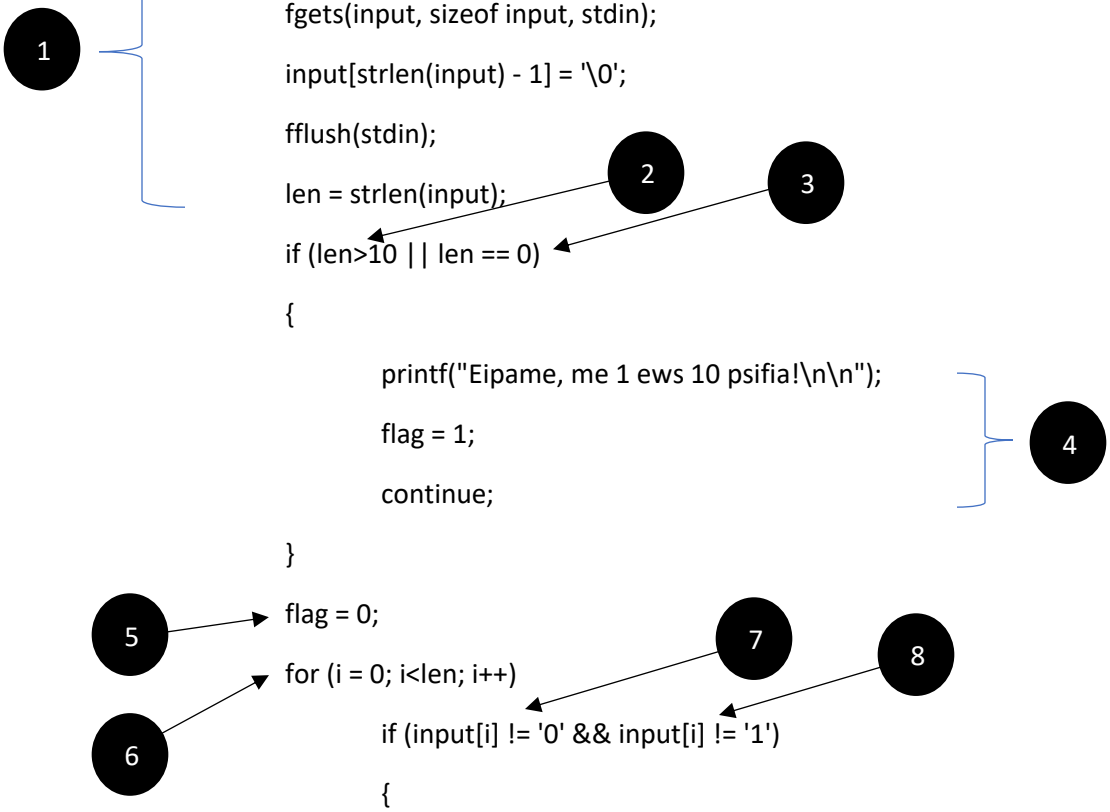
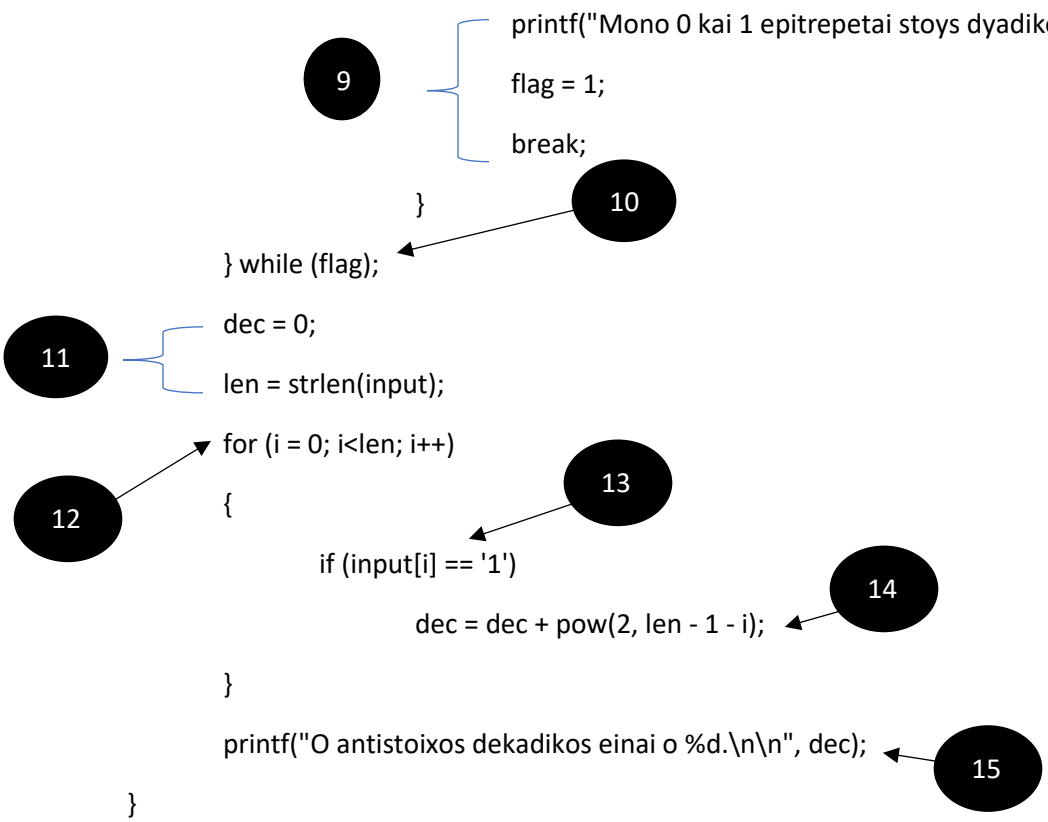
Αναφορά Λύσης 4ης Άσκησης

Ζητούμενο 1

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

void main(void)
{
	int i, flag, len, dec;
	char input[13];
	do {
		printf("Dwste ena dyadiko akeraio me 1 ews 10 psifia: ");
		fgets(input, sizeof input, stdin);
		input[strlen(input) - 1] = '\0';
		fflush(stdin);
		len = strlen(input);
		if (len>10 || len == 0)
		{
			printf("Eipame, me 1 ews 10 psifia!\n\n");
			flag = 1;
			continue;
		}
		flag = 0;
		for (i = 0; i<len; i++)
			if (input[i] != '0' && input[i] != '1')
			{
```





Σημειώσεις:

1. Χρησιμοποιώ το κίτρινο αστέρι για να δείξω τις περιοχές του γράφου.

2. Χρησιμοποιώ το μαύρο κυκλάκι για να δείξω τους κόμβους του γράφου.

