

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών

Π.Μ.Σ. Πληροφορική & Εφαρμογές

Προγραμματισμός Υπολογιστών

Εργασία Δεκεμβρίου-Ιανουαρίου 2020

A.

Να γραφεί πρόγραμμα στη γλώσσα C, το οποίο θα επιτελεί τα ακόλουθα:

(α) Μέσω επαναληπτικής πρότασης **do-while** θα λαμβάνει από το πληκτρολόγιο τις τιμές ενός πίνακα τεσσάρων αλφαριθμητικών **a_char[4][21]**. Η ανάγνωση των αλφαριθμητικών θα γίνεται με χρήση της **gets()**. Ακολούθως θα εμφανίζεται ο προκύπτων πίνακας στην οθόνη με χρήση της επαναληπτικής πρότασης **for**.

(β) Θα ελέγχει όλα τα αλφαριθμητικά του πίνακα αναζητώντας τον χαρακτήρα **"J"**. Σε κάθε επιτυχή ανεύρεση του χαρακτήρα θα εμφανίζονται στην οθόνη ο αριθμός του αλφαριθμητικού, καθώς και η θέση μέσα σε αυτό στην οποία βρέθηκε ο χαρακτήρας.

(γ) Θα λαμβάνει από το πληκτρολόγιο τις τιμές ενός πίνακα ακεραίων αριθμών **a_int[4][4]** και θα εμφανίζει τον προκύπτοντα πίνακα στην οθόνη.

(δ) Για την πρώτη και την τελευταία στήλη του πίνακα **a_int** θα υπολογίζει το γινόμενο των στοιχείων της στήλης, οι τιμές των οποίων δεν ανήκουν στο διάστημα $[-6,5]$. Οι τιμές των προκύπτοντων γινομένων θα εμφανίζονται στην οθόνη.

ΛΥΣΗ

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

```
#define N 4
#define LOWER -6
#define UPPER 5
```

```
int main()
```

```

{
    int a_int[N][N],i,j,column_prod_0,column_prod_3;
    char a_char[N][21];
    //-----
    // (i)
    i=1;
    do
    {
        printf("\n\nGive string no %d: ",i);
        gets(a_char[i-1]);
        i++;
    } while (i<5);
    printf("\n\nArray a_char:");
    for (i=0;i<N;i++) printf("\n\ta_char[%d] = %s\n",i,a_char[i]);
    //-----
    // (ii)
    for (i=0;i<N;i++)
        for (j=0;j<strlen(a_char[i]);j++)
        {
            if (a_char[i][j]=='J') printf("\n'J' appears at place no %d of string no %d",j+1,i+1);
        }
    //-----
    // (iii)
    printf("\n\n");
    for (i=0;i<N;i++)
        for (j=0;j<N;j++)
        {
            printf("\na_int[%d][%d]: ",i+1,j+1);
            scanf("%d",&a_int[i][j]);
        }
    printf("\n\nArray of integers:");
    for (i=0;i<N;i++)
    {
        printf("\n");
        for (j=0;j<N;j++) printf("\t%d",a_int[i][j]);
    }
    //-----
    // (iv)
    column_prod_0=1;
    column_prod_3=1;
    for (i=0;i<N;i++)
    {
        if ((a_int[i][0]<LOWER) || (a_int[i][0]>UPPER)) column_prod_0=column_prod_0*a_int[i][0];
        if ((a_int[i][3]<LOWER) || (a_int[i][3]>UPPER)) column_prod_3=column_prod_3*a_int[i][3];
    }
    printf("\n\tProduct of first column's values: %d",column_prod_0);
    printf("\n\tProduct of last column's values: %d",column_prod_3);
    return 0;
}

```

```

Give string no 1:  John

Give string no 2:  Radio DeeJay

Give string no 3:  Don't tell me Jokes

Give string no 4:  Temp

Array a_char:
    a_char[0] = John
    a_char[1] = Radio DeeJay
    a_char[2] = Don't tell me Jokes
    a_char[3] = Temp

'J' appears at place no 1 of string no 1
'J' appears at place no 10 of string no 2
'J' appears at place no 15 of string no 3

a_int[1][1]:  -8
a_int[1][2]:  2
a_int[1][3]:  3
a_int[1][4]:  6
a_int[2][1]:  6
a_int[2][2]:  9
a_int[2][3]:  7
a_int[2][4]:  6
a_int[3][1]:  2
a_int[3][2]:  0
a_int[3][3]:  3
a_int[3][4]: -4
a_int[4][1]:  7
a_int[4][2]:  7
a_int[4][3]:  5
a_int[4][4]: 10

Array of integers:
    -8      2      3      6
     6      9      7      6
     2      0      3     -4
     7      7      5     10
    Product of first column's values: -336
    Product of last column's values: 360

```

B.

Να γραφεί πρόγραμμα στη γλώσσα C, το οποίο θα επιτελεί τα ακόλουθα:

(α) Θα λαμβάνει από το πληκτρολόγιο τις τιμές ενός πίνακα αριθμών υποδιαστολής μονής ακρίβειας `b_float[2][3][2]`.

(β) Θα υπολογίζει το άθροισμα των στοιχείων του πίνακα `b_float`, οι τιμές των οποίων ανήκουν στο διάστημα [3,13]. Η τιμή του αθροίσματος θα εμφανίζεται στην οθόνη.

(γ) Μέσω επαναληπτικής πρότασης `while` θα λαμβάνει από το πληκτρολόγιο τις τιμές ενός πίνακα τριών αλφαριθμητικών `b_char[3][21]`. Η ανάγνωση των αλφαριθμητικών θα γίνεται με χρήση της `scanf()`. Ακολούθως θα εμφανίζεται ο προκύπτων πίνακας στην οθόνη με χρήση της επαναληπτικής πρότασης `for`.

