

## 4<sup>η</sup> Εργασία

1) Αρχικά γίνεται η απαρίθμηση των κόμβων:

```
#include <stdio.h>
int main()
{
    int n, r = 0, t, flag=0;

    printf("Enter a number\n");
    scanf("%d", &n);

    t = n;

    while (t!=0 && r==0)
    {
        r = r * 10;
        r = r + t%10;
        t = t/10;

        if(t==0)
        {
            printf("error\n");
            flag=1;
        }
    }

    if (n==r)

        printf("%d The number n is: \n", n);

    else

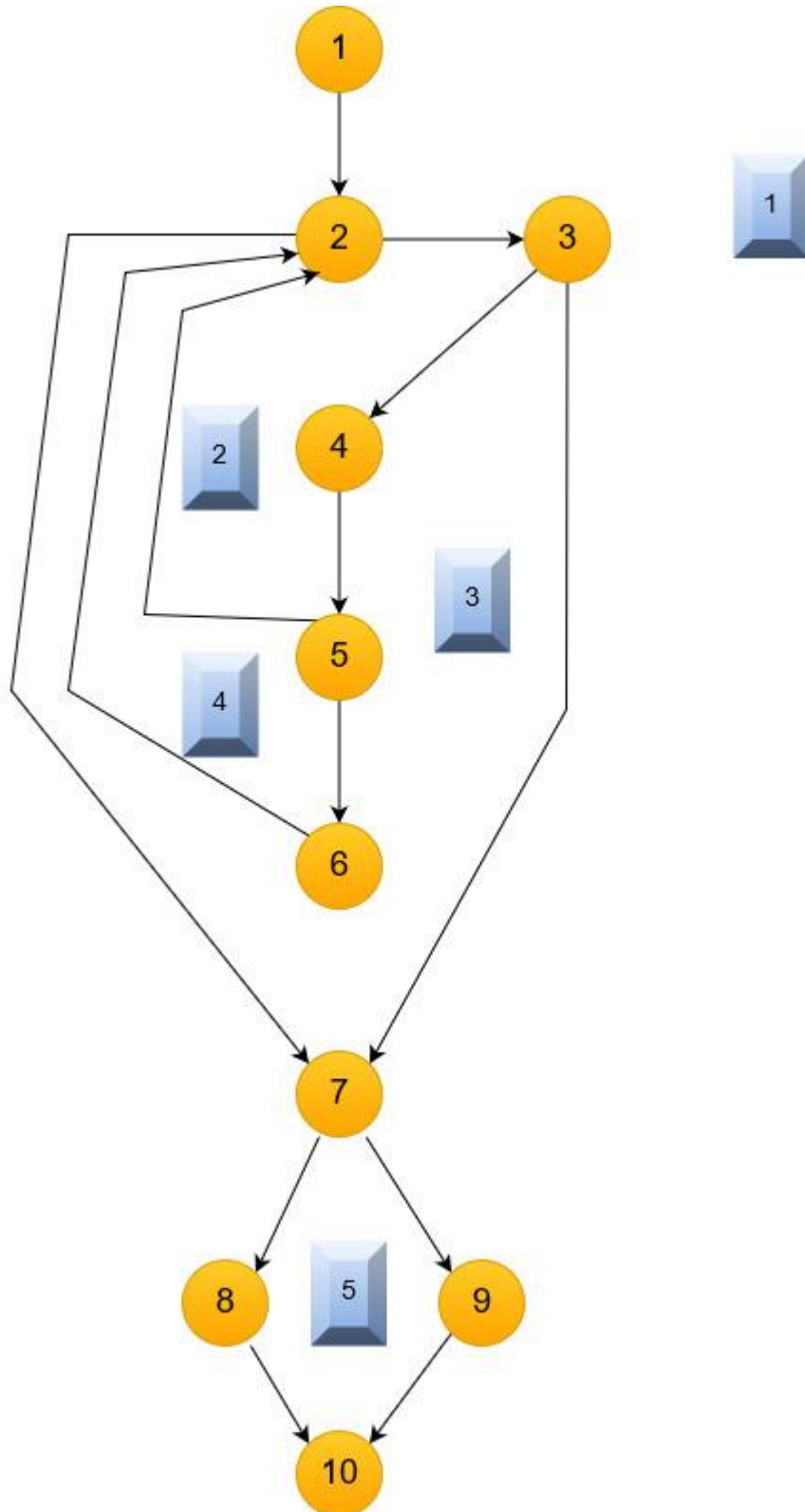
        printf("\n");

    return 0;
}
```

