 \leq$ ημέρες \leq 3
4	4	188	Κάτω άκρο διαστήματος $3 <$ ημέρες \leq 7
5	7	329	Άνω άκρο διαστήματος $3 <$ ημέρες \leq 7
6	8	360 336	Κάτω άκρο διαστήματος $7 <$ ημέρες \leq 20
7	20	900 840	Άνω άκρο διαστήματος $7 <$ ημέρες \leq 20
8	21	-1	Κάτω άκρο διαστήματος ημέρες > 20

Κατηγορίες λαθών	Περιπτώσεις λαθών	Παραδείγματα	Εντοπίζονται από	Διορθώνει	Δυνατή εκτέλεση
Συντακτικά (λάθη κατά την υλοποίηση)	<ul style="list-style-type: none"> * αδήλωτη μεταβλητή * αντικανονικό όνομα μεταβλητής/σταθεράς * μη ισορροπημένες παρενθέσεις/αγκύλες/εισαγωγικά * αναγραμματισμός εντολής * παράλειψη εντολής τερματισμού * λανθασμένη χρήση εντολής * ασυμβατότητα τύπων σε εντολή εκχώρησης/έκφραση * λάθος εμφώλευση δομών * λανθασμένη κλήση υποπρογράμματος 	<pre>2a <-- 5 x <-- 'hello μο <-- (α+β/2 ΤέοςΑν Διάβασε 8 x <-- 'hello'/2</pre>	Μεταγλωπιστής (αυτόματα) ή Προγραμματιστής	Προγραμματιστής	ΟΧΙ
Κατά την εκτέλεση (οδηγούν σε αντικανονικό τερματισμό)	<ul style="list-style-type: none"> * αναφορά σε απροσδιόριστη μεταβλητή/στοιχείο πίνακα * διαίρεση με το μηδέν * αρνητικό υπόριζο * παραβίαση ορίων πίνακα 	<pre>x <-- α / 0 κ <-- T_P(-2)</pre>	Άνθρωπος (Προγραμματιστής/ Χρήστης)	Προγραμματιστής	ΝΑΙ => αντικανονικός τερματισμός (crash)
Λογικά	<ul style="list-style-type: none"> * παράλειψη παρενθέσεων * λάθος χρήση αριθμητικού/συγκριτικού/λογικού τελεστή * λάθος αρχικοποίηση μεταβλητής * λάθος παράμετροι κλήσης υποπρογράμματος * λάθος σειρά εντολών * ατέρμων βρόχος 	<pre>μο <-- x+y+z/3 άρπιος <-- x div 2 = 0 γινόμενο <-- 0 Όσο χ=χ επανάλαβε</pre>	Άνθρωπος (Προγραμματιστής/ Χρήστης)	Προγραμματιστής	ΝΑΙ => λάθος αποτελέσματα

Getting Ideas to Solve a Bug

While Coding



While Eating



While Relaxing



While In Bathroom



1.

$x \leftarrow 10$

ΟΣΟ $x > 0$ ΕΠΑΝΑΛΑΒΕ

ΓΡΑΨΕ x

$x \leftarrow x - 3$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

$y \leftarrow T_P(x)$

Λάθος κατά την εκτέλεση: αρνητικό υπόριζο (-2)

3. Εισαγωγή φύλου Α/Γ με εγκυρότητα

Αρχή_επανάληψης

Διάβασε ϕ

Μέχρις_ότου $\phi = 'Α'$ ΚΑΙ $\phi = 'Γ'$

Λογικό λάθος : ατέρμων βρόχος

5.

για x από -10 μέχρι 10

Γράψε $1/x$

Τέλος_επανάληψης

Λάθος κατά την εκτέλεση: διαίρεση με το μηδέν

Περιγράψτε το λάθος και κατηγοριοποιήστε το ως:
συντακτικό(1),λογικό(2),κατά την εκτέλεση(3)

2. Υπολογισμός του γινομένου: $1 * 2 * 3 * \dots * 100$.

Για i από 1 μέχρι 100

$g \leftarrow 1$

$g \leftarrow g * i$

Τέλος_επανάληψης

Γράψε g

Λογικό λάθος: αρχικοποιείται σε κάθε επανάληψη

4.

Διάβασε x

Αν $x > 0$ τότε

$y \leftarrow x / 2$

Αλλιώς

$x \leftarrow x + 1$

Τέλος_αν

Γράψε y

Λάθος κατά την εκτέλεση: y απροσδιόριστο αν $x \leq 0$



Περιγράψτε το λάθος και κατηγοριοποιήστε το ως: συντακτικό(1),λογικό(2),κατά την εκτέλεση(3)

6. Μέσος όρος 100 εισαγόμενων βαθμών

Για i από 1 μέχρι 100

Διάβασε β

Τέλος_επανάληψης

$\Sigma \leftarrow 0$

Για i από 1 μέχρι 100

$\Sigma \leftarrow \Sigma + \beta$

Τέλος_επανάληψης

Γράψε $\Sigma / 100$

Λογικό λάθος: προστίθεται μόνο ο τελευταίος βαθμός



8.

