Α' Εξάμηνο
ΕΙΣΑΓΩΓΗ ΣΤΟ ΔΟΜΗΜΕΝΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

Ερωτήσεις Επανάληψης

Α. Ερωτήσεις

Στο τέλος δίνονται υποδείξεις για την απάντηση μερικών ερωτήσεων.

01

Ερώτηση: Στις παρακάτω εντολές τι πρόβλημα υπάρχει;

```
int i = 1;
double a = 0.5;
printf("Οι αριθμοί i και a είναι %d και %lf \n", i, a);
```

Απάντηση:

Α.Υπάρχουν μεταβλητές δύο διαφορετικών τύπων (int και float) στην ίδια εντολή printf
Β.Οι προδιαγραφές μορφοποίησης για τους δύο αριθμούς διαφέρουν ενώ έπρεπε να είναι ίδιες
Γ. Κανένα πρόβλημα.

02

Ερώτηση: Στον παρακάτω κώδικα, τι τιμή παίρνει τελικά η μεταβλητή iter;

```
int iter;
for(iter=0; iter <= 1000; iter++)
    printf("Επαναλήψεις = %d", iter+1);
```

Απάντηση:

Α. 999
Β. 1000
Γ. 1001

03

Ερώτηση: Στον παρακάτω κώδικα, τι τιμή παίρνει τελικά η μεταβλητή n;

```
int iter, n = 0;
for(iter=0; iter < 1000; iter++) n++;
```

Απάντηση:

Α. 999
Β. 1000
Γ. 1001

04

Ερώτηση: Η παρακάτω εντολή είναι σωστή ή λάθος; Υποθέστε ότι η προδιαγραφή μορφοποίησης είναι η κατάλληλη για τη μεταβλητή my_string.

```
scanf("%s", my_string);
```

Απάντηση:

Α. Σωστή, γιατί το my_string δε μπορεί παρά να είναι δείκτης ή πίνακας τύπου char
Β. Λάθος, γιατί λείπει ο τελεστής διεύθυνσης από τη μεταβλητή
Γ. Εξαρτάται από τη δήλωση του my_string (αν είναι πίνακας char ή δείκτης char*)

05

Ερώτηση: Ο παρακάτω ορισμός μιας συμβολικής σταθεράς είναι σωστός ή λάθος;

```
#define BIG 10000 /* δέκα χιλιάδες */
```

Απάντηση:

Α. Σωστός
Β. Λάθος. Λείπει ο τελεστής ανάθεσης =
Γ. Λάθος. Λείπει ο τελεστής ανάθεσης = και το ελληνικό ερωτηματικό στο τέλος

06

Ερώτηση: Τι τιμή θα πάρει το x μετά από την εκτέλεση των εντολών στην περίπτωση (α) και στην περίπτωση (β);

(α)
```
int x, y = 0;
x = y++;
```

(β)
```
int x, y = 0;
x = ++y;
```

Απάντηση:
Α. 0 και στις δύο περιπτώσεις
Β. 1 στην (α) και 0 στη (β)
Γ. 0 στην (α) και 1 στη (β)

07

Ερώτηση: Ποιο είναι το πρόβλημα στις παρακάτω εντολές;

```
scanf("%f", &x);
x1 = x*(x-1.0)*(x-2.0);
if( x1 );
    printf("Το αποτέλεσμα είναι %f \n", 1.0/x1);
```

Απάντηση:
Α. Κανένα. Το πρόγραμμα μεταγλωττίζεται κανονικά.
Β. Η if πρέπει να γραφτεί ως εξής: `if (x1 != 0);`
Γ. Η ύπαρξη του ; μετά από τη συνθήκη `( x1 )` συνεπάγεται λογικό σφάλμα.