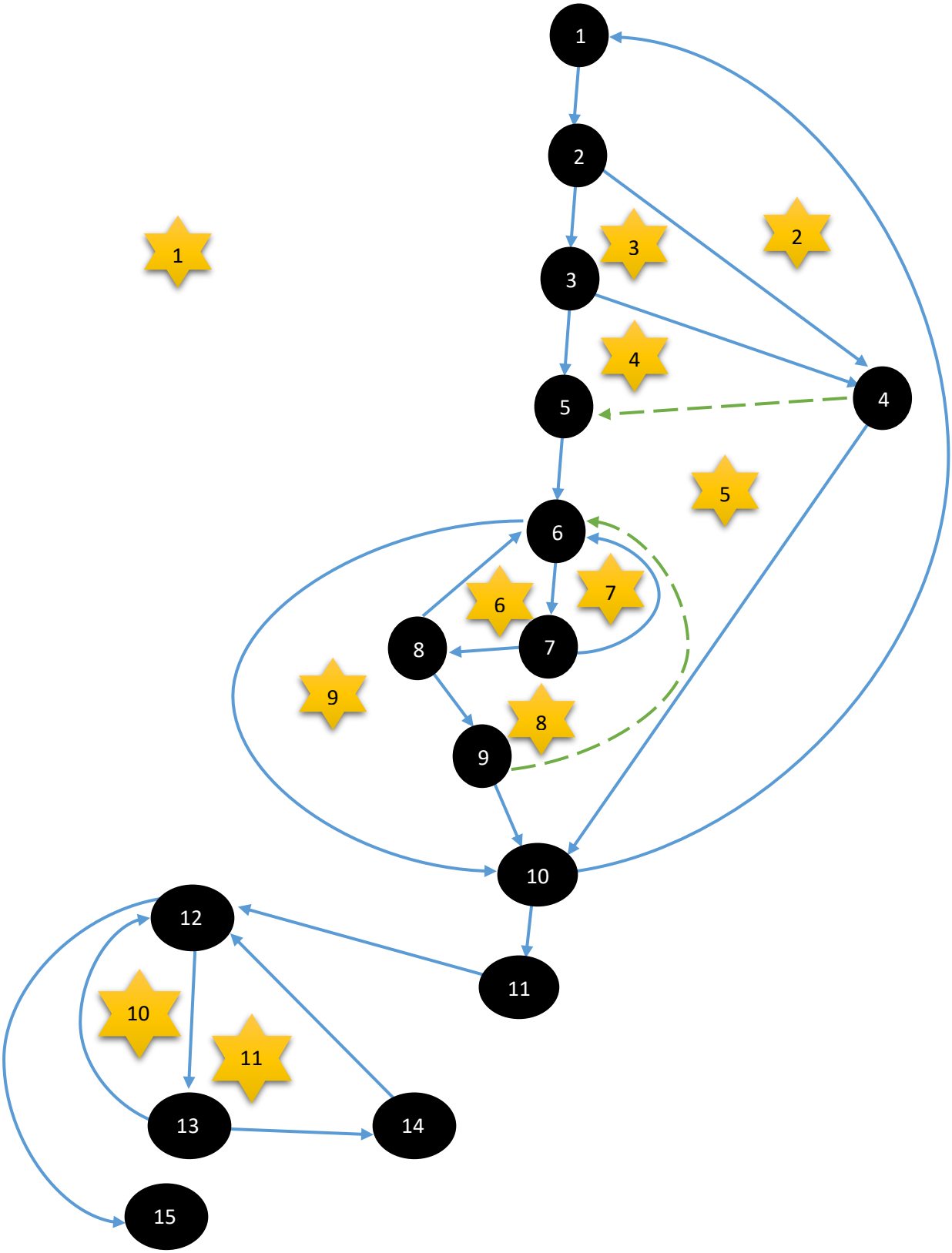
3. Στο εσωτερικό των παραπάνω σχημάτων, φαίνεται η αρίθμηση, η οποία έχει χρησιμοποιηθεί για όλα τα ερωτήματα.

4. Τα σχόλια έχουν αφαιρεθεί για διευκόλυνση της αρίθμησης και της αναγνωσιμότητας του κώδικα.

5. Κατά την αρίθμηση, θεωρώ ότι η ροή του κώδικα γίνεται μεταξύ των εντολών, οπότε δεν λαμβάνω υπόψη τα bracket {}.

6. Αγνοώ τις δηλώσεις μεταβλητών και την εισαγωγή των βιβλιοθηκών, καθώς δεν επηρεάζουν την αρίθμηση την οποία έχω χρησιμοποιήσει.

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



Κυκλωματική Πολυπλοκότητα:

1) V(g) = e – n + 2 * p = 24 – 15 + 2 * 1 = 11

2) V(g) = p + 1 = 10 + 1 (p-> 1(continue) + 1(break) + 1(do-while) + 1(for) + 1(for) + 2(if) + 2(if) +1(if))

3) V(g) = αριθμός περιοχών του γράφου = 11

