

1) Το πρόγραμμα για «τρέξει» θα πρέπει να εισάγουμε αριθμούς μεγαλύτερους του 0, για αριθμούς από 1 έως 39 προσθέτει τον αριθμό 11 στον εαυτό του και για τους αριθμούς από 40 και πάνω διπλασιάζεται και προστίθεται ο αριθμός 20

2)

Στο Give an integer please: εισάγουμε τον αριθμό που επιθυμούμε σε περίπτωση που βάλουμε λάθος αριθμό θα έχουμε το αποτέλεσμα που φαίνεται στο πρώτο screenshot. Ενώ στα υπόλοιπα screenshot εμφανίζονται τα αποτελέσματα (Number is) όπως περιεγράφηκαν στο πρώτο ερώτημα.

```
Give an integer please:0
Give an integer please:-1
Give an integer please:
```

```
Give an integer please:10
Number is 21
```

```
Give an integer please:23
Number is 34
```

```
Give an integer please:25
Number is 36
```

```
Give an integer please:30
Number is 41
```

```
Give an integer please:39
Number is 50
```

```
Give an integer please:40
Number is 100
```

```
Give an integer please:45
Number is 110
```

```
Give an integer please:50
Number is 120
```

```
Give an integer please:90
Number is 200
```

```
Give an integer please:329
Number is 678
```

3)

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int a, b=0, c=10;
```

```
1. do {
```

```
printf("Give an integer please:");
```

```
scanf("%d", &a);
```

```
}
```

```
2. while (a <= 0);
```

```
3. while (b!=a)
```

```
{
```

```
4. b++;
```

```
5. if (c>a) 6. a++;
```

```
7. else 8. c++;
```

```
}
```

```
if (9. a==b && 10. c!=b) 11. c++;
```

```
12. else 13. c=b;
```

```
14. if (c>50) 15. c=c+a;
```

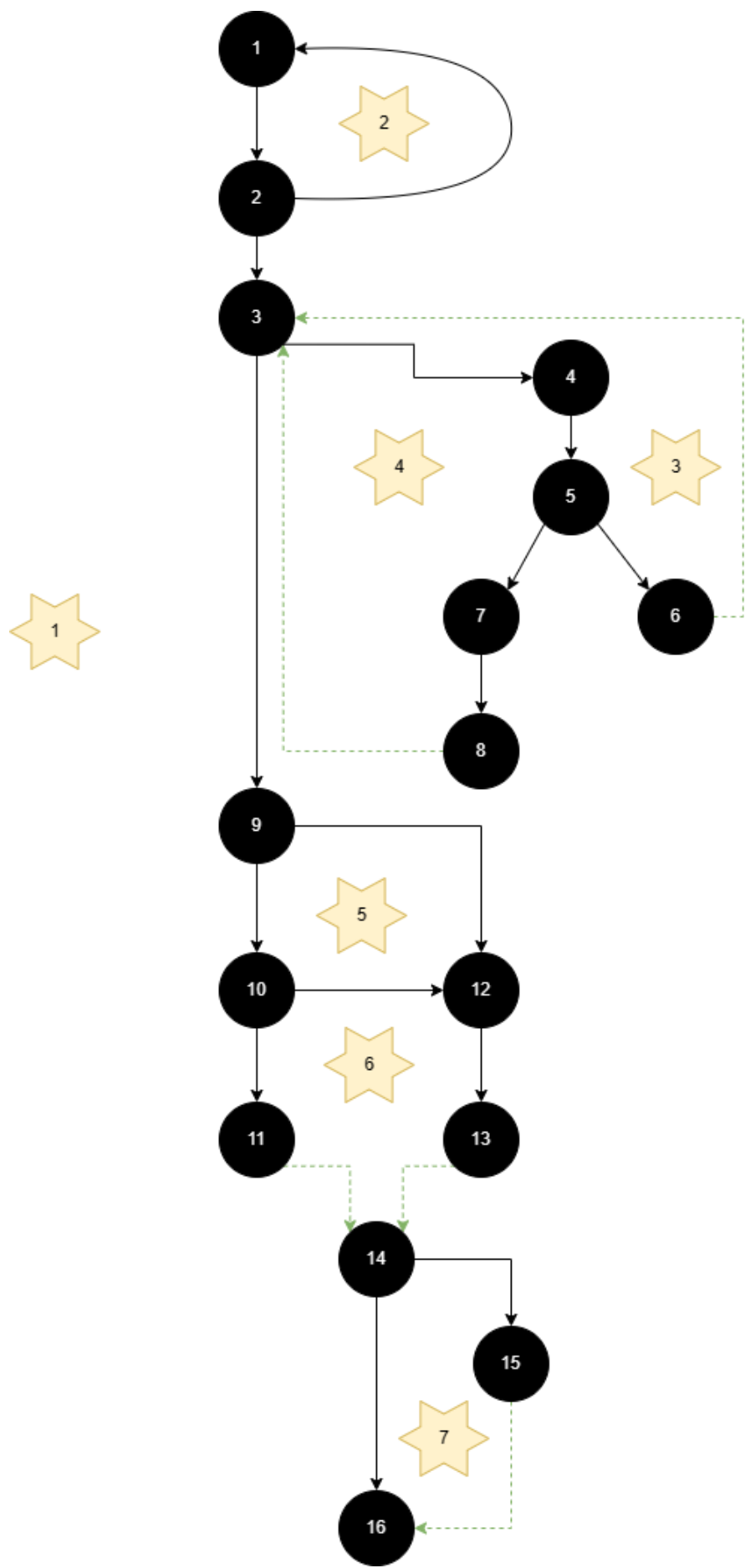
```
16. printf ("Number is %d \n", c);
```