08

Ερώτηση: Αν x = 1, y = 2, z = 3, τότε ποια η τιμή του N μετά από την εκτέλεση της παρακάτω εντολής;

```
N = (((N = ( x < y ) ? x : y)) < z ? N : z);
```

Απάντηση:
Α. 1
Β. 2
Γ. 3

09

Ερώτηση: Τι πρόβλημα υπάρχει με την κλήση της συνάρτησης στις παρακάτω γραμμές;

```
int *x, y = 0;
*x = ++y;
someFunc(x, y);
```

Απάντηση:
Α. Κανένα πρόβλημα
Β. Το ένα όρισμα περνάει κατά διεύθυνση και το άλλο κατά τιμή, ενώ έπρεπε να περνάνε και τα δύο με τον ίδιο τρόπο.
Γ. Στο πρώτο όρισμα έπρεπε να χρησιμοποιήσουμε τον τελεστή διεύθυνσης (&).

10

Ερώτηση: Τι αποτέλεσμα έχει η παρακάτω εντολή;

```
int data[1000] = {0};
```

Απάντηση:
Α. Δήλωση ακέραιου πίνακα με 1000 στοιχεία από τα οποία το πρώτο έχει τιμή μηδέν
Β. Δήλωση ακέραιου πίνακα με 1000 στοιχεία. Το όνομα αυτού (data) είναι και σταθερά δείκτη

στον πίνακα και αρχικά δείχνει στη διεύθυνση NULL (δηλαδή 0).
Γ. Δήλωση ακέραιου πίνακα με 1000 στοιχεία που παίρνουν όλα αρχική τιμή μηδέν.

11

Ερώτηση: Τι συμβαίνει όταν μία εξωτερική μεταβλητή χρησιμοποιείται σε μία συνάρτηση όπου υπάρχει και μία τοπική μεταβλητή με το ίδιο όνομα;
Απάντηση:
Α. Σφάλμα μεταγλώττισης. Μία από τις δύο πρέπει να αλλάξει όνομα.
Β. Η τοπική μεταβλητή υπερισχύει της εξωτερικής.
Γ. Σφάλμα εκτέλεσης.

12

Ερώτηση: Τι τιμή έχει η παρακάτω παράσταση για a = 5, b = 4, c = 3 και d = 2;
```
d + a * c / b
```

Απάντηση:
Α. 2
Β. 5
Γ. τίποτε από τα παραπάνω

13

Ερώτηση: Αν η μεταβλητή y αρχικά είναι ίση με -1 και η x είναι ίση με 0, ποια είναι η τιμή της y μετά από την εκτέλεση της επόμενης εντολής;
```
if(x++) y += x;
```

Απάντηση:
Α. 0
Β. 1
Γ. -1

14

Ερώτηση: Τι συμβαίνει όταν εκτελούνται οι παρακάτω εντολές;
```
sum = 0;
for(x = 0; x < MAX; sum += ++x);
```

Απάντηση:
Α. Αυξάνεται το x έως MAX-1 χωρίς να επέλθει καμία άλλη αλλαγή
Β. Υπολογίζεται το άθροισμα των αριθμών από 1 έως και MAX και αποδίδεται στη μεταβλητή sum
Γ. Τίποτε από τα παραπάνω

15

Ερώτηση: Τι περιεχόμενο θα έχει το αλφαριθμητικό a μετά από την εκτέλεση των παρακάτω;
```
char a[80]={'\0'}, b[]= "Kozani city", *p, *q;
p = a;
for(q = b; *q != '\0'; p++, q++)
    *p = (*q == ' ') ? '_' : *q;
```

Απάντηση:
Α. "Kozani city"
Β. "Kozani_city"
Γ. "Kozani"

16

Ερώτηση: Αν ορίσουμε
```
char *cities[] = {"Athens","Ioannina","Kozani"};
```
τότε που βρίσκεται το γράμμα z;

Απάντηση:
Α. Στο cities[2][3]
Β. Στο cities[2][2]
Γ. Στο cities[3][3]

17

Ερώτηση: Ποια είναι η τιμή της απόστασης που θα τυπωθεί με την εκτέλεση των παρακάτω;
```
struct simio {
    int x;
    int y;
} p = {0.0, 0.0}, *q = &p;
double d = sqrt(pow((double) ++q->x,2.0) + pow((double) ++q->y,2.0));
printf("Απόσταση από την αρχή: %lf\n", d);
```

Υπενθυμίζεται ότι η συνάρτηση pow(x, y) υψώνει το x στη δύναμη του y ενώ η sqrt(x) υπολογίζει την τετραγωνική ρίζα του x.

