

ΠΡΟΓΡΑΜΜΑΤΑ ΣΕ C

Γράψτε σε γλώσσα προγραμματισμού C τη συνάρτηση:

```
int b_to_d(int dyad[16])
```

που δέχεται ως είσοδο έναν θετικό ακέραιο δυαδικό αριθμό με τη μορφή πίνακα δυαδικών ψηφίων και επιστρέφει τον ισοδύναμο δεκαδικό αριθμό. Θεωρήστε ότι το `dyad[15]` είναι το περισσότερο σημαντικό ψηφίο του δυαδικού αριθμού και το `dyad[0]` το λιγότερο σημαντικό.

ΠΡΟΓΡΑΜΜΑ

```
int b_to_d(int dyad[16])
{
    int i, g = 1, s = 0;

    for (i = 0; i < 16; ++i)
    {
        s = s + dyad[i] * g;
        g = 2 * g;
    }
    return(s);
}
```

Γράψτε σε γλώσσα προγραμματισμού C ένα πρόγραμμα που θα διαβάζει από το πληκτρολόγιο έναν πίνακα θετικών ακεραίων αριθμών δύο διαστάσεων ($N \times M$) και στη συνέχεια θα αντιμετωπίζει την πρώτη με την τελευταία στήλη του. Στη συνέχεια το πρόγραμμα θα εκτυπώνει το νέο πίνακα. Εφαρμόστε αμυντικό προγραμματισμό στην ανάγνωση των στοιχείων του πίνακα, ώστε να είναι θετικοί αριθμοί.

ΠΡΟΓΡΑΜΜΑ

```
#include <stdio.h>
#define N 10 //Πλήθος γραμμών του πίνακα = 10
#define M 5 //Πλήθος στηλών του πίνακα = 5
main()
{
```

```

//Αρχή προγράμματος
int temp, i, j, x[N][M];
//Διάβασμα του πίνακα με αμυντικό προγραμματισμό
for (i = 0; i < N; i++)
    for (j = 0; j < M; j++)
        do {
            printf("Dose to stoixeio x[%d][%d]: ", i, j);
            scanf("%d",&x[i][j]);
        }
        while (x[i][j] <= 0);
//Εναλλαγή πρώτης και τελευταίας στήλης
for (i = 0; i < N; i++)
{
    temp = x[i][0];
    x[i][0] = x[i][M-1];
    x[i][M-1] = temp;
}
printf("\nPinakas Me enallagmenes stiles\n");
for (i = 0; i < N; i++)
{
    for (j = 0; j < M; j++)
        printf("%d\t", x[i][j]);
    printf("\n"); //Αλλαγή γραμμής
}
}

```

Γράψτε πρόγραμμα σε γλώσσα προγραμματισμού C, το οποίο θα διαβάζει 10 ακέραιες τιμές, θα τις αποθηκεύει σε έναν μονοδιάστατο πίνακα ακεραίων και θα εμφανίζει στην οθόνη τα δύο μεγαλύτερα στοιχεία του πίνακα.

ΠΡΟΓΡΑΜΜΑ

```

#include <stdio.h>

main()
{
    int i, max1, max2, a[10];

    for (i=0; i<10; i++)

