Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

ΚΕΦΑΛΑΙΟ 13 - (Ενότητα 5) - Εκσφαλμάτωση Προγράμματος

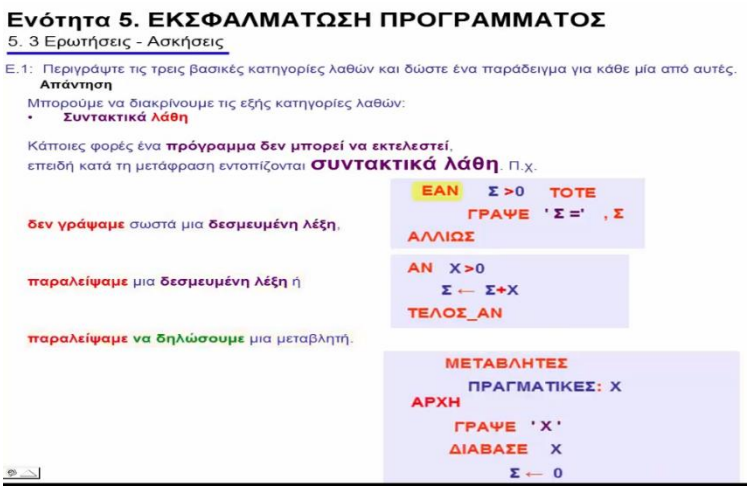
1. Ποιες είναι οι βασικές κατηγορίες λαθών που είναι δυνατό να παρουσιαστούν στα προγράμματα;

- Λάθη κατά την υλοποίηση (Συντακτικά λάθη)
- Λάθη κατά την εκτέλεση (Λάθη που οδηγούν σε αντικανονικό τερματισμό του προγράμματος)
- Λογικά λάθη (Λογικά λάθη που παράγουν λανθασμένα αποτελέσματα)

2. Που οφείλονται συνήθως τα λάθη κατά την υλοποίηση και πότε γίνονται αντιληπτά;

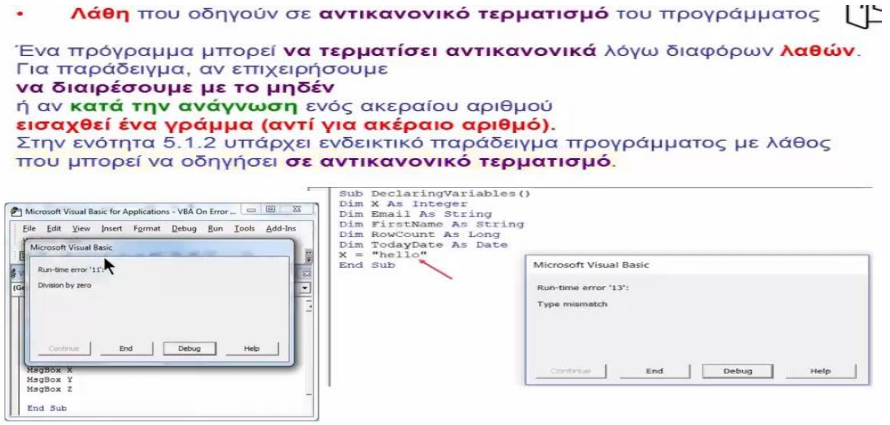
Τα λάθη κατά το χρόνο υλοποίησης προκαλούνται κυρίως από **λανθασμένη** σύνταξη εντολών προγράμματος. Τέτοια λάθη μπορεί να είναι η λανθασμένη συγγραφή μιας δεσμευμένης λέξης της γλώσσας προγραμματισμού ή η χρήση μιας δομής ελέγχου χωρίς την εντολή τερματισμού της.

Ανιχνεύονται από τον μεταγλωττιστή, ο οποίος εμφανίζει προς τον προγραμματιστή κάποιο προειδοποιητικό μήνυμα και δεν επιτρέπεται η εκτέλεσή του προγράμματος μέχρι να το διορθώσει ο προγραμματιστής.



