

**ΕΞΑΣΦΑΛΙΣΗ ΠΟΙΟΤΗΤΑΣ ΚΑΙ ΠΡΟΤΥΠΑ**ΕΡΓΑΣΙΑ 4<sup>η</sup>**1 ΕΡΩΤΗΜΑ**

Το πρόγραμμα ζητά ως είσοδο έναν θετικό ακέραιο. Στην περίπτωση που ο ακέραιος είναι μικρότερος του 40, τυπώνεται ο ίδιος ακέραιος, προσαυξημένος κατά 11. Στην περίπτωση που ο ακέραιος είναι μεγαλύτερος ή ίσος του 40, τυπώνεται ο ακέραιος διπλασιασμένος και προσαυξημένος κατά 20.

$$\text{output} = \begin{cases} \text{ακέραιος} + 11, & 0 < \text{ακέραιος} < 40 \\ 2 \cdot \text{ακέραιος} + 20, & \text{ακέραιος} \geq 40 \end{cases}$$

**2 ΕΡΩΤΗΜΑ**

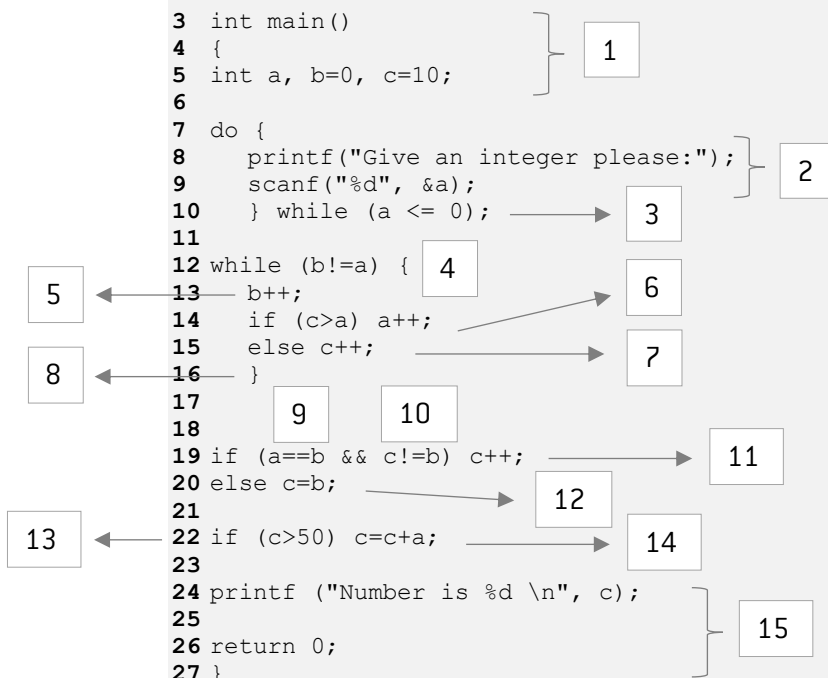
Είσοδος	Έξοδος
1	Number is 12
7	Number is 18
25	Number is 36
39	Number is 50
40	Number is 100
50	Number is 120
112	Number is 244
420	Number is 860
7523	Number is 15066
10000	Number is 20020

**3 ΕΡΩΤΗΜΑ**

```

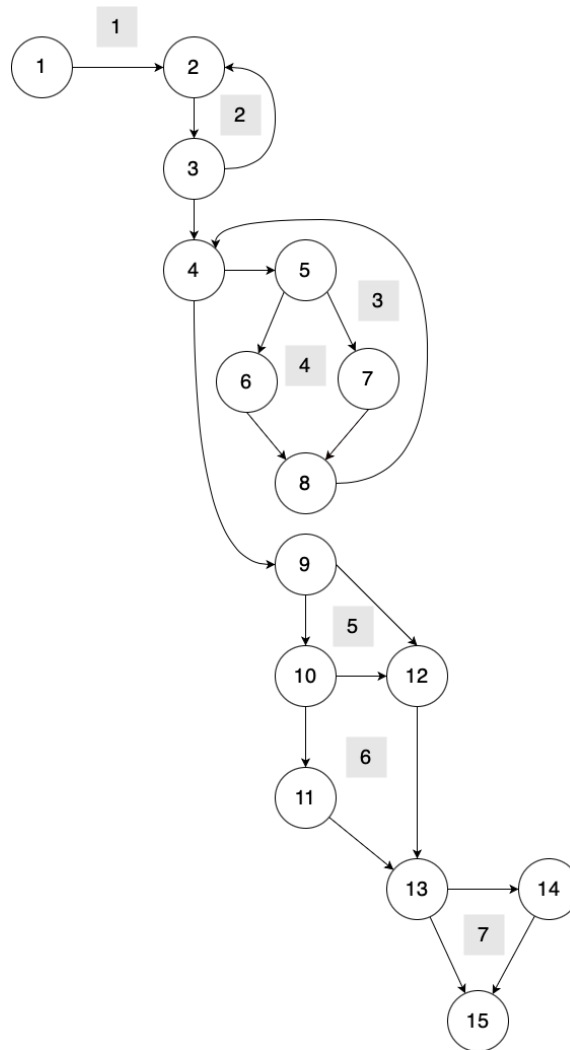
1 #include <stdio.h>
2
3 int main()
4 {
5     int a, b=0, c=10;
6
7     do {
8         printf("Give an integer please:");
9         scanf("%d", &a);
10    } while (a <= 0);
11
12    while (b!=a) {
13        b++;
14        if (c>a) a++;
15        else c++;
16    }
17
18    if (a==b && c!=b) c++;
19    else c=b;
20
21    if (c>50) c=c+a;
22
23    printf ("Number is %d \n", c);
24
25    return 0;
26 }
27