Σύμφωνα με τους τρεις παρακάτω τρόπους υπολογισμού, για τον παραπάνω γράφο, προκύπτει ότι η κυκλωματική πολυπλοκότητα ισούται με 11.

Ζητούμενο 2

Εξαρτήσεις Συνύπαρξης:

- E1: Αν σε ένα μονοπάτι υπάρχει ο κόμβος 4, το μήκος του εισαχθέντος αλφαριθμητικού είναι ίσο με το 0 ή μεγαλύτερο του 10, οπότε το μονοπάτι πρέπει να τερματίζει ξανά στον κόμβο 1.

- E2: Αν σε ένα μονοπάτι υπάρχει ο κόμβος 5, το μήκος του εισαχθέντος αλφαριθμητικού είναι μικρότερο του 10 και μεγαλύτερο του 0, οπότε πρέπει να εμφανίζονται οι κόμβοι 7 και 13 στο μονοπάτι.

- E3: Αν σε ένα μονοπάτι υπάρχει ο κόμβος 7 και δεν εμφανίζονται οι κόμβοι 8 και 9, όλοι οι χαρακτήρες του εισαχθέντος αλφαριθμητικού είναι ίσοι του “0”, οπότε δεν μπορεί να εμφανίζεται ο κόμβος 14 στο μονοπάτι.

