Εξασφάλιση Ποιότητας και Πρότυπα

4η Εργασία

Ερώτημα 1:

Επεξήγηση προγράμματος:

Το πρόγραμμα ζητά ως είσοδο έναν ακέραιο αριθμό. Βάζουμε από το πληκτρολόγιο έναν αριθμό. Όσο ο αριθμός αυτός είναι αρνητικός ακέραιος, τότε πρέπει να ξαναδώσουμε τιμή. Αν βάλουμε τον αριθμό μηδέν, πρέπει πάλι να ξαναδώσουμε νέα τιμή. Αν βάλουμε μη ακέραιο θετικό ή αρνητικό αριθμό, τότε το πρόγραμμα μας εγκλωβίζεται σε loop και εκτυπώνεται συνέχεια η πρόταση “Give an integer please:”. Τέλος, αν βάλουμε θετικό ακέραιο αριθμό, το πρόγραμμα συνεχίζεται κανονικά. Αν βάλουμε αριθμό κάτω από 40, τότε το αποτέλεσμα που θα πάρουμε είναι ο αριθμός που βάλαμε +11. Δηλαδή, αν βάλουμε τον αριθμό 5, το αποτέλεσμα που θα πάρουμε είναι το 16. Αλλιώς αν βάλουμε αριθμό πάνω ή ίσο με 40 τότε το αποτέλεσμα που θα πάρουμε είναι : ο αριθμός μας * 2 + 20 .Δηλαδή, αν βάλουμε τον αριθμό 333 , το αποτέλεσμα θα είναι 333*2 +20 = 686.

Παραδείγματα εκτελέσεων φαίνονται και παρακάτω στον πίνακα.

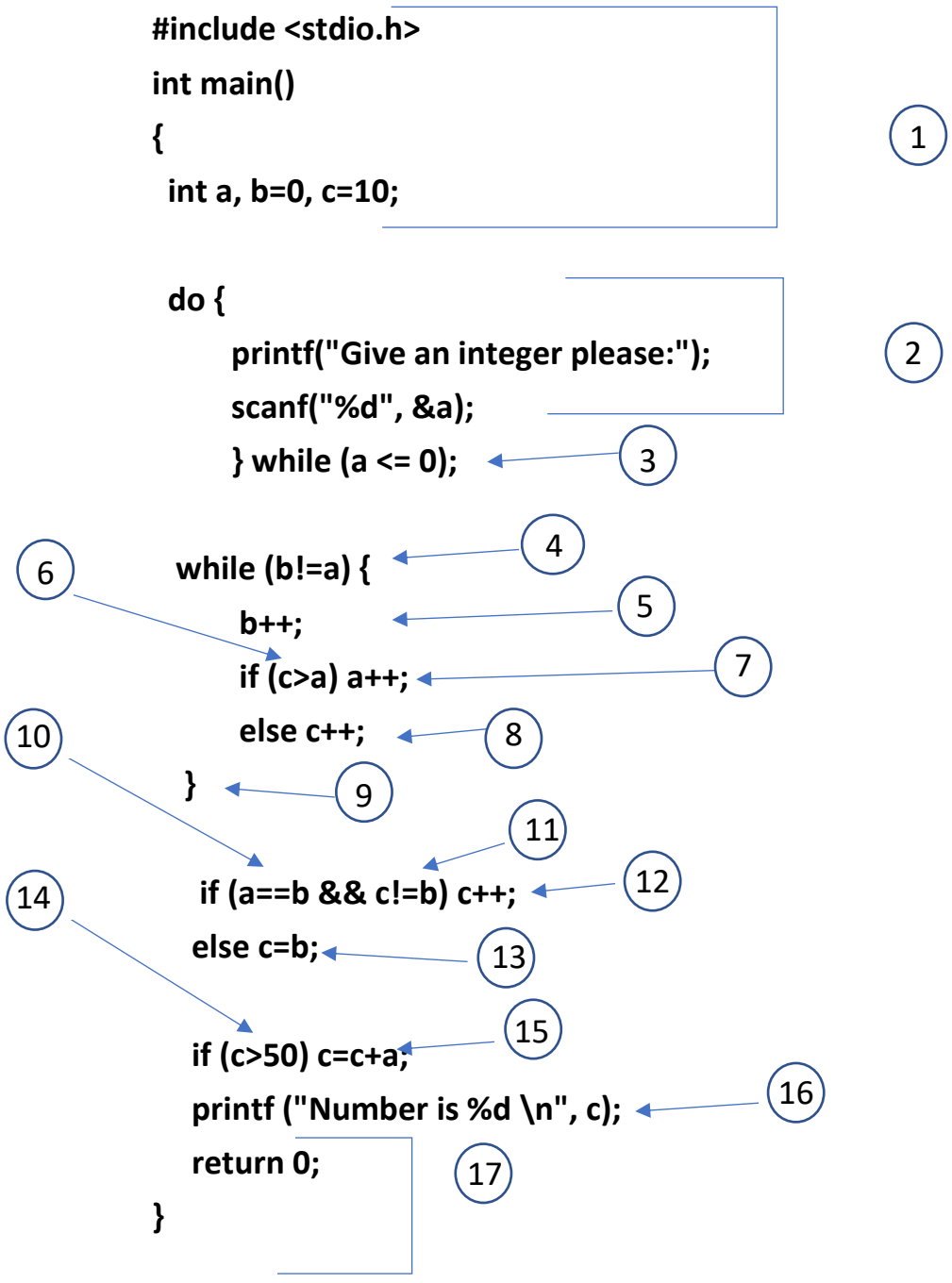
Ερώτημα 2:

Σύνολο τυχαίων τιμών εισόδου και αποτελέσματα

Τιμή	Αποτέλεσμα
0	**Give an integer please:**
1	**Number is 12**
-1	**Give an integer please:Give an integer please:Give an integer please:Give an integer please:Give an integer please:Give an integer please: …**
0.5	**Give an integer please:Give an integer please:Give an integer please:Give an integer please:Give an integer please:Give an integer please: …**
-0.5	**Give an integer please:Give an integer please:Give an integer please:Give an integer please:Give an integer please:Give an integer please: …**
5	**Number is 16**
39	**Number is 50**
40	**Number is 100**
333	**Number is 686**
5055	**Number is 10130**

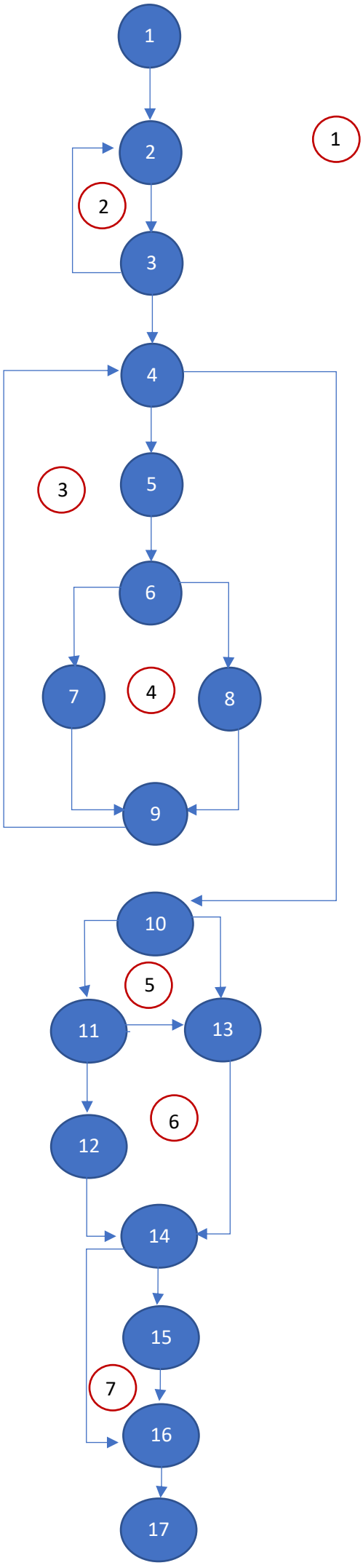
Ερώτημα 3:

```
#include <stdio.h>
int main()
{
 int a, b=0, c=10;

 do {
      printf("Give an integer please:");
      scanf("%d", &a);
      } while (a <= 0);

  while (b!=a) {
       b++;
       if (c>a) a++;
       else c++;
  }
   if (a==b && c!=b) c++;
   else c=b;

   if (c>50) c=c+a;
   printf ("Number is %d \n", c);
   return 0;
}
```