(δ) Για το αλφαριθμητικό με το μέγιστο μήκος θα αναζητείται ο χαρακτήρας "`r`". Σε κάθε επιτυχή ανεύρεση του χαρακτήρα, θα εμφανίζεται στην οθόνη η θέση μέσα στο αλφαριθμητικό στην οποία βρέθηκε ο χαρακτήρας.

ΛΥΣΗ

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

#define LOWER 3
#define UPPER 13

int main()
{
    int i,j,k,string_max;
    float b_float[2][3][2],sum;
    char b_char[3][21];
    //-----
    // (i)
    printf("\n\n");
    for (i=0;i<2;i++)
        for (j=0;j<3;j++)
            for (k=0;k<2;k++)
            {
                printf("\nb_float[%d][%d][%d]: ",i+1,j+1,k+1);
                scanf("%f",&b_float[i][j][k]);
            }
    printf("\n\nArray of real numbers:");
    //-----
    // (ii)
```

```

sum=0.0;
for (i=0;i<2;i++)
    for (j=0;j<3;j++)
        for (k=0;k<2;k++)
        {
            if ((b_float[i][j][k]>=LOWER) && (b_float[i][j][k]<=UPPER)) sum=sum+b_float[i][j][k];
        }
printf("\n\tSum of b_float elements: %f",sum);
//-----
// (iii)
i=0;
while (i<3)
{
    printf("\n\nGive string no %d: ",i+1);
    scanf("%s",b_char[i]);
    i++;
}
printf("\n\nArray b_char:");
for (i=0;i<3;i++) printf("\n\tb_char[%d] = %s\n",i,b_char[i]);
//-----
// (iv)
string_max=0;
if (strlen(b_char[1])>strlen(b_char[0]))
{
    if (strlen(b_char[1])>strlen(b_char[2])) string_max=1;
    else string_max=2;
}
else if (strlen(b_char[2])>strlen(b_char[0])) string_max=2;
/* Δε χρειάζεται άλλος έλεγχος καθώς εάν δεν ισχύσουν οι ανωτέρω συνθήκες, ισχύει η
ανάθεση string_max=0 */
printf("\n\nThe string with the maximum number of characters is no %d",string_max+1);
for (j=0;j<strlen(b_char[string_max]);j++)
{
    if (b_char[string_max][j]=='r') printf("\n'r' was found in the %d place",j+1);
}
return 0;
}

```

```
b_float[1][1][1]: 5
b_float[1][1][2]: 2
b_float[1][2][1]: 8
b_float[1][2][2]: 16
b_float[1][3][1]: -4
b_float[1][3][2]: 8
b_float[2][1][1]: 10
b_float[2][1][2]: 9
b_float[2][2][1]: 2
b_float[2][2][2]: 7
b_float[2][3][1]: 1
b_float[2][3][2]: 9

Array of real numbers:
    Sum of b_float elements: 56.000000

Give string no 1: Trevor_Grancis_Jr

Give string no 2: Terror

Give string no 3: Try

Array b_char:
    b_char[0] = Trevor_Grancis_Jr
    b_char[1] = Terror
    b_char[2] = Try

The string with the maximum number of characters is no 1
'r' was found in the 2 place
'r' was found in the 6 place
'r' was found in the 9 place
'r' was found in the 17 place
```

Γ.

Να γραφεί πρόγραμμα στη γλώσσα C, το οποίο θα επιτελεί τα ακόλουθα:

(α) Θα λαμβάνει από το πληκτρολόγιο τις τιμές ενός πίνακα ακεραίων αριθμών `arr_int[5][5]` και θα εμφανίζει τον προκύπτοντα πίνακα στην οθόνη.

(β) Για κάθε γραμμή του πίνακα `arr_int`:

(β.1) Θα υπολογίζει το άθροισμα των στοιχείων της περιττών γραμμών (1^η, 3^η, 5^η).

(β.2) Θα υπολογίζει το άθροισμα των στοιχείων της 2^{ης} γραμμής, οι τιμές των οποίων είναι θετικές.

(β.3) Θα εμφανίζει τα δύο αθροίσματα στην οθόνη.

ΛΥΣΗ ΤΗΣ Γ

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

```
#define N 5
```

```
int main()
```

```
{
```

```
    int i,j,arr_int[N][N],line_sum_1,line_sum_2;
```

```
    for (i=0;i<N;i++)
```

```
        for (j=0;j<N;j++)
```

```
        {
```

```
            printf("\narr_int[%d][%d]: ",i+1,j+1);
```

```
            scanf("%d",&arr_int[i][j]);
```

```
        }
```

```
    printf("\n\nArray of integers:");
```

```
    for (i=0;i<N;i++) {
```

```
        printf("\n");
```

```
        for (j=0;j<N;j++) printf("\t%d",arr_int[i][j]);
```

```
    }
```

```
    line_sum_1=0;
```

```
    for (i=0;i<N;i=i+2)
```

```
        for (j=0;j<N;j++) line_sum_1+=arr_int[i][j];
```

```
    line_sum_2=0;
```

```
    for (j=0;j<N;j++)
```

```
        if (arr_int[1][j]>0) line_sum_2+=arr_int[1][j];
```

```
    printf("\tsum of elements of the odd lines = %d\n",line_sum_1);
```

```
    printf("\tsum of elements of the second line whose values are greater than 0 = %d\n",line_sum_2);
```

```
return 0;  
}
```

Array of integers:

-13	-34	555	-234	30	
124	456	87	32	-23	
23	54	57	66	65	
56	-789	54	-678	45	
432	78	-12	23	31	sum of elements of the odd lines = 1121
					sum of elements of the second line whose values are greater than 0 = 699