Απάντηση:
Α. 0.0
Β. 1.0
Γ. 1.414214 (ρίζα του 2)

18

Ερώτηση: Υπάρχει λάθος στην παρακάτω δήλωση δομής;Αν ναι, ποιο είναι αυτό;
```
struct set {
    int x;
    int y;
    double z;
    char x;
};
```

Απάντηση:
Α. Δεν υπάρχει λάθος.
Β. Η δομή έχει μέλη με διαφορετικούς τύπους ενώ αυτό δεν επιτρέπεται
Γ. Η δομή έχει δύο μέλη με το ίδιο όνομα ενώ αυτό δεν επιτρέπεται

19

Ερώτηση: Τι αποτέλεσμα θα τυπωθεί με την εκτέλεση των παρακάτω εντολών;
```
int x = 1, y, *px = &x, *py = &y, *temp;
y = x++;
temp = py; py = px; px = temp;
printf("Αποτέλεσμα: %d\n", *px-*py);
```

Απάντηση:
Α. -1
Β. 0
Γ. 1

20

Ερώτηση: Τι μέγεθος σε bytes μπορεί να έχει ο παρακάτω πίνακας;

```
char pinakas[] = "Enas pinakas";
```

Απάντηση:
Α. 12
Β. 13
Γ. 96

21

Ερώτηση: Υπάρχει πρόβλημα στην παρακάτω συνάρτηση;(και ποιο;)

```
int input_value(int *x) {
    printf("Give me a number: ");
    scanf("%d", &x);
    return (*x);
}
```

Απάντηση:
Α. Όχι, είναι σωστή
Β. Ναι, περιττεύει ο τελεστής διεύθυνσης, &, στη scanf
Γ. Ναι, η συνάρτηση είναι τύπου void, αν και έχει ορίσματα

22

Ερώτηση: Τι έχετε να παρατηρήσετε σχετικά με τις επόμενες εντολές;

```
int x = 1;
while( x > 0) printf("Η τιμή είναι %d\n", x++);
```

Απάντηση:
Α. Η printf δε θα εκτελεστεί ούτε μία φορά
Β. Η printf θα εκτελεστεί μία φορά και θα τυπώσει την τιμή 1. Μετά, το x θα αυξηθεί κατά 1.
Γ. Ατέρμονος βρόχος· η printf θα τυπώνεται συνέχεια για διαδοχικά x

23

Ερώτηση: Τι θα έχει συμβεί στα ορίσματα της παρακάτω συνάρτησης όταν εκτελεστεί και επιστρέψει τον έλεγχο;

```
void swap(int *x, int *y) {
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
```

Απάντηση:
Α. Το ένα θα έχει πάρει την τιμή του άλλου
Β. Τίποτα.
Γ. Θα έχουν και τα δύο την ίδια τιμή.

24

Ερώτηση: Τι τιμή εκτυπώνεται όταν εκτελεστούν οι επόμενες εντολές;

```
char city[] = "Kozani city";
printf("String size: %d\n", sizeof(city)/sizeof(*city));
```

Απάντηση:
Α. 10
Β. 11
Γ. 12

25

Ερώτηση: Σε τι μπορεί να χρησιμεύουν οι παρακάτω εντολές;

```
printf("Press Enter...");
while((c = getchar()) != '\n') { }
```

Απάντηση:
Α. Στη συνεχή εισαγωγή χαρακτήρων από το πληκτρολόγιο
Β. Στην προσωρινή παύση της εκτέλεσης ενός προγράμματος μέχρι ο χρήστης να πατήσει Enter
Γ. Στην εισαγωγή ενός χαρακτήρα αλλαγής γραμμής από το πληκτρολόγιο

26

Ερώτηση: Πώς μπορούμε να παραστήσουμε την τιμή του στοιχείου x[i] ενός πίνακα x με αριθμητική δεικτών;