Παραδοχή: Θεωρώ ως 17 το τέλος του προγράμματος.

Με βάση την παραπάνω αρίθμηση ο γράφος ροής του προγράμματος είναι ο εξής:



Εύρεση κυκλωματικής πολυπλοκότητας:

Από τον παραπάνω γράφο προκύπτει ότι η κυκλωματική πολυπλοκότητα είναι **7**, γιατί:

1ος τρόπος: V(g) = e - n +2p = 22 – 17 + 2 = 7,

Όπου e = 22 οι ακμές του γραφήματος, n = 17 οι κόμβοι του γραφήματος και p = 1 ο αριθμός των συνεκτικών συνιστωσών του γράφου

2ος τρόπος: V(g) = p + 1 =7 ,

Όπου p = 6 οι απλές συνθήκες του κώδικα

Πιο συγκεκριμένα:

1. do-while(a<=0)
2. while (b!=a)
3. if (c>a)
4. if (a==b
5. && c!=b)
6. if (c>50)

3ος τρόπος: V(g) = οι περιοχές του γράφου =7

(απεικονίζονται πάνω στο γράφημα με κόκκινους κύκλους)

Ερώτημα 4:

Για τον εντοπισμό των βασικών μονοπατιών ακολουθούμε την εξής προσέγγιση:

• Πάρε το μικρότερο δυνατό μονοπάτι (με τις λιγότερες ακμές) το οποίο να είναι έγκυρο.

• Εμπλούτισε αυτό το μονοπάτι με όσο το δυνατόν λιγότερες νέες ακμές, ξεκινώντας από τον πρώτο κόμβο που έχεις αυτή τη δυνατότητα διακλάδωσης. Έλεγχος αν αυτό το μονοπάτι είναι έγκυρο, αλλιώς επανέλαβε αυτό το βήμα.

• Συνέχισε μέχρι να μην υπάρχουν νέες ακμές.

Ας δούμε πρώτα όλες τις εξαρτήσεις συνύπαρξης που υπάρχουν στον γράφο του λογισμικού μας:

E1. Δεν γίνεται σε ένα μονοπάτι να πάμε από τον κόμβο 10 στον κόμβο 13, καθώς ισχύει πάντα η πρώτη συνθήκη του if. Ακόμα, δεν γίνεται από τον κόμβο 11 να πάμε στον κόμβο 13, καθώς ισχύει πάντα και η δεύτερη συνθήκη του if. Συμπερασματικά, δεν χρησιμοποιείται ποτέ το else στον κόμβο 13.

E2. Η συνθήκη while(b!=a), δεν είναι δυνατό να εκτελεστεί μόνο μια φορά. Ο λιγότερος αριθμός φορών που μπορεί να εκτελεστεί είναι : 10. Οπότε δεν μπορούμε να πάμε κατευθείαν από τον κόμβο 4 στον κόμβο 10. Πρέπει να έχουμε μπει στο while τουλάχιστον 10 φορές πρώτα.

E3. Για το E2, δεν μπορεί να εκτελεστεί μόνο το μονοπάτι 4 – 5 – 6 – 7 – 9 στις 10 ή παραπάνω επαναλήψεις. Θα πρέπει να εκτελείται και το μονοπάτι 4 – 5 – 6 – 8 – 9.

Πρόκειται για ένα if-else, που κατά την διάρκεια των εκτελέσεων του while, μερικές φορές μπαίνει στο if και άλλες στο else.