```

```

{
    printf("\nΔώστε το %do στοιχείο του πίνακα: ", i + 1);
    scanf("%d", &a[i]);
}

max1 = a[0]>a[1] ? a[0] : a[1];
// Το max1 παίρνει τη μεγαλύτερη τιμή από τα δύο πρώτα
στοιχεία
max2 = a[0] + a[1] - max1;
// Το max2 παίρνει την άλλη τιμή από τα δύο πρώτα στοιχεία

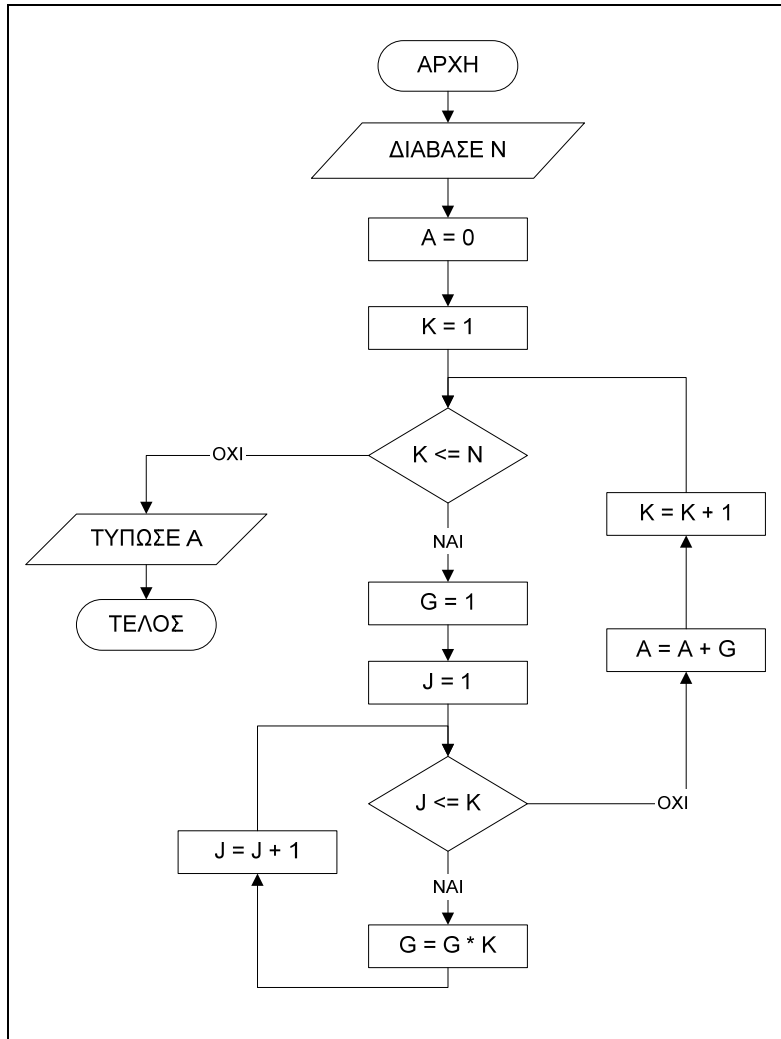
for (i=2; i<10; i++)
    if (a[i]>max1)
    {
        max2=max1;
        max1=a[i];
    }
    else if (a[i]>max2)
        max2=a[i];

printf("\nΤα δύο μεγαλύτερα στοιχεία του πίνακα είναι: %d
και %d", max1, max2);
} /* τέλος προγράμματος */

```

ΕΡΓΑΣΙΑ 1

Δίνεται το παρακάτω Διάγραμμα Ροής Προγράμματος - ΔΡΠ (flowchart), όπου το N είναι θετικός ακέραιος αριθμός μεγαλύτερος του μηδενός ($N > 0$).



α) Τι εκτυπώνει ο αλγόριθμος αν στην είσοδο δώσουμε την τιμή:

i) $N = 1$,

ii) $N = 2$, και

iii) $N = 3$.

β) Γράψτε πρόγραμμα σε γλώσσα προγραμματισμού C που να έχει την ίδια λειτουργία με το ΔΡΠ, εφαρμόζοντας επιπρόσθετα και αμυντικό προγραμματισμό στην εισαγωγή του N , έτσι ώστε να είναι ακέραιος αριθμός μεγαλύτερος του μηδενός ($N > 0$).

ΕΡΓΑΣΙΑ 2

Τι θα εμφανίσει το παρακάτω Πρόγραμμα μετά την εκτέλεσή του; Αιτιολογήστε την απάντησή σας.

```
#include <stdio.h>
main()
{int i, a=2,b=3, c[2]={7,8};
 printf("%d\n",a==b);
 printf("%d\n",a==b-1);
 a=b;
```

```
printf(“%d %d\n”,a,b);
c[a==b]=12;
printf(“%d %d\n”,c[0],c[1]);
}
```

ΕΡΓΑΣΙΑ 3

Να γραφεί αλγόριθμος σε ψευδοκώδικα και πρόγραμμα σε γλώσσα C, το οποίο να:

- b1) διαβάζει τα στοιχεία ενός πίνακα ακεραίων εκατό θέσεων
- b2) ταξινομεί τα στοιχεία του πίνακα χρησιμοποιώντας τον αλγόριθμο ταξινόμησης φυσαλίδας (Bubble Sort)
- b3) εμφανίζει τα στοιχεία του ταξινομημένου πίνακα

Παράδειγμα εκτέλεσης του προγράμματος:

Enter the 0 element of the table: 20
 Enter the 1 element of the table: 40
 Enter the 2 element of the table: 60
 Enter the 3 element of the table: 80
 Enter the 4 element of the table: 100
 Enter the 5 element of the table: 10
 Enter the 6 element of the table: 30
 Enter the 7 element of the table: 50
 Enter the 8 element of the table: 70
 Enter the 9 element of the table: 90
(κ.λ.π.)

