

Γραπτή Εξέταση στο Μάθημα  
Εισαγωγή στο Διαδικαστικό Προγραμματισμό

**Στοιχεία Φοιτητή**

Όνοματεπώνυμο: \_\_\_\_\_ Έτος: \_\_\_\_\_ Αριθμός  
Μητρώου: \_\_\_\_\_

**Διδάσκων:** Χ. Μακρής, Ε. Πολυχρονόπουλος

1°	2°	3°	4°	5°	6°	7°	8°	9°	10°

1) Υλοποιήστε μία συνάρτηση η οποία να δέχεται σαν είσοδο μία συμβολοσειρά υπολογίζει το πλήθος των συμβόλων της και την εκτυπώνει ανάστροφα.

2)

(i) να γραφεί συνάρτηση η οποία να παίρνει σαν όρισμα ένα πίνακα ακεραίων δύο διαστάσεων ( $n \times n$ ) και επιστρέφει ως τιμή το άθροισμα των στοιχείων της διαγωνίου του,  
(ii) να γραφεί συνάρτηση η οποία να παίρνει σαν όρισμα ένα πίνακα ακεραίων δύο διαστάσεων ( $n \times n$ ) και εκτυπώνει τον μεγαλύτερο αριθμό από κάθε γραμμή του πίνακα,  
(iii) να γραφεί κυρίως πρόγραμμα το οποίο να δηλώνει ένα πίνακα ακεραίων  $100 \times 100$ , διαβάξει τις τιμές των στοιχείων του και καλεί τις συναρτήσεις που ορίστηκαν στα ερωτήματα (i) και (ii).

3) Δώστε μία αναδρομική και μία επαναληπτική συνάρτηση για τον υπολογισμό της παρακάτω παράστασης:

$$S(n) = 1^3 + 2^3 + 3^3 + \dots + n^3.$$

4) Υλοποιήστε μία συνάρτηση κωδικοποίησης η οποία θα δέχεται σαν όρισμα ένα αλφαριθμητικό και θα αντικαθιστά κάθε χαρακτήρα του με τον χαρακτήρα που προκύπτει αν μετατοπίσουμε κάθε χαρακτήρα δύο θέσεις προς τα δεξιά στο αλφάβητο. Αν ξεπεράσουμε το  $z$  ή το  $Z$  η μετατόπιση θα γίνει από την αρχή του αλφαβήτου.

5) Υλοποιήστε συνάρτηση που δέχεται σαν όρισμα δύο αλφαριθμητικά και επιστρέφει σαν αποτέλεσμα την συνένωσή τους.

6) Υλοποιήστε μία συνάρτηση που παίρνει σαν όρισμα μια συμβολοσειρά  $p$  και μία συμβολοσειρά  $T$ , και εντοπίζει όλες τις εμφανίσεις της  $p$  μέσα στο  $T$ , χωρίς να λαμβάνονται υπόψιν οι διαφορές στη μορφή κεφαλαίων και μικρών (υπόδειξη: προτείνεται η χρήση της συνάρτησης `int tolower(int c)` της βιβλιοθήκης `<ctype.h>` που μετατρέπει το όρισμά της σε πεζό).

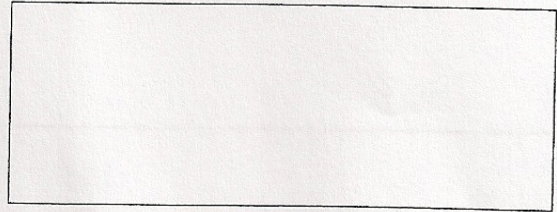
7) Το ακόλουθο πρόγραμμα υπολογίζει το πλήθος των πρώτων αριθμών από το 2 έως το 49. Εντοπίστε το λογικό σφάλμα που υπάρχει στον υπολογισμό και διορθώστε το.

```
#include <stdio.h>
main()
{ int i, p=0, c=0, j;
  for (i=2; i<50; i++)
  {
    for (j=2; j<i; j++)
      If (i%j==0) c++;
```

```

    If (c==0) p++;
}
printf("Plithos = %d\n", p);
}

```

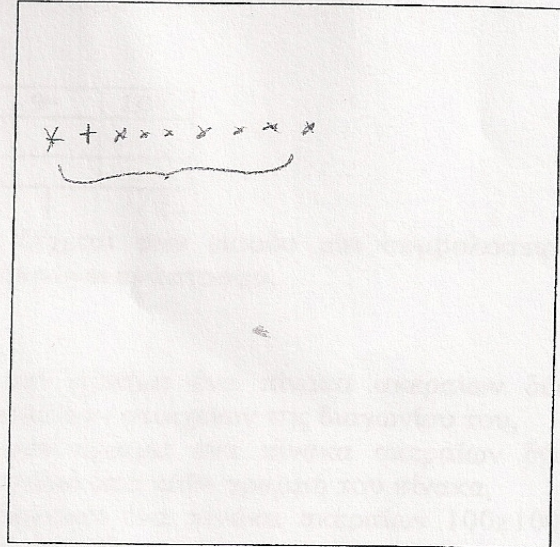


8) Τι έξοδο θα δώσει ο παρακάτω κώδικας όταν εκτελεστεί;

```

#include <stdio.h>
int main()
{
    int b=1, a=1;
    do {
        b++;
        if (a>3) continue;
        printf("*");
        while (a*b<3)
        {
            b++;
            printf("+");
        }
    } while (b<10);
}

```

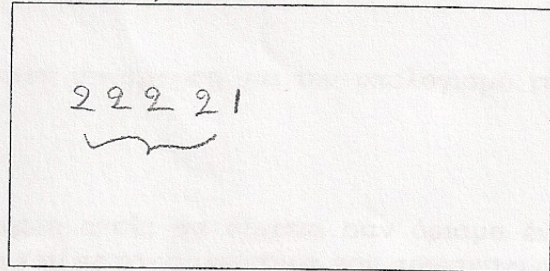


9) Τι έξοδο θα δώσει ο παρακάτω κώδικας όταν εκτελεστεί;

```

#include <stdio.h>
int main()
{
    int values[]={2,1,2,1,2}, *p=values, *q=p+1;
    printf("%d%d%d%d%d\n", values[2], *p, *(q+1),
    *q+1, q-&values[0]);
}

```



10) Τι έξοδο θα δώσει ο παρακάτω κώδικας όταν εκτελεστεί;

```

#include <stdio.h>
int x=10;
void out1();
void out2();
int main()
{
    int i;
    for (i=1; i<=2; i++) out1();
    for (i=1; i<=2; i++) out2();
}
void out1()
{
    static int x=4;
    printf("%d\n",x++);
}
void out2()
{
    int x=15;
    printf("%d\n", x++);
}

```