Με βάση τις παραπάνω εξαρτήσεις το μικρότερο έγκυρο μονοπάτι είναι το εξής:

M1: 1 – 2 – 3 – (4 – 5 – 6 – 7 – 9) ^{9} – 4 – 5 – 6 – 8 – 9 – 4 – 10 – 11 – 12 – 14 – 16 – 17

Παραδοχή: Λόγω των εξαρτήσεων E2 και E3, προκύπτει το παραπάνω μονοπάτι. Ο ελάχιστος αριθμός επαναλήψεων του while είναι 10, οπότε το μέρος του μονοπατιού που αντιστοιχεί στην if του while , το υψώνω εις την ενάτη, για να έχει καλύτερη εμφάνιση και να μην χρειαστεί να επαναληφθεί το τμήμα αυτό του μονοπατιού 9 φορές. Ακόμα, επειδή την τελευταία φορά που μπαίνει στο while, θα περάσει στο else. Οπότε το έχω διαχωρίσει από τις υπόλοιπες επαναλήψεις.

Στη συνέχεια, ακολουθώντας τον αλγόριθμο έχουμε (με περίγραμμα εμφανίζονται η νέα ή οι νέες ακμές που προστίθενται σε σχέση με τα προηγούμενα βασικά μονοπάτια):

M2: 1 – 2 – 3 – 2 – 3 –(4 – 5 – 6 – 7 || 8 – 9)^{x} – 4 – 10 – 11 – 12 – 14 – 16 – 17

M3: 1 – 2 – 3 – (4 – 5 – 6 – 7 || 8– 9) ^{y} – 4 – 10 – 11 – 12 – 14 – 15 – 16 – 17

Όπου 11<=x< 49 και y >= 49

<u>Παραδοχή:</u> Στο μονοπάτι M1 είπαμε ότι μπαίνει στο else του while μόνο την τελευταία φορά. Στα υπόλοιπα μονοπάτια δεν ξέρουμε πόσες φορές θα μπει στο if και πόσες στο else, για αυτό χρησιμοποιώ για συντομία τον συμβολισμό «||», όπου σημαίνει 7 OR 8.

Σε αυτό το σημείο, οι μόνες ακμές που δεν συμπεριλαμβάνονται σε κανένα βασικό μονοπάτι είναι οι ακμές **10-13, 11-13, 13-14** . Τυπικά θα έπρεπε να δίναμε μονοπάτια τα οποία θα περιέχουν αυτές τις ακμές, αλλά πρακτικά θα είναι αδύνατο να ελεγχθούν, έτσι δεν χρειάζεται να το κάνουμε. Συνεπώς, το πρόγραμμά μπορεί να ελεγχθεί με **3** μονοπάτια, δηλαδή λιγότερα από την κυκλωματική πολυπλοκότητα η οποία αποτελεί άνω όριο των βασικών μονοπατιών.

Ερώτημα 5:

Κάποιες ενδεικτικές περιπτώσεις ελέγχου για τα μονοπάτια είναι:

Μονοπάτι	Περιγραφή	Περίπτωση Ελέγχου (input)	Αναμενόμενο Αποτέλεσμα (έξοδος προγράμματος)
M1	Δίνουμε ως είσοδο το 1	1	Number is 12
M2	Δίνουμε ως είσοδο πρώτα 0 και μετά 20	0 20	Give an integer please: Number is 31
M2	Δίνουμε ως είσοδο πρώτα –18 και μετά 5	-18 5	Give an integer please: Number is 16
M3	Δίνουμε ως είσοδο το 41	41	Number is 102
M3	Δίνουμε ως είσοδο το 156	156	Number is 332

Παραδοχή: Για το M1 δεν υπάρχει δεύτερη τιμή καθώς θεωρήσαμε ότι το ελάχιστο μονοπάτι μπαίνει 10 φορές στο while. Οποιαδήποτε τιμή εκτός του 1 και να βάλουμε οι επαναλήψεις θα βγαίνουν πάνω από 10.