Sorted Table: 10 20 30 40 50 60 70 80 90 100 κ.λ.π.

ΕΡΓΑΣΙΑ 4

Τι θα εμφανίσει στην οθόνη το παρακάτω πρόγραμμα όταν εκτελεστεί; Αιτιολογήστε την απάντησή σας.

```
#include <stdio.h>
main()
{
    int i, j=10;
    for (i=0; i<=20; ++i)
        if (i%3==1)
        {
            printf ("%d ", i);
            j--;
        }
    printf ("\nj=%d\n", j);
}
```

```
}
```

ΕΡΓΑΣΙΑ 5

Τι θα εμφανίσει στην οθόνη το παρακάτω πρόγραμμα όταν εκτελεστεί; Αιτιολογήστε την απάντησή σας.

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

void stars(int* p, int sizeP)
{
    /* δήλωση μεταβλητών */
    int i=0,j,error=0;//θέτοντας αρχικά στο error την τιμή 0 υποθέτουμε ότι όλα τα
στοιχεία του πίνακα είναι έγκυρα.

    /* έλεγχος τιμών των στοιχείων του πίνακα p */
    for (i=0;i<sizeP;i++)
    {
        if (*(p+i) < 0 || *(p+i) > 20)//Ο συμβολισμός *(p+i) αναφέρεται στο
στοιχείο p[i]
        {
            printf("element %d is out of limit\n",i+1);
            error=1;
            break;
        }
    }

    if (error==1)/*δεν γίνεται εκτύπωση άστρων */
        printf("Den ginetai kamia ektyposi\n");

    else /*εκτύπωση άστρων */
    {
        for (i=0;i<sizeP;i++)
        {
            for(j=1;j<=*(p+i);j++)//Η επανάληψη αυτή τυπώνει 1 αστεράκι
κάθε φορά που εκτελείται και επαναλαμβάνεται τόσες φορές όσο είναι και ο αριθμός
στη θέση i του πίνακα
                printf("*");

            printf("\n");//αλλάζουμε γραμμή για το επόμενο στοιχείο του
πίνακα
        }
    }
}
```

```

    }
}

void main()
{
    int size,i,*x;

    printf("dose megethos pinaka\n");
    scanf("%d",&size);

    x=(int*)malloc(size*sizeof(int));

    for (i=0;i<size;i++)
    {
        printf("dose %d stoixei\n",i+1);
        scanf("%d",(x+i));//Επειδή το x+i είναι διεύθυνση δεν χρειάζεται να
        χρησιμοποιήσουμε το σύμβολο &
    }

    stars(x,size);
}

```

ΕΡΓΑΣΙΑ 6

Ποια η λειτουργία του ακόλουθου προγράμματος; Το πρόγραμμα παρουσιάζει σφάλμα κατά την εκτέλεση του; Αν ναι ποιο ακριβώς είναι το σφάλμα αυτό και σε ποιες από τις ακόλουθες περιπτώσεις εκτέλεσης εμφανίζεται; Επίσης εφόσον υπάρχει σφάλμα να το διορθώσετε.

Περιπτώσεις εκτέλεσης:

1^η εκτέλεση 3,4,5,6,0

2^η εκτέλεση 7,7,2,1,3

3^η εκτέλεση 3,6,9,12,13

4^η εκτέλεση 25,1,89,2,3

```

#include <stdio.h>
void main()
{
    int i, c=0,x,sum=0;

    for (i=0;i<5;i++)
    {
        scanf("%d",&x);

        if (x%3==0)
            continue;

        if (x%5==0)
            break;

        sum+=x;
        c++;
    }
}

```

```

    }

    printf("Average = %.2fn", (float)sum/c);
}

```

ΕΡΓΑΣΙΑ 7

Ποια η λειτουργία του ακόλουθου προγράμματος; Το πρόγραμμα παρουσιάζει σφάλμα κατά την εκτέλεση του; Αν ναι να το διορθώσετε

```

#include <stdio.h>

void main()
{
    int i, p=0, c=0, j;

    for (i=1; i<50; i++)
    {
        for (j=2; j<i; j++)
            if (i%j==0)
                c++;

        if (c<2)
            p++;
    }

    printf("Plithos = %d\n", p);
}

```

ΕΡΓΑΣΙΑ 8

Το ακόλουθο πρόγραμμα θέλουμε να τυπώνει όλους τους άρτιους αριθμούς από το 1-10 και υπολογίζει το πλήθος τους. Αν παρουσιάζει λογικά/συντακτικά σφάλματα τότε να τα διορθώσετε ώστε να λειτουργεί σωστά.

