**Πανεπιστήμιο Θεσσαλίας**

**Τμήμα Πληροφορικής**

**Ακ. Έτος 2014-2015 – Σεπτέμβριος 2015**

**Διδάσκων:** Πέτρος Λάμψας

**Εισαγωγή στον Προγραμματισμό**

**Ον/μο Εξεταζόμενου: Α.Μ.:**

**Εξάμηνο: Βαθμός:**

**ΟΜΑΔΑ Α**

**Μέρος Α – Ερωτήσεις Πολλαπλής Επιλογής (70 Μονάδες)**

**Διάρκεια εξέτασης:** 2 ώρες.

Βαθμολογείστε με άριστα το 100. Στα θέματα πολλαπλής επιλογής η ορθή απάντηση βαθμολογείται με 5 μονάδες, ενώ η εσφαλμένη απάντηση προκαλεί την αφαίρεση 1 μονάδας. Οι απαντήσεις δίνονται πάνω στα θέματα. Οι κόλες που θα σας μοιραστούν μπορούν να χρησιμοποιηθούν μόνο για πρόχειρο και πρέπει να επιστραφούν **υποχρεωτικά** με τα θέματα.

(1) Πόσα Χ θα εκτυπώσει η ακόλουθη πρόταση;

for (y=0; y<5; y++)

 for (x=5; x>0;x--)

 puts(“X”);

(α) 5

(β) 10

(γ) 25

(δ) 50

(2) Γιατί η εντολή for της C δεν αποτελεί ασφαλή βρόχο με καθορισμένο πλήθος επαναλήψεων;

(α) Η μεταβλητή του βρόχου μπορεί να μεταβληθεί μέσα στο σώμα του βρόχου

(β) Η μεταβλητή του βρόχου μπορεί να αρχικοποιηθεί με τιμή διάφορη του μηδενός

(γ) Η τιμή της μεταβλητής του βρόχου μπορεί να ελεγχθεί

(δ) Η μεταβλητή του βρόχου μπορεί να χρησιμοποιηθεί στο σώμα του βρόχου

(3) Ποιες από τις παρακάτω δομές ελέγχου της C είναι ένας βρόχος, που ελέγχει τη συνθήκη επανάληψης στην αρχή;

(α) do...while

(β) for

(γ) switch

(δ) while

(4) Τι θα εμφανίσει ο ακόλουθος κώδικας;

void main(void)

{

int i;

for(i=0; i<5; i++)

;

printf("%d", i);

}

(α) 01234

(β) 012345

(γ) Σφάλμα κατά τη μεταγλώττιση λόγω του ; στη γραμμή 5

(δ) 5

(5) Τι θα εμφανίσει στην οθόνη το παρακάτω απόσπασμα κώδικα;

int x = 5, y = 9, z;

z = x++ ; y = ++z - x++ ; x = y-- - --z ;

printf("%d %d %d\n", x, y, z);

(α) -7 -1 6

(β) -7 -2 5

(γ) -5 -1 5

(δ) -6 0 6

(6) Έστω το ακόλουθο τμήμα κώδικα:

int x[5];

x[5] = 12;

Υποθέτοντας ότι αυτό το τμήμα κώδικα υπάρχει σε έναν κατά τα άλλα σωστό κώδικα, τι αποτέλεσμα θα φέρει η εκτέλεση της δεύτερης εντολής;

(α) Την ανάθεση στο πέμπτο στοιχείο του πίνακα x της τιμής 12

(β) Την ανάθεση στο έκτο στοιχείο του πίνακα x της τιμής 12

(γ) Πιθανό σφάλμα κατά την εκτέλεση του προγράμματος αφού το στοιχείο x[5] είναι εκτός των ορίων του πίνακα

(δ) Σφάλμα κατά τη μεταγλώττιση του προγράμματος λόγω εσφαλμένης δήλωσης των ορίων του πίνακα

(7) Ας υποθέσουμε ότι η μεταβλητή x αναπαριστά έναν πίνακα ακεραίων και ότι έχει εκτελεστεί ο ακόλουθος κώδικας:

for(i=0;i<5;i++)

 x[i] = i\*i;

Έστω ότι το στοιχείο x[0] αποθηκεύεται στη διεύθυνση μνήμης 4500 και ότι κάθε ακέραιος καταλαμβάνει στη μνήμη 4 bytes. Ποια είναι η τιμή της κάθε μιας από τις ακόλουθες εκφράσεις;

(α) x (β) &x[0] (γ) \*x

(δ) x[1] (ε) &x[1] (στ) x+2

(ζ) \*(x+2) (η) \*(&x[2] +1)

(8) Έστω η μεταβλητή ptr και η μεταβλητή a που βρίσκεται στη διεύθυνση μνήμης 6000. Υποθέτουμε μέγεθος τύπου int 4 bytes Ποια θα είναι η τιμή της ptr μετά την εκτέλεση των επόμενων εντολών;

int a, \*ptr;

ptr=&a;

ptr=ptr+3;

(α) Δεν μπορούμε να γνωρίζουμε

(β) 6003

(γ) 6006

(δ) 6012

(9) Πόσες φορές θα εκτελεστεί η πρόταση που βρίσκεται μέσα στο πλαίσιο;

printf(“Αρχή”);

for (i=1; i<=5; i++) {

 printf(“\*\*\*”);

 for (j=1; j<=3; j++)

 printf(“%d\n”, j);

 for (k=1; k<=8; k++) {

 printf(“Νίκος”);

 for (l=1; l<=4; l++)

printf(“%d\n”, l);