Απάντηση:
Α. Ως `x+i`
Β. Ως `&x+i`
Γ. Ως `*(x+i)`

27

Ερώτηση: Το παρακάτω πρόγραμμα χρησιμοποιεί μια αναδρομική συνάρτηση για να τυπώσει ένα αλφαριθμητικό, αλλά έχει ένα bug. Ποιο είναι αυτό;;

```
1 #include <stdio.h>
2 void display(char *c) {
3     if(c != '\0') {
4         putchar(*c);
5         display(++c);  }
6 }
7 int main() {
8     char s[] = "Kozani city";
9     display(s);
10 }
```

Απάντηση:
Α. Στη γραμμή 3 πρέπει να γράφει *c αντί για c
Β. Στη γραμμή 4 πρέπει να γράφει c αντί για *c
Γ. Στη γραμμή 9 πρέπει να γράφει &s αντί για s

28

Ερώτηση: Ποιο από τα παρακάτω δε μπορεί να χρησιμοποιηθεί ως όνομα μεταβλητής;

Απάντηση:
Α. 7times
Β. times_7
Γ. _seven_times

29

Ερώτηση: Υπάρχει λάθος στην παρακάτω δήλωση δομής;Αν ναι, ποιο είναι αυτό;

```
struct set {
    int x;
    int y;
    double *z;
    struct set *next;
};
```

Απάντηση:
Α. Δεν υπάρχει λάθος
Β. Η δομή έχει μέλη δείκτες ενώ αυτό δεν επιτρέπεται
Γ. Η δομή έχει μέλος του ίδιου τύπου με τον εαυτό της ενώ αυτό δεν επιτρέπεται

30

Ερώτηση: Τι θα έχει συμβεί στα ορίσματα της παρακάτω συνάρτησης όταν εκτελεστεί και επιστρέψει τον έλεγχο;

```
void swap(int x, int y) {
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```

Απάντηση:
Α. Το ένα θα έχει πάρει την τιμή του άλλου
Β. Τίποτα.
Γ. Θα έχουν και τα δύο την ίδια τιμή.

31

Ερώτηση: Ποια από τις παρακάτω εντολές θέλει απαραιτήτως ελληνικό ερωτηματικό στο τέλος;

Απάντηση:
Α.for
Β. while
Γ. do...while

32

Ερώτηση: Τι τιμή εκτυπώνεται όταν εκτελεστούν οι επόμενες εντολές;

```
char city[] = {'K', 'o', 'z', 'a', 'n', 'i', ' ', 'c', 'i', 't',
'y'};
printf("String size: %d\n", sizeof(city)/sizeof(*city));
```

Απάντηση:
Α. 10
Β. 11
Γ. 12

33

Ερώτηση:Τι πρόβλημα μπορεί να υπάρξει αν μια αναδρομική συνάρτηση κληθεί επαναληπτικά πάρα πολλές φορές;

Απάντηση:

Α. Ατέρμονος βρόχος.
Β. Θα εξαντληθεί η στοίβα με αποτέλεσμα να μην επαρκεί η μνήμη
Γ. Κανένα πρόβλημα

34

Ερώτηση: Τι αποτέλεσμα θα έχει η εντολή break στο παρακάτω τμήμα προγράμματος;

```
for(i=0; i<1000; i++) {
    for(j=i+1; j<1000; j++) {
        ...
        if(x[i][j] < 0) break;
        ...
    }
}
```

Απάντηση:
Α. Η ροή της εκτέλεσης θα φύγει από το εσωτερικό for (αυτό με το μετρητή j) και θα μεταβιβαστεί στην αμέσως επόμενη εντολή που υπάρχει στο εξωτερικό for (αυτό με το μετρητή i)
Β. Η ροή της εκτέλεσης θα φύγει και από τα δυο for και θα πάει στην αμέσως επόμενη εντολή του προγράμματος.
Γ. Θα διακοπεί η τρέχουσα επανάληψη και το πρόγραμμα θα αρχίσει αμέσως την επόμενη επανάληψη (για την τρέχουσα τιμή του i και την επόμενη τιμή του j)

35

Ερώτηση: Πώς εξισώνουμε γενικά έναν πίνακα A με έναν άλλο πίνακα B;

Απάντηση:
Α. Με εντολή απόδοσης τιμής A = B;
Β. Με δομή επανάληψης, π.χ. for, κατά την οποία εξισώνονται τα στοιχεία ένα-ένα
Γ. Με εντολή απόδοσης τιμής *A = *B;

36

Ερώτηση: Με δεδομένες τις παρακάτω δηλώσεις, τι ισχύει για το συμβολισμό p+i;