```



Κόμβος	Κώδικας
1	γρ. 3-5
2	γρ. 8-9
3	[a <= a]
4	γρ. 12
5	γρ. 13
6	γρ. 14
7	γρ. 15
8	γρ. 16
9	if {a == b}
10	if {c != b}
11	c++;
12	γρ. 20
13	if {c > 50}
14	c = c+a;
15	γρ. 24-27

Επομένως το γράφημα ροής είναι το εξής:



Οι κόμβοι 1-2 προσδιορίζουν τις σειριακές εντολές, οι 2-3 την `do-while` επανάληψη, οι 4-5-6-7-8 την `while` επανάληψη. Στην `while` ο κόμβος 4 προσδιορίζει την συνθήκη που επιτρέπει την επανάληψη, ο 5 προσδιορίζει την εντολή `b++` της `while`, ενώ οι 5-6-7-8 την `if` της `while`. Οι κόμβοι 9-10-11-12-13 προσδιορίζουν την σύνθετη `if`. Τέλος οι 13-14-15 προσδιορίζουν την `if` χωρίς την συνθήκη `else`, με τον 15 να προσδιορίζει το τελείωμα του κώδικα.

Όσον αφορά την κυκλωματική πολυπλοκότητα:

- Πρώτος τρόπος:

$$V(g) = e - n + 2p = 20 - 15 + 2 \cdot 1 = 7$$

- Δεύτερος τρόπος:

Οι εντολές του κώδικα είναι η `do-while` {1}, η `while` {1}, η `if` της `while` {1}, η σύνθετη `if` {2} και η `if` στο τέλος {1}. Συνολικά 6 εντολές.

$$V(g) = \text{εντολές} + 1 = 6 + 1 = 7$$

- Τρίτος τρόπος:

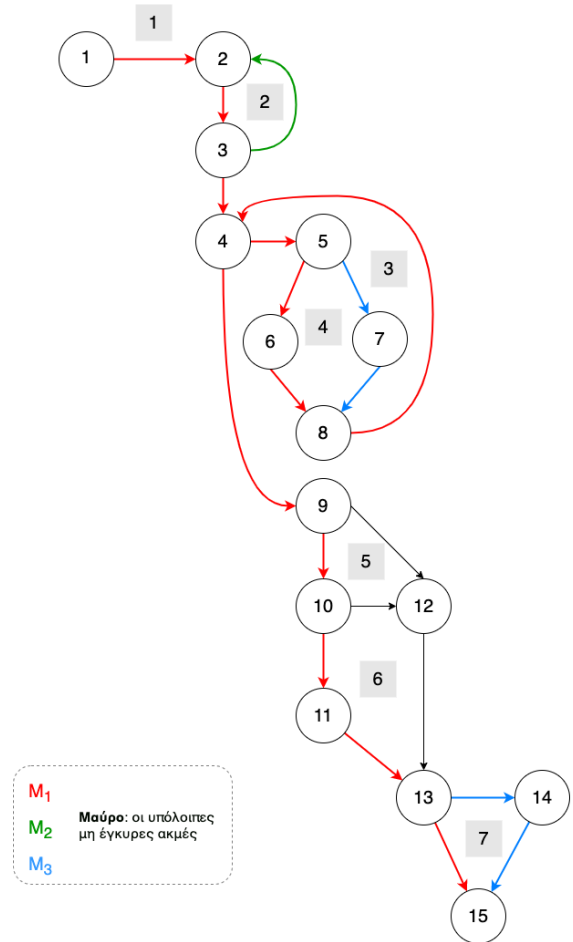
Υπάρχουν 7 διαφορετικές περιοχές στο γράφημα.

## 4 ΕΡΩΤΗΜΑ

Το συντομότερο μονοπάτι είναι το  $M_0$ : 1-2-3-4-9-10-11-13-15. Σε αυτό το μονοπάτι ακολουθείται η δεύτερη διακλάδωση στον κόμβο 4, δηλαδή δεν εκτελείται η *while* (γρ. 12-16 / κόμβοι 5-6-7-8). Λόγω της  $E_1$ , αυτό το μονοπάτι είναι αδύνατο.