```
return 0;
```

```
}
```

Παρατηρήσεις:

Δεν προσμετρώ την δήλωση μεταβλητών και την εισαγωγή των βιβλιοθηκών καθώς δεν επηρεάζει κάπου το σχεδιάγραμμα.



Κυκλωματική Πολυπλοκότητα:

$$1) V(G) = e - n + 2 * p = 21 - 16 + 2 * 1 = 7$$

$$2) V(G) = \text{εντολές} + 1 = 6 + 1 \text{ (εντολές} \rightarrow 1(\text{do-while}) + 1(\text{while}) + 1(\text{if-else}) + 2(\text{if-else}) + 1(\text{if})) = 7$$

$$3) V(G) = \text{αριθμός περιοχών του γράφου} = 7$$

4)

E1: Αν βρισκόμαστε στον κόμβο 2 τότε θα πρέπει αναγκαστικά ο χρήστης να εισάγει αριθμό μεγαλύτερο 0 έτσι ώστε να προχωρήσουμε στον κόμβο 3, διαφορετικά θα είμαστε μόνιμα στην λούπα 1-2.

E2: Αν βρισκόμαστε στον κόμβο 3 αναγκαστικά θα περάσουμε στον κόμβο 4 αφού το $b = 0$ και το $a \neq 0$, στην συνέχεια, ανάλογα με τον αριθμό που θα εισάγουμε θα περάσει από τους κόμβους 4, 5, 6, 7, 8 με οποιαδήποτε σειρά ανάλογα τον αρχικό αριθμό.

E3: Αν βρισκόμαστε στον κόμβο 9 αναγκαστικά θα μπαίνουμε στον κόμβο 10 και στην συνέχεια τον κόμβο 11, ύστερα από δοκιμές παρατήρησα ότι δεν παρατήρησα ότι στα βήματα 12 και 13 δεν μπαίνει ποτέ, οπότε θα έχουμε τα βήματα 9-10-11-14

E4: Στο βήμα 14 παρατηρούμε ότι μπαίνει ανάλογα τον αρχικό μας αριθμό. Για αριθμούς από 1-39 από το βήμα 11 πηγαίνει στο 16. Ενώ για αριθμούς από 40+ έχουμε την ακολουθία 14-15-16

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

$$M1: 1-2-[3-4-5-6-]^{11}-3-9-10-11-14-16$$

Το μικρότερο που μπορούμε να εισάγουμε είναι για όταν εισάγουμε το 1 για το οποίο θα έχουμε 10 επαναλήψεις συν τον αριθμό. Πχ για το 2 θα έχουμε 12 επαναλήψεις κ.ο.κ.

$$M2: 1-2-[3-4-5-6-]^n-3-9-10-11-14-16$$

Όπου n είναι ο αριθμός των επαναλήψεων. Το $M2$ ισχύει για αριθμούς από 2-9

$$M3: 1-2-[3-4-7-8-]^n-[3-4-5-6-3-4-7-8-]^n-9-10-11-16$$

Όπου n είναι ο αριθμός των επαναλήψεων. Το $M3$ ισχύει για αριθμούς από 10-39

$$M4: 1-2-[3-4-7-8-]^n-[3-4-5-6-3-4-7-8-]^n-9-10-11-14-15-16$$

Όπου n είναι ο αριθμός των επαναλήψεων. Το $M4$ ισχύει για αριθμούς από 40+

5)

Μονοπάτι	Περιγραφή	Input	Αναμενόμενο αποτέλεσμα
M1	Θα πρέπει να εισάγουμε τον αριθμό 1	1	12
M2	Θα πρέπει να εισάγουμε τον αριθμό 7	7	18
M3	Θα πρέπει να εισάγουμε τον αριθμό 30	30	41
M4	Θα πρέπει να εισάγουμε τον αριθμό 45	45	110

Για κάθε μονοπάτι υπάρχει μόνο **1** τιμή ελέγχου καθώς κάθε αριθμός έχει ένα διαφορετικό μονοπάτι με διαφορετικό συνδυασμό συγκεκριμένων ακολουθιών κόμβων.