Περιεχόμενα

Ζητούμενο 1

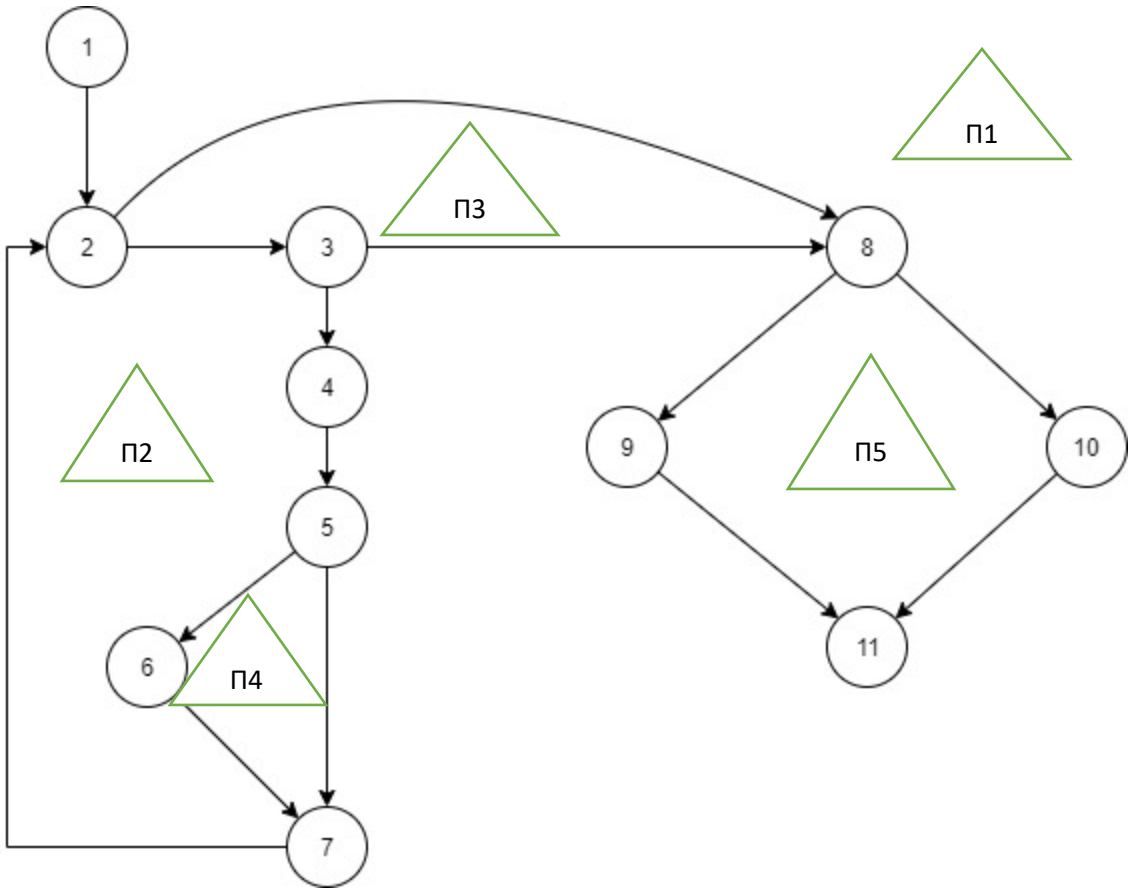
Κώδικας

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;                                                    Κόμβος 1

while (t!=0 && r==0 )          Κόμβος 2          Κόμβος 3

    {
                                  Κόμβος 4
     r = r * 10;
     r = r + t%10;
     t = t/10;

      if(t==0)   Κόμβος 5

       {
         printf("error\n");
         flag=1;
        }                                                 Κόμβος 6

   }                                                      Κόμβος 7

if (n==r) Κόμβος 8
   printf("%d The number n is: \n", n);
        Κόμβος 9
   else
   printf("\n");

        Κόμβος 10

   return 0;

   }

        Κόμβος 11
```

Γράφος ροής

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



Κυκλωματική πολυπλοκότητα

Από τον παραπάνω γράφο προκύπτει ότι η κυκλωματική πολυπλοκότητα είναι 5, γιατί:

V(g) = e – n + 2 = 14 – 11 + 2 = 5

Επίσης, V(g) = p + 1(όπου p οι απλές συνθήκες) = 4 + 1 = 5, αφού έχουμε 2 συνθήκες στη while(κόμβοι 2 κ 3), και μία σε κάθε if(κόμβοι 5 και 8).

Ακόμη, V(g) = περιοχές γράφου (όπως φαίνονται στο σχήμα με τη μορφή τριγώνων) =
Π1 + Π2 + Π3 + Π4 + Π5 = 5

Ζητούμενο 2

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. Ένα μονοπάτι που περιέχει τον κόμβο 2 και όχι τον κόμβο 3, θα περιέχει και τους κόμβους 8-9-11. Όμως, εάν περιέχει τους κόμβους 8-9-11, δε σημαίνει ότι δε θα περιέχει και τον κόμβο 3.

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

- M1: 1-2-8-9-11

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

- M2: 1 - 2 – 3 - 4 – 5 – 6 – 7 – 2 – 8 – 9 - 11
- M3: 1 - 2 - 3 - 4 - 5 - 7 - 2 - 3 - 8 - 10 - 11

Συνεπώς, το πρόγραμμά μπορεί να ελεγχθεί με 3 βασικά μονοπάτια, δηλαδή λιγότερα από την κυκλωματική πολυπλοκότητα η οποία αποτελεί άνω όριο των βασικών μονοπατιών.

Σημειώσεις:

1. όλες οι ακμές περιλαμβάνονται τουλάχιστον 1 φορά στα παραπάνω μονοπάτια.

2. κάθε μονοπάτι διαφέρει από τα άλλα τουλάχιστον σε μία ακμή,

3. αυτή είναι μία μόνο από τις ορθές λύσεις (δηλαδή αν ακολουθηθεί διαφορετική μέθοδος καταγραφής των μονοπατιών, το σύνολο των βασικών μονοπατιών μπορεί να είναι διαφορετικό αλλά πάντα θα είναι μεγέθους 3 και θα καλύπτει όλες τις ακμές τουλάχιστον 1 φορά).

Ζητούμενο 3

Κάποιες ενδεικτικές περιπτώσεις ελέγχου για τα μονοπάτια είναι:

Μονοπάτι	**Περιγραφή**	**Περίπτωση Ελέγχου(input)**	**Αναμενόμενο αποτέλεσμα(έξοδος προγράμματος)**
M1	Το μηδέν.	0	0 The number n is:
M2	Ένας αριθμός από το 1 μέχρι το 9 ή από το -9 μέχρι το -1	5	error 5 The number n is:
M3	Ένας αριθμός από το 11 μέχρι το +∞ ή από το -∞ μέχρι το -11(εκτός από όποτε αλλάζει το πρώτο ψηφίο πχ. 20, 40, 100, 300, 1000)	11	