3. Που οφείλονται συνήθως τα λάθη κατά την εκτέλεση και πότε γίνονται αντιληπτά;

Τα λάθη που προκαλούν τον **αντικανονικό** τερματισμό της εφαρμογής και το κρέμασμα του συστήματος εμφανίζονται σε πραγματικό περιβάλλον εκτέλεσης. Τέτοια λάθη είναι δυνατό να προκληθούν από την προσπάθεια διαίρεσης ενός αριθμού με το μηδέν, την κλήση μιας διαδικασίας με δεδομένα που δεν μπορεί να χειριστεί, όπως η αναζήτηση διαγραμμένων αρχείων, η υπερχείλιση μιας αριθμητικής μεταβλητής ή από δυσλειτουργία του υλικού μέρους του υπολογιστή, όπως η καταστροφή του σκληρού δίσκου του συστήματος, ο τερματισμός μιας σύνδεσης δικτύου και η αποσύνδεση του εκτυπωτή.



4. Που οφείλονται συνήθως τα λογικά λάθη και πότε γίνονται αντιληπτά;

Τα λογικά λάθη είναι συνήθως **λάθη σχεδιασμού** και δεν προκαλούν τη διακοπή της εκτέλεσης του προγράμματος. Ενώ ο μεταγλωττιστής της γλώσσας προγραμματισμού δεν ανιχνεύει κανένα συντακτικό λάθος και κατά την εκτέλεση του προγράμματος δεν παρουσιάζονται ανεπιθύμητες καταστάσεις σφαλμάτων, τελικά δεν παράγονται τα επιθυμητά αποτελέσματα. Η ανίχνευση τέτοιων λαθών δεν είναι δυνατό να πραγματοποιηθεί από κάποιο εργαλείο του υπολογιστή και διαπιστώνονται μόνο με τη διαδικασία ελέγχου (testing) και την ανάλυση των αποτελεσμάτων των προγραμμάτων.

5. Τι ονομάζεται εκσφαλμάτωση και ποιος ο στόχος της;

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.

6. Ποιος ο ρόλος των σχολίων στην εκσφαλμάτωση;

Η εισαγωγή γραμμών με σχόλια σε ένα πρόγραμμα υποβοηθά σημαντικά την εκσφαλμάτωση. Χρησιμοποιούνται προκειμένου να διαχωρίζονται οι επεξηγηματικές φράσεις από τις λέξεις-κλειδιά του αλγορίθμου.

5. 3 Ερωτήσεις - Ασκήσεις

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.

Απάντηση	Σ	Λ
1. Στον έλεγχο «μαύρου κουτιού» τα σενάρια ελέγχου βασίζονται στις προδιαγραφές του προγράμματος. 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.	☐	☑
2. Τα σενάρια ελέγχου περιλαμβάνουν και μη έγκυρες τιμές εισόδου.	☑	☐
3. Κατά τον έλεγχο ακραίων τιμών ελέγχονται οι ακραίες τυχαίες τιμές από κάθε διάστημα της εισόδου. Καλή στρατηγική θεωρείται να γίνεται ο έλεγχος των ακραίων τιμών κάθε διαστήματος της εισόδου (boundary value analysis)	☐	☑
4. Ο έλεγχος «μαύρου κουτιού» μπορεί να εφαρμοστεί και σε υποπρογράμματα.	☑	☐

7. Εκσφαλμάτωση λογικών λαθών σε δομή επιλογής

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

- τη συνθήκη ή τις συνθήκες
- τις ομάδες εντολών που εκτελούνται όταν μια συνθήκη είναι αληθής ή ψευδής.

Στην ανίχνευση ενός λογικού λάθους στις δομές επιλογής δεν αρκεί η μεμονωμένη μελέτη των συνθηκών και των ομάδων εντολών που εκτελούνται όταν μια συνθήκη είναι αληθής ή ψευδής, αλλά χρειάζεται να μελετηθεί το αποτέλεσμα που παράγει ο συνδυασμός των συνθηκών και των ομάδων εντολών.

Βήμα 1ο: Δημιουργία **ισοδύναμων** διαστημάτων
Σύμφωνα με την εκφώνηση υπάρχουν τα ακόλουθα **έγκυρα** διαστήματα τιμών **εισόδου**:
- **0 ≤ Γ.Μ.Ο. < 9,5**
- **9,5 ≤ Γ.Μ.Ο. ≤ 20**

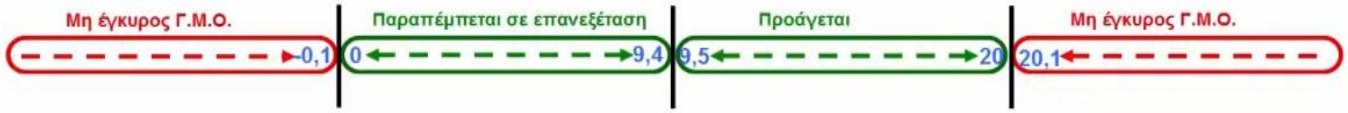
Επίσης υπάρχουν τα ακόλουθα μη έγκυρα διαστήματα τιμών εισόδου:
- **Γ.Μ.Ο. < 0**
- **Γ.Μ.Ο. > 20**

Ε.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.