- E4: Αν σε ένα μονοπάτι υπάρχει ο κόμβος 8 και δεν εμφανίζεται ο κόμβος 9, υπάρχει τουλάχιστον μία φορά ο χαρακτήρας “1” εντός του εισαχθέντος αλφαριθμητικού, οπότε πρέπει να εμφανίζεται ο κόμβος 14 στο μονοπάτι.

- E5: Αν σε ένα μονοπάτι υπάρχει ο κόμβος 9, οι χαρακτήρες του εισαχθέντος αλφαριθμητικού είναι διάφοροι του χαρακτήρα “0” ή του χαρακτήρα “1”, οπότε το μονοπάτι πρέπει να τερματίζει ξανά στον κόμβο 1.

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

M1: 1 – 2 – 3 – 5 – 6 – 7 – 6 – 10 – 11 – 12 – 13 – 12 – 15

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

M2: 1 – 2 – 4 – 10 – 1 – 2 – 3 – 5 – 6 – 7 – 6 – 10 – 11 – 12 – 13 – 12 – 15

M3: 1 – 2 – 3 – 4 – 10 – 1 – 2 – 3 – 5 – 6 – 7 – 6 – 10 – 11 – 12 – 13 – 12 – 15

M4: 1 – 2 – 3 – 5 – 6 – 7 – 8 – 6 – 10 – 11 – 12 – 13 – 14 – 12 – 15

M5: 1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 1 – 2 – 3 – 5 – 6 – 7 – 6 – 10 – 11 – 12 – 13 – 12 – 15

Σε αυτό το σημείο, οι μόνες ακμές που δεν συμπεριλαμβάνονται σε κανένα βασικό μονοπάτι είναι οι ακμές 4->5 και 9->6. 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 5 βασικά μονοπάτια, δηλαδή λιγότερα από την κυκλωματική πολυπλοκότητα(V(g) = 11), η οποία αποτελεί άνω όριο των βασικών μονοπατιών. Επίσης, θεωρώ ότι τα μονοπάτια M1, M2, M3, τελειώνουν στον κόμβο 1, καθώς σε αυτό το σημείο ο χρήστης πρέπει να εισάγει ξανά ένα αλφαριθμητικό εισόδου προκειμένου να συνεχιστεί η εκτέλεση του προγράμματος.

Ζητούμενο 3

Μονοπάτι	**Περιγραφή**	**Περίπτωση Ελέγχου(input)**	**Αναμενόμενο Αποτέλεσμα(Έξοδος προγράμματος)**
M1	Θα πρέπει να δοθεί αλφαριθμητικό με μήκος μεγαλύτερο του 10, ανεξαρτήτου χαρακτήρα.	11111111111	"Eipame, me 1 ews 10 psifia!"
M2	Θα πρέπει να δοθεί αλφαριθμητικό με μήκος ίσο με 0.		"Eipame, me 1 ews 10 psifia!"
M3	Θα πρέπει το αλφαριθμητικό να έχει μήκος μεταξύ του 1 και του 10, ενώ θα πρέπει τουλάχιστον ένας χαρακτήρας του αλφαριθμητικού να	1a	"Mono 0 kai 1 epitrepetai stoys dyadikoys arithmous!"

	είναι διάφορος του “1” ή του “0”.		
M4	Θα πρέπει το αλφαριθμητικό να έχει μήκος μεταξύ του 1 και του 10, ενώ θα πρέπει όλοι οι χαρακτήρες του αλφαριθμητικού να είναι “0”.	00	"O antistoixos dekadikos einai o 0."
M5	Θα πρέπει το αλφαριθμητικό να έχει μήκος μεταξύ του 1 και του 10, ενώ θα πρέπει οι χαρακτήρες του αλφαριθμητικού να είναι “0” ή “1”.	01	"O antistoixos dekadikos einai o 1."

Σημείωση: Λόγω της χρήσης της συνάρτησης fgets(), όλα τα αλφαριθμητικά εισόδου πρέπει να τελειώνουν με τον χαρακτήρα “\n”(newline character), με σκοπό την επιτυχή είσοδο του δεδομένου αλφαριθμητικού.