```
int i, *p;
```

Απάντηση:
Α. παριστάνει μία θέση μνήμης που βρίσκεται κατά i bytes μετατοπισμένη σε σχέση με αυτή όπου δείχνει ο δείκτης p.
Β. παριστάνει μία θέση μνήμης που βρίσκεται κατά iN bytes μετατοπισμένη σε σχέση με αυτή όπου δείχνει ο δείκτης p, αν N bytes είναι το μέγεθος του τύπου int.
Γ. είναι μία παράσταση όπου η μεταβλητή p μετατρέπεται αυτόματα στον ανώτερο τύπο της παράστασης (εδώ int) και τελικά, το αποτέλεσμα είναι τύπου int και ισούται με το άθροισμα των p και i.

37

Ερώτηση: Πώς μπορούμε να παραστήσουμε τη διεύθυνση του στοιχείου x[i] ενός πίνακα x με αριθμητική δεικτών;

Απάντηση:
Α. Ως x+i
Β. Ως &(x+i)
Γ. Ως *(x+i)

38

Ερώτηση:Τι αποτέλεσμα θα έχει η εντολή goto στο παρακάτω τμήμα προγράμματος;

```
for(i=0; i<1000; i++) {
    for(j=i+1; j<1000; j++) {
        ...
        if(x[i][j] < 0) goto label;
        ...
    }
}
...
label: printf("Βρέθηκαν αρνητικά στοιχεία\n");
```

Απάντηση:
Α. Η ροή της εκτέλεσης θα φύγει από το εσωτερικό for (αυτό με το μετρητή j) και θα μεταβιβαστεί στην αμέσως επόμενη εντολή που υπάρχει στο εξωτερικό for (αυτό με το μετρητή i). Όταν τελειώσει το εξωτερικό for η ροή της εκτέλεσης θα πάει κατευθείαν στην εντολή που έχει την ετικέτα label υπερπηδώντας όλες τις ενδιάμεσες.
Β. Η ροή της εκτέλεσης θα φύγει και από τα δύο for και θα πάει κατευθείαν στην εντολή που έχει την ετικέτα label υπερπηδώντας όλες τις ενδιάμεσες.
Γ. Θα διακοπεί η τρέχουσα επανάληψη και το πρόγραμμα θα αρχίσει αμέσως την επόμενη επανάληψη (για την τρέχουσα τιμή του i και την επόμενη τιμή του j). Όταν ολοκληρωθούν οι υπόλοιπες επαναλήψεις η ροή της εκτέλεσης θα πάει κατευθείαν στην εντολή που έχει την ετικέτα label υπερπηδώντας όλες τις ενδιάμεσες.

39

Ερώτηση: Ποια ή ποιες συναρτήσεις πρέπει να υπάρχουν απαραίτητα σε κάθε πρόγραμμα γραμμένο σε C;

Απάντηση:
Α. Η συνάρτηση main
Β. Οι main και printf
Γ. Καμία. Ο προγραμματιστής ορίζει όλες τις συναρτήσεις που χρειάζεται κάθε φορά.

40

Ερώτηση: Τι σημαίνει η παρακάτω εντολή;

```
if(x = someFunction(y)) printf("Αποτέλεσμα: %f", x);
```

Απάντηση:
Α. Αν η μεταβλητή x είναι ίση με την τιμή που επιστρέφει η συνάρτηση someFunction για όρισμα ίσο με y τότε τύπωσε το x
Β. Το x θα πάρει τιμή ίση με αυτή που επιστρέφει η συνάρτηση someFunction για όρισμα ίσο με y. Αν αυτή είναι διαφορετική από μηδέν τότε τύπωσε το x.
Γ. Το x θα πάρει τιμή ίση με αυτή που επιστρέφει η συνάρτηση someFunction για όρισμα ίσο με y. Αν αυτή είναι ίση με μηδέν τότε τύπωσε το x.

41

Ερώτηση: Τι έχετε να παρατηρήσετε σχετικά με τις επόμενες εντολές;