Βήμα 2ο: Καθορισμός **ακραίων** τιμών διαστημάτων
Για **να υπολογίσουμε τα άκρα που λείπουν** από τα διαστήματα των τιμών **εισόδου**,
θα προσθέσουμε ή **θα αφαιρέσουμε 0, 1**
από το άκρο του **προηγούμενου** ή **επόμενου διαστήματος** αντίστοιχα, αφού **η εκφώνηση** απαιτεί «**ο έλεγχος** του Γ.Μ.Ο. να γίνει **με ακρίβεια ενός δεκαδικού ψηφίου**». Καταλήγουμε έτσι στο ακόλουθο διάγραμμα.

Βήμα 3ο: Δημιουργία **σεναρίων ελέγχου**
Χρησιμοποιώντας το παραπάνω διάγραμμα **δημιουργούμε ένα σενάριο ελέγχου για κάθε ακραία τιμή** εισόδου.

Α/Α	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 ≤ Γ.Μ.Ο. < 9,5
3	9,4	Παραπέμπεται σε επανεξέταση	Άνω άκρο διαστήματος 0 ≤ Γ.Μ.Ο. < 9,5
4	9,5	Προάγεται	Κάτω άκρο διαστήματος 9,5 ≤ Γ.Μ.Ο. ≤ 20
5	20	Προάγεται	Άνω άκρο διαστήματος 9,5 ≤ Γ.Μ.Ο. ≤ 20
6	20,1	Μη έγκυρος Γ.Μ.Ο	Κάτω άκρο διαστήματος Γ.Μ.Ο.> 20



8. Εκσφαλμάτωση λογικών λαθών σε δομή 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.

Στους υπολογισμούς που γίνονται εντός των δομών επανάληψης χρειάζεται να δοθεί ιδιαίτερη προσοχή στο αν θα συμπεριλάβουμε στον υπολογισμό την τιμή που λαμβάνει κάποια μεταβλητή στην τελευταία επανάληψη.

Συμβουλή: Κατά την εκσφαλμάτωση των δομών επανάληψης χρειάζεται να δίνετε 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
- στην αλληλουχία των εντολών του βρόχου και στη σειρά εκτέλεσής τους
- στο κριτήριο της περατότητας
- στην πρώτη επανάληψη και στην περίπτωση που ο βρόχος επανάληψης δεν πρέπει να εκτελεστεί ούτε μία φορά
- στην τελευταία επανάληψη

9. Εκσφαλμάτωση λογικών λαθών σε πίνακες

Κατά την εκσφαλμάτωση προγραμμάτων που χρησιμοποιούν πίνακες χρειάζεται να δίνετε ιδιαίτερη προσοχή:

- στο μέγεθος των πινάκων κατά τη δήλωσή τους,
- στους δείκτες των πινάκων κατά την προσπέλασή τους,
- στη μη υπέρβαση των ορίων του πίνακα.

10. Εκσφαλμάτωση λογικών λαθών στα υποπρογράμματα

Κατά την εκσφαλμάτωση προγραμμάτων που χρησιμοποιούν υποπρογράμματα χρειάζεται να δίνεται προσοχή στον εντοπισμό λογικών λαθών που σχετίζονται με:

- την κλήση του υποπρογράμματος και το πέρασμα των παραμέτρων
- τα λοιπά λογικά λάθη που εμφανίζονται και στα προγράμματα.

***** 5.2.5 Μέθοδος ελέγχου «Μαύρο Κουτί»

11. Ποιος ο ρόλος των σεναρίων ελέγχου (test case);