**Σημείωση:** όπως αναφέρεται στην διαφάνεια 33 των διαλέξεων 11-12 «οι εντολές exit, return μετράνε στις εντολές διακλάδωσης μόνο όταν δεν βρίσκονται στο τέλος τους προγράμματος ή της ρουτίνας»

οι απλές συνθήκες σημειώνονται με **κίτρινο** στον κώδικα.

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



## Υπολογισμός κυκλωματικής πολυπλοκότητας

### 1<sup>ος</sup> τρόπος

$$V(g) = e - n + 2p = 13 - 10 + 2 = 5,$$

όπου **e**: αριθμός ακμών **n**: αριθμός κορυφών **p**: συνεκτικές συνιστώσες

### 2<sup>ος</sup> τρόπος

$$V(g) = \text{Περιοχές γράφου} = 5 \text{ (όπως φαίνονται στο σχήμα)}$$

### 3<sup>ος</sup> τρόπος

$$V(g) = 4 \text{ απλές συνθήκες (while(=2) + if + if =4) + 1 = 5}$$

(οι απλές συνθήκες είναι εκείνες που έχουν σημειωθεί με κίτρινο στον κώδικα)

## **2) Εύρεση συντομότερου μονοπατιού**

Παρατηρώντας τον γράφο, βρίσκουμε ότι υπάρχουν 2 συντομότερα μονοπάτια, το **1-2-7-8-10** και το **1-2-7-9-10**. Από αυτά το **1-2-7-8-10** είναι έγκυρο, ενώ το **1-2-7-9-10** δεν μπορεί να εκτελεστεί, διότι αν από τον κόμβο 2 πάει κατευθείαν στον κόμβο 7, αυτό σημαίνει ότι  $t=n=r=0$ , επομένως η συνθήκη της if (7) θα είναι αληθής (το  $r$  παραμένει ίσο με την αρχική του τιμή (0) καθώς δεν θα εκτελεστεί καμία επανάληψη).

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

**E1.** Αν σε ένα μονοπάτι υπάρχει ο κόμβος 6 τότε υποχρεωτικά η επόμενη ακμή θα είναι η 2-7.

**E2.** Αν σε ένα μονοπάτι υπάρχει η ακμή 5-2, δεν μπορεί να υπάρχει ο κόμβος 8.

**E3.** Ο κόμβος 3 δεν μπορεί να ακολουθείται από την ακμή 7-8.

**E4.** Αν σε ένα μονοπάτι δεν υπάρχει ο κόμβος 3, τότε δεν μπορεί να υπάρχει η ακμή 7-9 (σημαίνει ότι έχει δοθεί ως είσοδος το 0).

Με βάση την παραπάνω παρατήρηση και τις εξαρτήσεις συνύπαρξης το μικρότερο έγκυρο μονοπάτι είναι:

**M1: 1-2-7-8-10**

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-**2-3-4-5-6-2**-7-8-10 νέες ακμές: 2-3, 3-4, 4-5, 5-6, 6-2

**M3:** 1-2-3-4-**5-2-3-7-9-10** νέες ακμές: 5-2, 3-7, 7-9, 9-10

Παρατηρούμε πως όλες οι ακμές συμπεριλαμβάνονται σε κάποιο βασικό μονοπάτι. Συνεπώς, το πρόγραμμα μπορεί να ελεγχθεί με 3 βασικά μονοπάτια, δηλαδή λιγότερα από την κυκλωματική πολυπλοκότητα η οποία αποτελεί άνω όριο των βασικών μονοπατιών.

3) Κάποιες ενδεικτικές περιπτώσεις ελέγχου για τα μονοπάτια είναι:

Μονοπάτι	Περιγραφή (Δίνουμε ως είσοδο:)	Περίπτωση ελέγχου (input)	Αναμενόμενο αποτέλεσμα (έξοδος προγράμματος)
M1	το 0.	0	Enter a number 0 0 The number n is:
M2	το 4.	4	Enter a number 4 error 4 The number n is:
M3	το 23.	23	Enter a number 23