```
int x = -1;
while( x ) printf("Η τιμή είναι %d\n", x++);
```

Απάντηση:
Α. Η printf δε θα εκτελεστεί ούτε μία φορά

Β. Η printf θα εκτελεστεί μία φορά και θα τυπώσει την τιμή -1. Μετά, το x θα αυξηθεί κατά 1.
Γ. Ατέρμονος βρόχος· η printf θα τυπώνεται συνέχεια για διαδοχικά x

42

Ερώτηση: Τι αποτέλεσμα θα έχει η εντολή break στο παρακάτω τμήμα προγράμματος;

```
for(i=0; i<1000; i++) {
    for(j=i+1; j<1000; j++) {
        ...
        if(x[i][j] < 0) break;
        ...
    }
}
```

Απάντηση:
Α. Η ροή της εκτέλεσης θα φύγει από το εσωτερικό for (αυτό με το μετρητή j) και θα μεταβιβαστεί στην αμέσως επόμενη εντολή που υπάρχει στο εξωτερικό for (αυτό με το μετρητή i)
Β. Η ροή της εκτέλεσης θα φύγει και από τα δύο for και θα πάει στην αμέσως επόμενη εντολή
Γ. Θα διακοπεί η τρέχουσα επανάληψη και το πρόγραμμα θα αρχίσει αμέσως την επόμενη επανάληψη (για την τρέχουσα τιμή του i και την επόμενη τιμή του j)

43

Ερώτηση: Τι δεν πάει καλά με τις παρακάτω εντολές και πώς διορθώνεται;

```
while( x > 0 );
{ x--; y++; z = x*y}
```

Απάντηση:
Α. Οι εντολές στο σώμα της while (αυτές που περικλείονται από άγκιστρα) πρέπει να γραφτούν σε διαφορετική σειρά η κάθε μία αλλιώς ο μεταγλωττιστής θα δώσει μήνυμα για συντακτικό σφάλμα
Β. Αν το x είναι θετικό όταν φτάσουμε στη while θα έχουμε ατέρμονο βρόχο. Πρέπει να σβήσουμε το ελληνικό ερωτηματικό στην πρώτη σειρά.
Γ. Αν το x είναι θετικό όταν φτάσουμε στη while θα έχουμε ατέρμονο βρόχο. Πρέπει να σβήσουμε το ελληνικό ερωτηματικό στην πρώτη σειρά και να προσθέσουμε ένα στο τέλος της δεύτερης αλλιώς θα έχουμε συντακτικό σφάλμα.

44

Ερώτηση: Τι πρέπει ή μπορεί να αλλάξει στον παρακάτω ορισμό ενός πίνακα;

```
int x[][] = {{5, 4}, {3, 2}, {1, 0}};
```

Απάντηση:
Α. Τίποτε δε χρειάζεται να αλλάξει, επειδή ο ορισμός των διαστάσεων γίνεται αυτόματα με την απόδοση αρχικών τιμών των στοιχείων.
Β. Απαιτείται ο ορισμός τουλάχιστον της δεύτερης διάστασης.
Γ. Πρέπει να δηλωθούν ρητά τα μεγέθη και των δύο διαστάσεων.

45

Ερώτηση: Γενικά, μπορούμε να ορίσουμε δείκτες που να δείχνουν σε...

Απάντηση:
Α. ... απλές μεταβλητές όπως int, float, char.
Β. ... απλές μεταβλητές όπως int, float, char, πίνακες διαφόρων τύπων και δομές δεδομένων.
Γ. ...απλές μεταβλητές όπως int, float, char, πίνακες διαφόρων τύπων, δομές δεδομένων, άλλους δείκτες και συναρτήσεις.

46

Ερώτηση: Έστω η γραμμική συνδεδεμένη λίστα list με στοιχεία του παρακάτω τύπου:

```
struct lll {
    float value;
    struct lll *next;
};
```

Τι εντολή μπορούμε να χρησιμοποιήσουμε για να διατρέξουμε τα στοιχεία της;

Απάντηση:

Α. `for ( i = 0; i < sizeof(list)/sizeof(list[0]); i++ )`
Β. `for ( p = list; p != NULL; p = p->next )`
Γ. `for ( i = 0; i < sizeof(list)/sizeof(list[0]); p = p->next )`

47

Ερώτηση: Αν το s είναι ένα αλφαριθμητικό, τι σημαίνει η παρακάτω εντολή;

```
if ( !s ) return 0;
```

Απάντηση:

Α. Αν το αλφαριθμητικό s είναι κενό να επιστραφεί η τιμή 0.
Β. Αν το αλφαριθμητικό s έχει μόνο χαρακτήρες κενού διαστήματος '' να επιστραφεί η τιμή 0.
Γ. Αν το αλφαριθμητικό s περιέχει τουλάχιστον ένα χαρακτήρα να επιστραφεί η τιμη 0.

48

Ερώτηση: Υπάρχει λάθος στην παρακάτω εντολή και ποιο;

```
int X[3] = {0, 1, 2, 3, 4};
```

Απάντηση:

Α. Όχι, ο μεταγλωττιστής αλλάζει αυτόματα το μέγεθος του πίνακα για να μπούνε όλα τα στοιχεία.
Β. Ναι, οι πίνακες δε μπορούν να αρχικοποιούνται ταυτόχρονα με τη δήλωσή τους.
Γ. Ναι, δίνονται περισσότερα στοιχεία από το δηλωμένο μέγεθος του πίνακα.

49

Ερώτηση: Τι σημαίνει η παρακάτω εντολή;

```
if(x == someFunction(y)) printf("Αποτέλεσμα: %f", x);
```

Απάντηση:

Α. Αν η μεταβλητή x είναι ίση με την τιμή που επιστρέφει η συνάρτηση someFunction για όρισμα ίσο με y τότε τύπωσε το x
Β. Το x θα πάρει τιμή ίση με αυτή που επιστρέφει η συνάρτηση someFunction για όρισμα ίσο με y. Αν αυτή είναι διαφορετική από μηδέν τότε τύπωσε το x.
Γ. Το x θα πάρει τιμή ίση με αυτή που επιστρέφει η συνάρτηση someFunction για όρισμα ίσο με y. Αν αυτή είναι ίση με μηδέν τότε τύπωσε το x.

50

Ερώτηση:Πώς μπορούμε να τυπώσουμε τα μέλη της μεταβλητής myFriend;