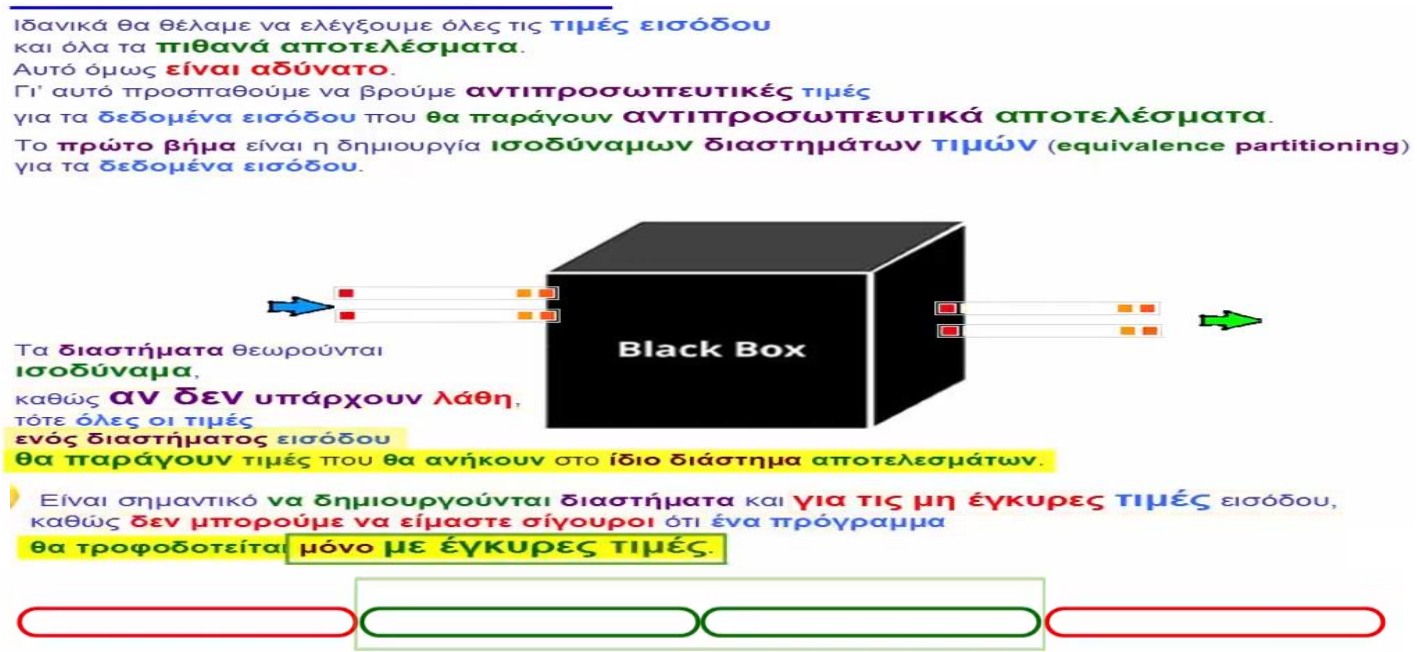
Ένα σενάριο ελέγχου (test case) περιγράφει τα δεδομένα εισόδου ολόκληρου του προγράμματος ή τμήματος του προγράμματος (διαδικασία, συνάρτηση) και τα αναμενόμενα αποτελέσματα. Τα σενάρια ελέγχου εκτελούνται, είτε σε πραγματικό περιβάλλον προγραμματισμού είτε εικονικά με δημιουργία πίνακα τιμών των μεταβλητών.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

12. Πώς εφαρμόζεται η τεχνική ελέγχου μαύρου κουτιού;