 }

 printf(“---”);

}

(α) 8

(β) 160

(γ) 20

(δ) 40

(10) Ποιά τιμή θα εμφανίσει το ακόλουθο απόσπασμα κώδικα;

int temp[5]={10, 19, 23, 8, 9};

int \*p;

p=temp;

printf(“%d”,\*(p+3));

(α) 23

(β) 8

(γ) 13

(δ) 26

(11) Τι θα εμφανίσει στην οθόνη το ακόλουθο πρόγραμμα;

#include <stdio.h>

void draw\_box(int, int);

main() {

 draw\_box(8,10);

 return 0;

}

void draw\_box(int row, int column) {

 int col, r;

 for(r=row;row>0;row--) {

 for(col=column;col>0;col--)

 printf(“X”);

 printf(“\n”);

 }

}

(α) Θα εμφανίσει row\*column X σε μια σειρά

(β) Ένα πίνακα διαστάσεων 10x8 που όλες οι τιμές του θα είναι Χ

(γ) Ένα πίνακα διαστάσεων 10x10 που όλες οι τιμές του θα είναι Χ

(δ) Ένα πίνακα διαστάσεων 8x10 που όλες οι τιμές του θα είναι Χ

(12) Τι θα εμφανίσει στην οθόνη το παρακάτω κομμάτι κώδικα αν ο χρήστης πληκτρολογήσει την εξής ακολουθία χαρακτήρων: ‘B’ ‘F’ ‘Q’ ‘P’;

for( ; ; ) {

 printf("Pass, Miss, Fail, Quit?\n");

 do {

 printf ("Enter your selection: ");

 ch = getch();

 printf(“%c ”, ch);

 } while(ch!='P' && ch!='M' && ch!='F' && ch!='Q');

 if(ch == 'Q') break;

 printf(“Finally!”);

}

|  |  |
| --- | --- |
| (α) Enter your selection: B B Enter your selection: F F Finally! Enter your selection: Q Q Enter your selection: P P Finally! | (β) Enter your selection: B B Enter your selection: F F Finally! Enter your selection: Q Q |
| (γ) Enter your selection: B B Enter your selection: F F Finally! | (δ) Ατέρμον βρόχος |

(13) Τι θα εμφανίσει στην οθόνη το παρακάτω πρόγραμμα:

#include <stdio.h>

int i;

float sum=0.0;

struct employee {

 char name[20];

 float salary;

};

struct employee a[3]={{"Black",1246.0},

 {"Smith",1154.0},

 {"Peterson",1432.0}};

main() {

 for (i=2; i>=0; i--) {

 printf("%f->%s\n", a[i].salary, a[i].name);

 sum = sum + a[i].salary;

 }

 printf("%f\n", sum);

}

|  |  |
| --- | --- |
| (α) 1154.0->Smith 1432.0->Peterson 2586.0 | (β) 1154.0->Smith 1246.0->Black 2400.0 |
| (γ) Peterson->1432.0 Smith->1154.0 Black->1246.0 3832.0 | (δ) 1432.0->Peterson 1154.0->Smith 1246.0->Black 3832.0 |

(14) Τι εκτυπώνει το παρακάτω απόσπασμα κώδικα;

#include <stdio.h>

int main(void){

 int i;

 for (i=1; i<101; i++){

 if(!(i%2)) continue;

 printf(“%d”, i);

 }

 return 0;

}

(α) Όλους τους αριθμούς από το 1 μέχρι το 100

(β) Μόνο τους άρτιους αριθμούς μεταξύ 1 και 100

(γ) Μόνο τους περιττούς αριθμούς μεταξύ 1 και 100

(δ) Μόνο τα υπόλοιπα των αριθμών από το 1 μέχρι το 100

**ΜΕΡΟΣ Β – Ανάπτυξη Κώδικα (30 Μονάδες)**

(1) Η παρακάτω συνάρτηση *f* χαρακτηρίζεται, γενικά στον προγραμματισμό, ως αναδρομική, διότι είναι μια συνάρτηση που καλεί τον εαυτό της. Κατά τα άλλα, υπακούει σε όλους τους κανόνες που γνωρίζετε για τις συναρτήσεις. Στο παρακάτω απόσπασμα κώδικα, προσθέστε στη γραμμή 2 κατάλληλο κώδικα ώστε όταν το πρόγραμμα αυτό εκτελείται να επιστρέφει την ακολουθία τιμών 1 2 3 4 5 6 7 8 9.

1 void f(int x) {

2

3 f(x-1);

4 printf("%d ", x);

5 }

6

7 int main(void) {

8 f(9);

9 printf("\n");

10 return 0;

11}

(2) Στο πλαίσιο που ακολουθεί γράψτε πρόγραμμα που να υπολογίζει το *n!* (Σημείωση: . Αν *n=0* τότε *n!=1*).

(3) Στο πλαίσιο που ακολουθεί γράψτε ένα πρόγραμμα που να εκτυπώνει τους αριθμούς: 1 3 6 9 12 15 18 21.