```
struct person {
    char *name;
    char *address;
    char *phone;
} myFriend = {"Lakis Makis", "Kozanis 15", "2461099999"};
```

Απάντηση:
Α. `printf("%s\n", myFriend);`
Β. `printf("%s, %s, %s\n",myFriend.name, myFriend.address, myFriend.phone);`
Γ. `printf("%20c, %20c, %20c\n",myFriend.name, myFriend.address, myFriend.phone);`

Β. Υποδείξεις

-
02. Σκεφτείτε ότι ο μετρητής (iter) θα πρέπει να πάρει τιμή που να διαψεύσει τη Συνθήκη Ελέγχου για να σταματήσει η επανάληψη.
03. Σκεφτείτε ότι η μεταβλητή n θα αυξάνεται μέχρι και την τελευταία φορά που θα ισχύει η Συνθήκη Ελέγχου.
04. Θυμηθείτε ότι το όνομα ενός πίνακα είναι επίσης και σταθερά δείκτη.
06. Αναλύστε τις εντολές που περιέχουν τον τελεστή αύξησης σε δύο εντολές ανάθεσης με τη σειρά που υποδηλώνει η θέση του τελεστή αύξησης.
07. Λάβετε υπ' όψιν ότι προφανώς υπολογίζεται η παράσταση 1/(x(x-1)(x-2)) άρα πρέπει πρώτα να ελεγχθεί αν ο παρονομαστής είναι μηδέν.
08. Παρατηρείστε ότι έχουμε ένθεση τελεστών επιλογής ? :. Δείτε πρώτα τι κάνει η εσωτερική παράσταση και μετά υπολογίστε την εξωτερική
09. Προσέξτε ότι η μία μεταβλητή ορίζεται ως δείκτης άρα περιέχει διευθύνσεις και το πέρασμά της είναι έγκυρο ως πέρασμα κατά διεύθυνση.
13. Σκεφτείτε αν ο έλεγχος της συνθήκης θα γίνει πριν ή μετά από την αύξηση της x
14. Παρατηρείστε ότι στη θέση της παράστασης μεταβολής (το τρίτο τμήμα του for) υπάρχει μια ολόκληρη εντολή ανάθεσης (κάτι το οποίο επιτρέπεται γενικά).
15. Το κλειδί για την απάντηση είναι στην εντολή ανάθεσης κάτω από το for που περιέχει τον τελεστή επιλογής ? :. Δείτε επίσης ότι και οι δύο δείκτες p και q περιλαμβάνονται στην παράσταση μεταβολής του for που εκτελείται μέχρι το q να συναντήσει τον χαρακτήρα τερματισμού.
16. Θυμηθείτε ότι η αρίθμηση των στοιχείων πινάκων αρχίζει από το μηδέν
17. Παρατηρείστε ότι στην εντολή υπολογισμού του d υπάρχουν τελεστές αύξησης ++ στα μέλη του q και λάβετε υπ' όψιν ότι ο τελεστής -> έχει υψηλότερη προτεραιότητα από τον ++.
-
19. Με βάση τη θέση του τελεστή αύξησης ++ βρείτε αν το y θα πάρει την αρχική ή αυξημένη τιμή του x και μετά παρατηρείστε ότι οι δείκτες παίρνουν ο ένας την τιμή του άλλου.
20. Θυμηθείτε ότι τα αλφαριθμητικά έχουν και ένα χαρακτήρα τερματισμού.
21. Παρατηρείστε ότι το όρισμα περνάει στη συνάρτηση κατά διεύθυνση.
-
23. Θυμηθείτε ότι όταν τα ορίσματα περνάνε κατά διεύθυνση τότε μπορούν να μεταβληθούν ενώ όταν περνάνε κατά τιμή μένουν αμετάβλητα επειδή η συνάρτηση δημιουργεί και επεξεργάζεται τοπικά αντίγραφα αυτών.
24. Θυμηθείτε ότι τα αλφαριθμητικά έχουν και ένα χαρακτήρα τερματισμού.
-
27. Αφού το αλφαριθμητικό s μπορεί να εννοηθεί και ως δείκτης, έχουμε πέρασμα κατά διεύθυνση. Τότε η παράμετρος εισόδου της συνάρτησης είναι δείκτης και το περιεχόμενό της ανακτάται με τη βοήθεια του τελεστή περιεχομένου.
-
30. Θυμηθείτε ότι όταν τα ορίσματα περνάνε κατά διεύθυνση τότε μπορούν να μεταβληθούν ενώ όταν περνάνε κατά τιμή μένουν αμετάβλητα επειδή η συνάρτηση δημιουργεί και επεξεργάζεται τοπικά αντίγραφα αυτών.

31.
32. Προσέξτε ότι πρόκειται για απλό πίνακα χαρακτήρων και όχι αλφαριθμητικό.
-
40. Διακρίνετε ανάμεσα στον τελεστή απόδοσης τιμής = και στον τελεστή ελέγχου ισότητας = =.
-
46. Προσέξτε ότι δεν έχουμε να κάνουμε με πίνακα κι έτσι δε μπορούμε να χρησιμοποιήσουμε αριθμοδείκτες.
47. Προσέξτε ότι ο χαρακτήρας κενού διαστήματος ' ' είναι και αυτός ένας χαρακτήρας όπως όλοι οι άλλοι. Να μην το συγχέετε με το αλφαριθμητικό που δεν έχει καθόλου χαρακτήρες!
-
49. Διακρίνετε ανάμεσα στον τελεστή απόδοσης τιμής = και στον τελεστή ελέγχου ισότητας = =.