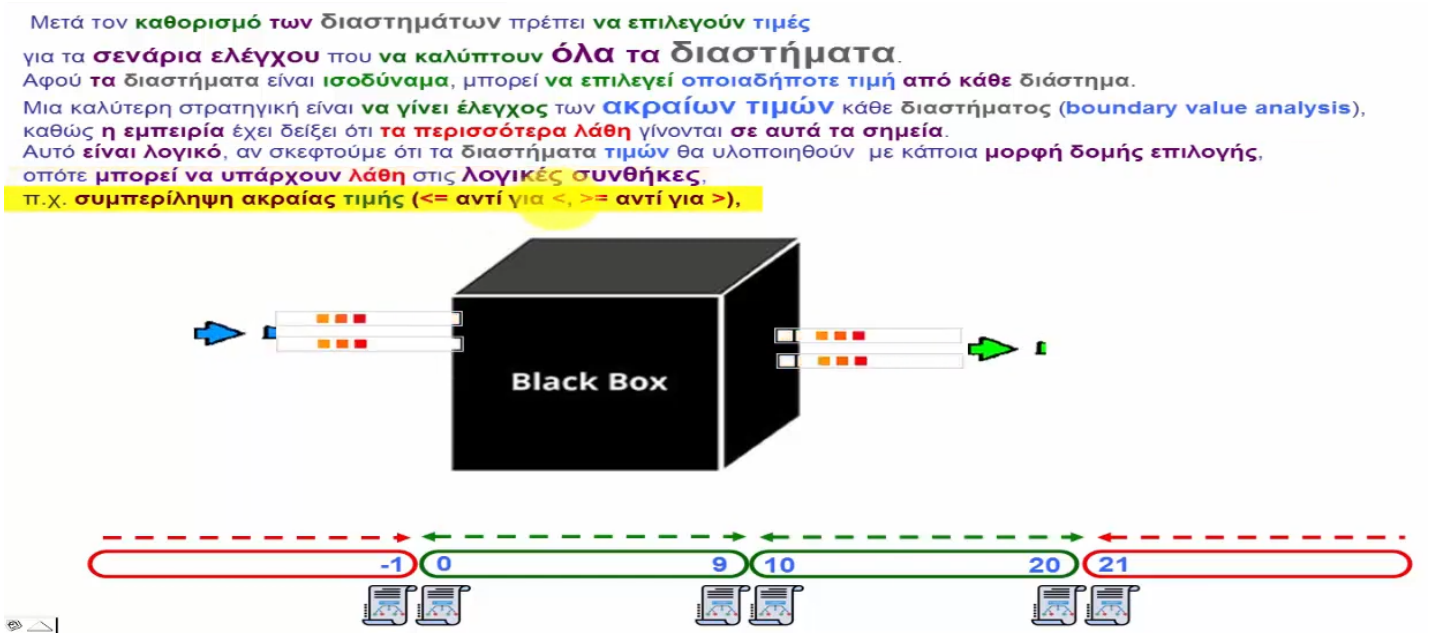
Μια δημοφιλής τεχνική ελέγχου είναι ο έλεγχος μαύρου κουτιού (black-box testing).Ονομάζεται έτσι επειδή τα δεδομένα εισόδου στα σενάρια ελέγχου προκύπτουν από τις προδιαγραφές του προγράμματος, αγνοώντας εντελώς τον κώδικα. Δηλαδή το πρόγραμμα μοιάζει σαν να βρίσκεται μέσα σε ένα μαύρο κουτί που κρύβει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

13. Ποιος ο ρόλος των αντιπροσωπευτικές τιμές στις τιμές εισόδου;

Ιδανικά θα θέλαμε να ελέγξουμε όλες τις τιμές εισόδου και όλα τα πιθανά αποτελέσματα. Αυτό όμως είναι αδύνατο. Γι' αυτό προσπαθούμε να βρούμε αντιπροσωπευτικές τιμές για τα δεδομένα εισόδου που θα παράγουν αντιπροσωπευτικά αποτελέσματα. Το πρώτο βήμα είναι η δημιουργία ισοδύναμων διαστημάτων τιμών (equivalence partitioning) για τα δεδομένα εισόδου. Τα διαστήματα θεωρούνται ισοδύναμα, καθώς αν δεν υπάρχουν λάθη, τότε όλες οι τιμές ενός διαστήματος εισόδου θα παράγουν τιμές που θα ανήκουν στο ίδιο διάστημα αποτελεσμάτων.



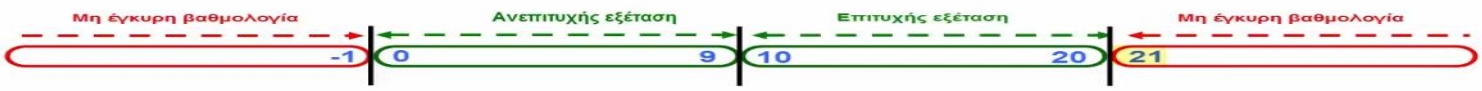
Μετά τον καθορισμό των διαστημάτων πρέπει να επιλεγούν τιμές για τα σενάρια ελέγχου που να καλύπτουν όλα τα διαστήματα. Αφού τα διαστήματα είναι ισοδύναμα, μπορεί να επιλεγεί οποιαδήποτε τιμή από κάθε διάστημα. Μια καλύτερη στρατηγική είναι να γίνει έλεγχος των ακραίων τιμών κάθε διαστήματος (boundary value analysis), καθώς η εμπειρία έχει δείξει ότι τα περισσότερα λάθη γίνονται σε αυτά τα σημεία. Αυτό είναι λογικό, αν σκεφτούμε ότι τα διαστήματα τιμών θα υλοποιηθούν με κάποια μορφή δομής επιλογής, οπότε μπορεί να υπάρχουν λάθη στις λογικές συνθήκες, π.χ. συμπερίληψη ακραίας τιμής (<= αντί για <, >= αντί για >), παράλειψη ακραίας τιμής (< αντί για <=, > αντί για >=).



