

Guía práctica de estudio 11: Arreglos unidimensionales y multidimensionales

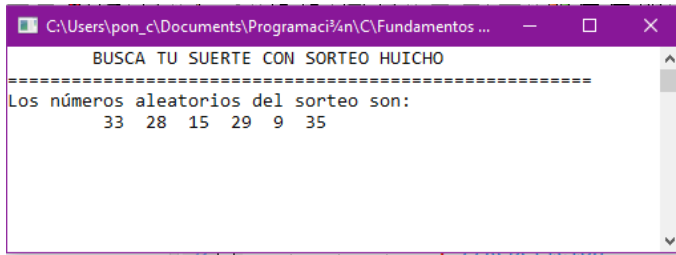
Codificar en lenguaje C lo siguientes ejercicios:

1. Elaborar un programa en C que realice la emulación de elección de los 6 número entero a jugar en un sorteo con un random de 1 a 39 empleando número aleatorio in repetir para ir almacenado en un arreglo unidimensional estático e imprimirlo en consola.

CÓDIGO EN C

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #define numsor 6
5
6  int main()
7  {
8      int numale[numsor];
9      int i, j;
10     int numrepetidos;
11     srand (time(NULL));
12
13     printf("\tBUSCA TU SUERTE CON SORTEO HUICHO\n");
14     printf("===== \n");
15
16     //Generación de Los números aleatorios
17     for (i=0; i<=numsor; i++)
18     {
19         numale[i] = rand()%39;
20     }
21     numrepetidos = 1;
22
23     //Comprobar números repetidos
24     while (numrepetidos == 1)
25     {
26         numrepetidos = 0;
27         for (i=0; i<=numsor; i++)
28         {
29             for (j=i+1; j<=numsor; j++)
30             {
31                 if (numale[i] == numale[j] && numale[i] != 39)
32                 { //Repetición
33                     numale[i] = numale[i] + 1;
34                     numrepetidos = 1;
35                 }
36             }
37         }
38     }
39
40     //Impresión de Los números aleatorios
41     printf("Los n%cmoros aleatorios del sorteo son:\n",163);
42     printf("\t");
43     for (i=0; i<=numsor; i++)
44     {
45         printf(" %d ",numale[i]);
46     }
47
48     getchar();
49     return 0;
```

PANTALLA DE EJECUCIÓN Y PRUEBA DE ESCRITORIO



2. Para un arreglo bidimensional estático con dimensiones máxima de 50 por 50, solicitar al usuario el número de renglones y columnas que desea, así como los valores para cada elemento, imprimir los valores en forma de matriz y el promedio de todos los números.

CÓDIGO EN C

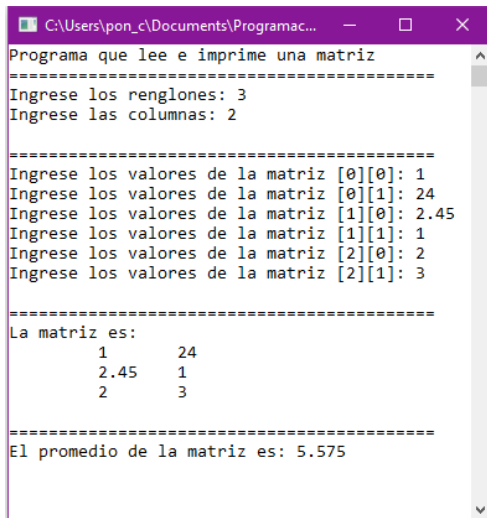
```
1  #include <stdio.h>
2  #define rengloni 50
3  #define columnai 50
4
5  int main(int argc, char **argv)
6  {
7      double matrix[rengloni][columnai]={'\0'};
8      short renglon, columna;
9      short limrenglones, limcolumnas;
10     double promedio, suma;
11
12     // CONDICIONES DE MATRIZ
13     printf("Programa que lee e imprime una matriz");
14     printf("\n===== \n");
15     do
16     {
17         printf ("Ingrese los renglones: ");
18         scanf("%hd", &limrenglones);
19     }while(limrenglones<1 || limrenglones>rengloni);
20
21     do
22     {
23         printf ("Ingrese las columnas: ");
24         scanf("%hd", &limcolumnas);
25     }while(limcolumnas<1 || limcolumnas>columnai);
26
27     //INGRESO DE DATOS
28
29     printf("\n===== \n");
30     for (renglon=0; renglon<limrenglones; renglon++)
31     {
32         for (columna=0; columna<limcolumnas; columna++)
33         {
34             printf("Ingrese los valores de la matriz [%hd][%hd]: ", renglon, columna);
35             scanf("%lf", &matrix[renglon][columna]);
36         }
37     }
38
39     //IMPRESION DE DATOS
40     printf("\n===== \n");
41     printf("La matriz es: \n");
```

```

42
43     for (renglon=0; renglon<limrenglones; renglon++)
44     {
45         for (columna=0; columna<limcolumnas; columna++)
46         {
47             printf("\t %g ",matrix[renglon][columna]);
48         }
49         printf("\n");
50     }
51
52
53     suma=0; //inicializamos la variable
54     for(renglon=0;renglon<limrenglones;renglon++)
55     {
56         for(columna=0;columna<limcolumnas;columna++)
57         {
58             suma=suma+matrix[renglon][columna];
59         }
60     }
61     promedio=suma/(limrenglones*limcolumnas);
62
63     printf("\n===== \n");
64     printf("El promedio de la matriz es: %g \n",promedio);
65
66
67
68
69
70     getchar();
71     getchar();
72     return 0;
73 }

```

PANTALLA DE EJECUCIÓN Y PRUEBA DE ESCRITORIO



The screenshot shows a Windows command prompt window titled "C:\Users\pon_c\Documents\Programac...". The program is titled "Programa que lee e imprime una matriz". It prompts the user to enter the number of rows (3) and columns (2). Then, it prompts the user to enter the values of the matrix. The input values are: [0][0]: 1, [0][1]: 24, [1][0]: 2.45, [1][1]: 1, [2][0]: 2, [2][1]: 3. The program then displays the matrix and calculates the average (5.575).