Εισαγωγή 10 τιμών και έλεγχος αν ήταν όλες άρτιες

άρτιες \leftarrow Αληθής

Για i από 1 μέχρι 10

Διάβασε x

Αν $x \bmod 2 = 0$ τότε

άρτιες \leftarrow Αληθής

Αλλιώς

άρτιες \leftarrow Ψευδής

Τέλος_αν

Τέλος_επανάληψης

Αν άρτιες = Αληθής τότε

Γράψε 'Όλες άρτιες'

Αλλιώς

Γράψε 'Όχι όλες άρτιες'

Τέλος_αν

Λογικό λάθος: το αποτέλεσμα θα κριθεί από την τελευταία τιμή

7.

Σε πίνακα $A[100]$ ταξινομημένο με φθίνουσα σειρά υπολογισμός του πλήθους των στοιχείων που ισούνται με τη μέγιστη τιμή

$i \leftarrow 1$

$\pi \leftarrow 0$

ΟΣΟ $A[i] = A[1]$ ΕΠΑΝΑΛΑΒΕ

$i \leftarrow i + 1$

$\pi \leftarrow \pi + 1$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ π

Λάθος κατά την εκτέλεση: αν όλα τα στοιχεία είναι ίσα, αναφορά στο στοιχείο $A[101]$

Διόρθωση:

$i \leftarrow 1$

$\pi \leftarrow 0$

$f \leftarrow$ Ψευδής

ΟΣΟ $i \leq 100$ ΚΑΙ $f =$ Ψευδής ΕΠΑΝΑΛΑΒΕ

Αν $A[i] = A[1]$ τότε

$i \leftarrow i + 1$

$\pi \leftarrow \pi + 1$

Αλλιώς

$f \leftarrow$ Αληθής

Τέλος_αν

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ π

Περιγράψτε το λάθος και κατηγοριοποιήστε το ως: συντακτικό(1),λογικό(2),κατά την εκτέλεση(3)

9. Δίνεται ο πραγματικός A[100] με τις πωλήσεις (€) ενός πωλητή που είναι το πολύ 100 ή τερματίζονται με το -1. Ποιος ο μέσος όρος των πωλήσεων;

```
s <-- 0
```

```
p <-- 0
```

```
i <-- 1
```

Όσο A[i] <> -1 ΚΑΙ i <= 100 επανάλαβε

```
s <-- s + A[i]
```

```
p <-- p + 1
```

```
i <-- i + 1
```

ΤέλοςΕπανάληψης

Αν p <> 0 τότε

Γράψε s/p

Αλλιώς

Γράψε 'καμία'

ΤέλοςΑν

Λάθος κατά την εκτέλεση στην Όσο εάν όλα τα στοιχεία του A είναι διαφορετικά του -1 και φτάσει να ελέγξει το A[101]. **Δίορθωση:**

```
s <-- 0
```

```
p <-- 0
```

```
i <-- 1
```

```
f <-- Ψευδής
```

Όσο i <= 100 ΚΑΙ f = Ψευδής επανάλαβε

Αν A[i] <> -1 τότε

```
s <-- s + A[i]
```

```
p <-- p + 1
```

```
i <-- i + 1
```

Αλλιώς

```
f <-- Αληθής
```

ΤέλοςΑν

ΤέλοςΕπανάληψης

Αν p <> 0 τότε

Γράψε s/p

Αλλιώς

Γράψε 'καμία'

ΤέλοςΑν

Why programmer always prefer dark mode ?

- because light attracts bugs



Αν οι εισοδοι είναι περισσότερες, τότε κάθε μία πρέπει να ελεγχθεί ανεξάρτητα από τις άλλες. Για να γίνει αυτό, δημιουργούνται σενάρια ελέγχου όπου μία από τις εισόδους λαμβάνει όλες τις ακραίες τιμές ενώ οι υπόλοιπες διατηρούνται σταθερές δίνοντας μια οποιαδήποτε έγκυρη τιμή. Αυτό επαναλαμβάνεται για όλες τις εισόδους. (5.2.5 Μέθοδος ελέγχου «Μαύρο Κουτί»).

π.χ. επιλογή σε σώμα στρατού όσων έχουν

- βάρος στο ακέραιο [70, 90] και
- ύψος στο [1.80, 2] (με ακρίβεια 2 δεκαδικών ψηφίων)

Σενάρια ελέγχου			
Είσοδος			
A/A	βάρος	ύψος	Αναμενόμενο αποτέλεσμα
1	69	1.90	απορρίπτεται
2	70	1.90	δεκτός
3	90	1.90	δεκτός
4	91	1.90	απορρίπτεται
5	80	1.70	απορρίπτεται
6	80	1.80	δεκτός
7	80	2.00	δεκτός
8	80	2.01	απορρίπτεται

80: σταθερή έγκυρη τιμή βάρους

1.90: σταθερή έγκυρη τιμή ύψους

69,70,90,91: ακραίες τιμές βάρους

1.70,1.80,2.00,2.01: ακραίες τιμές ύψους