Δημιουργία σεναρίων ελέγχου

Το τελευταίο βήμα
είναι **να δημιουργήσουμε**
ένα **σενάριο ελέγχου**
για κάθε ακραία τιμή.
Κάθε σενάριο πρέπει **κατ' ελάχιστο**
να περιλαμβάνει
την **τιμή εισόδου**,
το **αναμενόμενο αποτέλεσμα**
(σύμφωνα με την εκφώνηση του προβλήματος) και
περιγραφή της περίπτωσης που **ελέγχεται**.

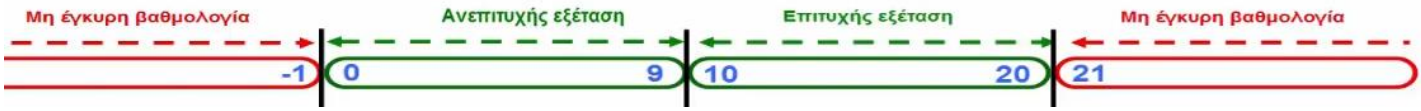
Α/Α	Είσοδος	Αναμενόμενο αποτέλεσμα	Περίπτωση που ελέγχεται
1	-1	Μη έγκυρη βαθμολογία	Άνω άκρο διαστήματος βαθμός < 0
2	0	Ανεπιτυχής εξέταση	Κάτω άκρο διαστήματος 0 <= βαθμός < 10
3	9	Ανεπιτυχής εξέταση	Άνω άκρο διαστήματος 0 <= βαθμός < 10
4	10	Επιτυχής εξέταση	Κάτω άκρο διαστήματος 10 <= βαθμός <= 20
5	20	Επιτυχής εξέταση	Άνω άκρο διαστήματος 10 <= βαθμός <= 20
6	21	Μη έγκυρη βαθμολογία	Κάτω άκρο διαστήματος βαθμός > 20



Βλέπουμε λοιπόν, ότι ακόμα
και ένα πολύ **απλό πρόγραμμα**
με **μία μόνο είσοδο**,
η οποία δεν υφίσταται καμία επεξεργασία,
απαιτεί έξι σενάρια ελέγχου.

```
ΠΡΟΓΡΑΜΜΑ    Βαθμολογία
  ΜΕΤΑΒΛΗΤΕΣ
     ΑΚΕΡΑΙΕΣ:      βαθμολογία
ΑΡΧΗ
          ΓΡΑΨΕ     ' Δώσε την βαθμολογία:  '
          ΔΙΑΒΑΣΕ   βαθμολογία

    ΑΝ  βαθμολογία >= 10  ΚΑΙ  βαθμολογία <= 20  ΤΟΤΕ
          ΓΡΑΨΕ ' Επιτυχής εξέταση  '
    ΑΛΛΙΩΣ_ ΑΝ βαθμολογία >= 0  ΚΑΙ βαθμολογία < 10  ΤΟΤΕ
          ΓΡΑΨΕ ' Ανεπιτυχής εξέταση  '
    ΑΛΛΙΩΣ
          ΓΡΑΨΕ ' Μη έγκυρη βαθμολογία '
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```



14. Πώς γίνεται ο έλεγχος στα υποπρογράμματα;

Η τεχνική που περιγράφηκε ανωτέρω, μπορεί να εφαρμοστεί και σε υποπρογράμματα. Αυτό είναι συνηθισμένο σε μεγάλα προγράμματα που αποτελούνται από χιλιάδες γραμμές κώδικα τα οποία είναι οργανωμένα σε υποπρογράμματα. Σε αυτές τις περιπτώσεις πρώτα ελέγχεται κάθε υποπρόγραμμα μεμονωμένα. Αφού διαπιστωθεί η ορθή λειτουργία του καθενός, μόνο τότε πραγματοποιείται έλεγχος ολόκληρου του προγράμματος.

15. Πώς γίνεται ο έλεγχος αν έχουμε πολλές τιμές εισόδου;

Αν οι είσοδοι 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. Γίνεται κατανοητό ότι η αύξηση των δεδομένων εισόδου οδηγεί σε μεγάλη αύξηση των σεναρίων ελέγχου, ενώ η διαδικασία αρχίζει να γίνεται περίπλοκη.