```

#include <stdio.h>

void main()
{
    int c=0, j;
    float i;

    for (i=10; i>=0; i--)
    {
        if (i%2=0)
            printf("%d\n", i);
            c++;
    }

    printf("Plithos Artion= %d\n", c);
}

```



```
}
```

ΕΡΓΑΣΙΑ 9

Τι ακριβώς υπολογίζει το ακόλουθο πρόγραμμα; Δοκιμάστε να το εκτελέσετε με δικά σας δεδομένα και βρείτε ποιο γνωστό αλγόριθμο υλοποιεί; Να προσθέσετε ένα επιπλέον περιορισμό ώστε το πρόγραμμα να δέχεται μόνο θετικούς ακεραίους (αμυντικός προγραμματισμός)

```
#include <stdio.h>

void main()
{
    int sum=0,x,y;

    printf("Enter 2 positive integers \n");
    scanf("%d%d", &x, &y);

    while (y)
    {
        if (y%2==1)
            sum+=x;

        x*=2;
        y/=2;
    }

    printf("Result= %d\n", sum);
}
```

ΕΡΓΑΣΙΑ 10

Χωρίς να εκτελέσετε το πρόγραμμα να βρείτε ποιες τιμές τυπώνει. Στη συνέχεια να το ξαναγράψετε αντικαθιστώντας όλες τις εντολές for με εντολές while.

```
#include <stdio.h>
void main()
{
    int sum=0,j=0,i=3;

    for (;i<=9;i+=3)
        for (j=i; j>=0;j-=3)
            for(;;)
            {
                sum+=i*j;

                if (sum>80)
                {
                    printf("Result= %d\n",sum);
                    break;
                }
            }
}
```

```
}  
}  
}
```

ΕΡΓΑΣΙΑ 11

Να δηλώσετε ένα πίνακα ακεραίων 3X3 και να τον διαβάσετε με τυχαίες τιμές. Στη συνέχεια να υπολογίσετε το άθροισμα των στοιχείων του μέχρι το πρώτο αρνητικό στοιχείο (αν υπάρχει τέτοιο). Εκεί να τερματίζονται και οι 2 εντολές for με τις οποίες διαχειριζόμαστε τον πίνακα με κατάλληλη χρήση της εντολής break και το πρόγραμμα να τυπώνει το άθροισμα των στοιχείων που έχει υπολογίσει μέχρι εκείνη τη στιγμή. Αν δεν δοθεί αρνητικό στοιχείο στον πίνακα τότε να υπολογιστεί το άθροισμα όλων των στοιχείων του πίνακα

ΕΡΓΑΣΙΑ 12

Σε μια κινηματογραφική αίθουσα υπάρχουν συνολικά 100 θέσεις. Για να κάνουν κρατήσεις θέσεων οι θεατές επικοινωνούν με την έκδοση εισιτηρίων και ενημερώνονται για αν υπάρχουν κενές θέσεις στην αίθουσα. Να γραφεί πρόγραμμα στη γλώσσα C που να διαβάζει τον αριθμό των θέσεων που θέλει να κλείσει κάθε φορά ένας θεατής και αν υπάρχουν διαθέσιμες θέσεις τότε να εμφανίζεται ο αριθμός των θέσεων που παραμένουν κενές μετά την κράτηση, ο συνολικός αριθμός κρατήσεων μέχρι εκείνη τη στιγμή και οι θέσεις της τελευταίας κράτησης. Σε περίπτωση που ο αριθμός των θέσεων που θέλει κάποιος θεατής είναι μεγαλύτερος από τις διαθέσιμες θέσεις τότε να εμφανίζεται ο μέγιστος αριθμός θέσεων στις οποίες μπορεί να κάνει κράτηση και να τον ρωτά αν συμφωνεί. Αν ο θεατής απαντήσει θετικά τότε να γίνεται η κράτηση και το πρόγραμμα να τερματίζεται εμφανίζοντας το πλήθος κρατήσεων που έγιναν και το μήνυμα ότι η αίθουσα είναι πλήρης. Σε διαφορετική περίπτωση το πρόγραμμα να τερματίζεται εμφανίζοντας το πλήθος κρατήσεων που έγιναν και το μήνυμα ότι η αίθουσα δεν είναι πλήρης καθώς και τον αριθμό των θέσεων που θα παραμείνουν κενές