```

C:\Users\pon_c\Documents\Programac...
Programa que lee e imprime una matriz
=====
Ingrese los renglones: 3
Ingrese las columnas: 2

=====
Ingrese los valores de la matriz [0][0]: 1
Ingrese los valores de la matriz [0][1]: 24
Ingrese los valores de la matriz [1][0]: 2.45
Ingrese los valores de la matriz [1][1]: 1
Ingrese los valores de la matriz [2][0]: 2
Ingrese los valores de la matriz [2][1]: 3

=====
La matriz es:
    1      24
  2.45    1
    2      3

=====
El promedio de la matriz es: 5.575

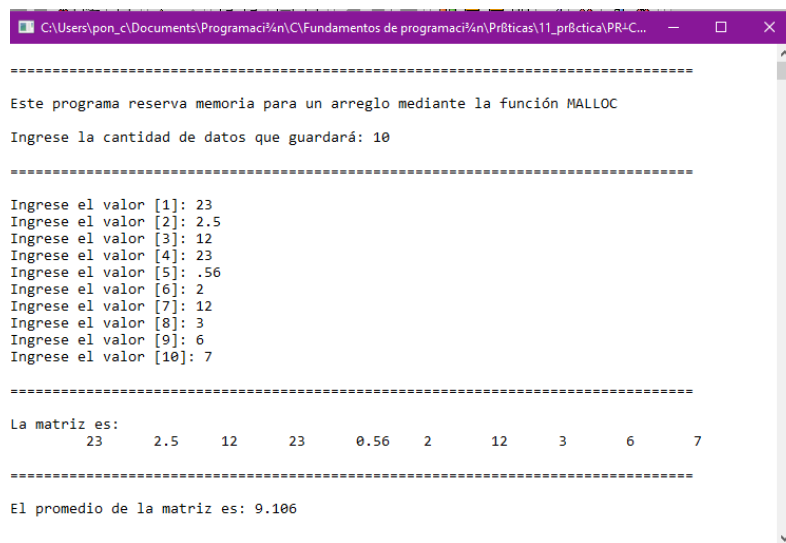
```

3. Empleando memoria dinámica y aritmética de apuntadores reservar espacio para un arreglo de tamaño dado por el usuario, quien ingresará una serie de números flotantes y el programa los mostrará en consola, así como su promedio

CÓDIGO EN C

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char **argv)
5  {
6      short renglon;
7      int limmatriz;
8      double promedio, suma;
9      double *pointer=NULL; //Para memoria dinámica
10
11     printf("\n===== \n\n");
12     printf("Este programa reserva memoria para un arreglo mediante la funci%cn MALLOC\n\n", 162);
13     printf("Ingrese la cantidad de datos que guardar%c: ", 160);
14     scanf("%d", &limmatriz);
15     pointer = (int*)malloc(limmatriz*sizeof(int));
16
17     if(pointer==NULL)
18     {
19         printf("\nLo lamentamos, memoria insuficiente");
20         exit(0);
21     }
22
23     //INGRESO DE DATOS
24
25     printf("\n===== \n\n");
26     for (renglon=0; renglon<limmatriz; renglon++)
27     {
28         printf("Ingrese el valor [%hd]: ", renglon+1);
29         scanf("%lf", &pointer[renglon]);
30     }
31
32     //IMPRESION DE DATOS
33
34     printf("\n===== \n\n");
35     printf("La matriz es: \n");
36
37     for (renglon=0; renglon<limmatriz; renglon++)
38     {
39         printf("\t %g ", pointer[renglon]);
40     }
41
42
43     suma=0; //inicializamos la variable
44     for(renglon=0; renglon<limmatriz; renglon++)
45     {
46         suma=suma+pointer[renglon];
47     }
48     promedio=suma/limmatriz;
49
50     printf("\n\n===== \n\n");
51     printf("El promedio de la matriz es: %g \n", promedio);
52
53     free(pointer);
54     fflush(stdin); //Si no limpio el buffer después de hacer todo se muere el programa :(
55     getchar();
56     getchar();
57     return 0;
58 }
59
```

PANTALLA DE EJECUCIÓN Y PRUEBA DE ESCRITORIO



```
=====
Este programa reserva memoria para un arreglo mediante la función MALLOC
Ingrese la cantidad de datos que guardará: 10
=====
Ingrese el valor [1]: 23
Ingrese el valor [2]: 2.5
Ingrese el valor [3]: 12
Ingrese el valor [4]: 23
Ingrese el valor [5]: .56
Ingrese el valor [6]: 2
Ingrese el valor [7]: 12
Ingrese el valor [8]: 3
Ingrese el valor [9]: 6
Ingrese el valor [10]: 7
=====
La matriz es:
      23      2.5      12      23      0.56      2      12      3      6      7
=====
El promedio de la matriz es: 9.106
```

CONCLUSIONES

Alfonso Murrieta Villegas

En la presente práctica mediante el uso de arreglo bidimensionales aprendí a cómo utilizarlos y sobretodo la importancia que tienen estos para poder guardar bastantes datos de una manera más ordenada, además con ayuda de apuntadores también aprendí a como reservar de manera temporal un espacio en la memoria RAM para así guardar estos arreglos.

Sin duda alguna una característica elemental de la programación sobre todo para poder optimizar los programas de la mejor manera posible.