Για τις εξαρτήσεις συνύπαρξης ισχύει:

- $E_1$ : Κατά τη πρώτη εκτέλεση της *while* της γρ. 12, η συνθήκη  $b \neq a$  πάντα θα είναι αληθής, οπότε δεν υπάρχει περίπτωση να μην εκτελεστεί η *while*. Αυτό συμβαίνει γιατί το  $a$  δεν γίνεται να είναι 0 (αλλιώς ακόμα θα βρισκόμασταν στη *do-while* των γρ. 7-10), ενώ το  $b$  πάντα θα είναι 0 λόγω της αρχικοποίησής του. Επομένως σίγουρα θα πρέπει να εκτελεστεί έστω μια φορά το μονοπάτι 4-5-6-8 (ή 4-5-7-8). Επιπλέον, για να φτάσουμε στο κόμβο 9 (δηλαδή να υπάρχει το μονοπάτι 4-9), πρέπει να έχει τελειώσει η εκτέλεση της *while*, άρα είναι απαραίτητη η ύπαρξη του μονοπατιού 8-4.
- $E_2$ : Η συνθήκη  $a == b$  στην γρ. 19 της σύνθετης *if* προφανώς θα ισχύει πάντα, βρισκόμενη ακριβώς μετά την *while*. Επομένως για τον έλεγχο των συνθηκών της *if* είναι πάντα απαραίτητος ο έλεγχος της δεύτερης συνθήκης  $c \neq b$  (κόμβος 10), γιατί ποτέ δεν θα εξαχθεί συμπέρασμα αποκλειστικά με την πρώτη. Άρα το μονοπάτι 9-12 δεν γίνεται να υπάρξει.
- $E_3$ : Η συνθήκη  $c \neq b$  (γρ. 19 / κόμβος 10) δεν γίνεται να είναι αναληθής, εξαιτίας των αλληλεπάρλληλων αυξήσεων της *while* (γρ. 12-15). Επομένως η *else* συνθήκη της ίδιας *if* (γρ. 20) δεν θα ενεργοποιηθεί ποτέ. Άρα το μονοπάτι 10-12 δεν γίνεται να υπάρξει. Επομένως δεν μπορεί να υπάρξει και το μονοπάτι 12-13. Έτσι, συνδυαστικά για αυτή την *if* το μόνο πιθανό μονοπάτι είναι το 9-10-11-13.
- $E_4$ : Ο κόμβος 15, δηλαδή η *else* συνθήκη της τελικής *if* ενεργοποιείται μόνο στη περίπτωση που ο ακέραιος είναι μεγαλύτερος του 40. Επομένως ένα μονοπάτι για να είναι έγκυρο θα πρέπει να περιλαμβάνει ή τους κόμβους 7 και 14 (ακέραιος  $> 40$ ), ή τους 6 και 13 ( $0 < \text{ακέραιος} < 40$ ).



Επομένως το μικρότερο έγκυρο μονοπάτι είναι το  $M_1$ : 1-2-3-4-5-6-8-4-9-10-11-13-15.

Ακολουθώντας τον αλγόριθμο, δημιουργούμε και τα υπόλοιπα μονοπάτια προσθέτοντας τις εναπομείνουσες ακμές.

$M_2$ : 1-2-3-2-3-4-5-6-8-4-9-10-11-13-15

$M_3$ : 1-2-3-4-5-7-8-4-9-10-11-13-14-15

Τα δυνατά μονοπάτια όπου μπορεί να ελεγχθεί το πρόγραμμα είναι 3, άρα πράγματι  $V(g) = 7 \geq 3$ .

Τα τέσσερα μονοπάτια τα οποία ήταν αδύνατα αφορούν τις ακμές 3-4-9, 10-12, 9-12 και την συσχέτιση μεταξύ 7ου και 14ου κόμβου, και έτσι είναι αδύνατο να υπάρξουν δεδομένου του κώδικα. Τα υπόλοιπα μονοπάτια είναι έγκυρα και αφορούν τις περιπτώσεις παρακάτω:

## 5 ΕΡΩΤΗΜΑ

Μονοπάτι	Περιγραφή	Περίπτωση ελέγχου (input)	Αναμενόμενο αποτέλεσμα (έξοδος προγράμματος)
$M_1$	Δίνουμε ως είσοδο έναν θετικό ακέραιο μικρότερο του 40.	5	Number is 16
$M_2$	Δίνουμε ως είσοδο έναν ακέραιο ο οποίος είναι αρνητικός ή 0. Ξαναδίνουμε ως είσοδο έναν θετικό ακέραιο.	-2 20	Give a number please: Number is 31
$M_3$	Δίνουμε ως είσοδο έναν ακέραιο μεγαλύτερο του 40.	50	Number is 120