



Alumno:
Ramírez Monjaraz Miguel Ángel

Profesor:
Pedro Nunez Yepiz

Ingeniería en computación
Grupo: 432

Materia:
Programación - Estructurada

Tema:
Actividad 9 - Librerías
ANEXOS

Fecha de entrega:
10 de Octubre del 2023



Libreria propia

MARM_ACTV9_01_432_PE.cpp

C Gauss.h



```
Gauss.h > valid_num(int, int, char [])
1 //                      Ensenada, Baja California a 07 de Octubre del 2023 //
2 //Nombre:                Miguel Angel Ramirez Monjaraz //
3 //Matricula:             372205 //
4 //Programa:              Libreria PROPIA, DEL MIGUEL, SOBRE FUNCIONES REUTILIZABLES //
5 //Nombre del programa: MARM_ACT9_01
6 #include <stdio.h>
7 #include <stdlib.h>
8 #define A 4
9
10
11 int busque_Sec(int vect[], int n,int num);
12 int sin_repetir(int vect[], int n,int num);
13 int sin_repetirMat(int mat[][A], int m, int n,int num);
14 int valid_num(int ri, int rf, char msge[]);
15 void ordenar(int vect[],int m);
16
17 int valid_num(int ri, int rf, char msge[])
18 {
19     char xnum[30];
20     int num;
21     do
22     {
23         puts(msge);
24         fflush(stdin);
25         gets(xnum);
26         num = atoi(xnum);
27     } while (num < ri || num > rf);
28     return num;
29 }
30
31 void ordenar(int vect[], int m)
32 {
33     int i, j;
34     int temp;
35     for (i = 0; i < m - 1; i++)
36     {
```



```
36     {
37         for (j = i + 1; j < m; j++)
38         {
39             if (vect[j] < vect[i])
40             {
41                 temp = vect[i];
42                 vect[i] = vect[j];
43                 vect[j] = temp;
44             }
45         }
46     }
47 }
48
49 int busque_Sec(int vect[], int n, int num)
50 {
51     for (int i = 0; i < n; i++)
52     {
53         if (num == vect[i])
54         {
55             return i;
56         }
57     }
58     return -1;
59 }
60
61 int sin_repetir(int vect[], int n, int num)
62 {
63     for (int i = 0; i < n; i++)
64     {
65         if (num == vect[i])
66         {
67             return 1;
68         }
69     }
70     return 0;
71 }
```

```
72
73 int sin_repetirMat(int mat[][A], int m, int n, int num)
74 {
75     for (int i = 0; i < n; i++)
76     {
77         for (int j = 0; j < m; j++)
78         {
79             if (num == mat[i][j])
80             {
81                 return 1;
82             }
83         }
84     }
85     return 0;
86 }
87
88
```



Código principal

```
MARM_ACTV9_01_432_PE.cpp X Gauss.h
MARM_ACTV9_01_432_PE.cpp > llena_mat(int[][4], int, int, int, int)
1 // Ensenada, Baja California a 07 de Octubre del 2023 //
2 //Nombre: Miguel Angel Ramirez Monjaraz //
3 //Matricula: 372205 //
4 //Programa: Que contiene como menu varias acciones, desde llenar un vector //
5 // hasta una matriz, ademas de tener una opcion individual //
6 //Nombre del programa: MARM_ACT9_01
7
8
9 #include <stdio.h>
10 #include <stdlib.h>
11 #include <time.h>
12 #include "Gauss.h"
13 #define M 15
14 #define A 4
15
16 int llena_mat(int mat[][A], int m, int n, int ri, int rf);
17 int llenar_vect(int vect[], int m, int ri, int rf);
18
19 void mostrar_mat(int mat[][A], int m, int n);
20 void mostrar_vect(int vect[], int m);
21
```

```
22 void menu();
23 int main()
24 {
25     menu();
26 }
27 void menu()
28 {
29     int resp = 1;
30     int op;
31     do
32     {
33         printf("    MENU    \n");
34         printf("1 - Llenar vector aleatorio sin repetir\n");
35         printf("2 - Llenar matriz aleatorio sin repetir\n");
36         printf("3 - Mostrar vector llenado aleatoriamente\n");
37         printf("4 - Mostrar matriz llenado aleatoriamente\n");
38         printf("5 - Ordenar el vector\n");
39         printf("6 - Buscar valor en el vector\n");
40         printf("0 - Salir\n");
41         op = valid_num(1,6,"Ingrese la opcion a realizar:");
42         switch (op)
43         {
44             case 1:
45                 int vect[M];
46                 int mat[A][A];
47                 int num;
48                 llenar_vect(vect, M, 100, 200);
49                 break;
50             case 2:
51                 llena_mat(mat, A, A, 1, 16);
52                 break;
53             case 3:
54                 mostrar_vect(vect, M);
55                 break;
```



```
58         break;
59     case 5:
60         ordenar(vect, M);
61         break;
62     case 6:
63         printf("Que numeor buscas?: ");
64         scanf("%d",&num);
65         busque_Sec(vect, M, num);
66         break;
67     default:
68         break;
69     }
70     printf("Presiona:\n");
71     printf("1 - Continuar\n");
72     printf("2 - Para salir\n");
73     resp = valid_num(1,2,"Deseas continuar? \n");
74 } while (resp == 1);
75 }
76
77 int llenar_vect(int vect[], int m, int ri, int rf)
78 {
79     int numero, rango;
80     rango = (rf - ri) + 1;
81     srand(time(NULL));
82     for (int i = 0; i < m; i++)
83     {
84         do
85         {
86             numero = (rand() % rango) + ri;
87
88             } while (sin_repetir(vect, m, numero) == 1);
89         vect[i] = numero;
90     }
91     return vect[M];
92 }
93
```



```
3
4  int llena_mat(int mat[][4], int m, int n, int ri, int rf)
5  {
6      int numero, rango;
7      srand(time(NULL));
8      rango = (rf - ri) + 1;
9      for (int i = 0; i < m; i++)
10     {
11         for (int j = 0; j < n; j++)
12         {
13             do
14             {
15                 numero = (rand() % rango) + ri;
16             } while (sin_repetirMat(mat,m,n,numero) == 1);
17             mat[i][j] = numero;
18         }
19     }
20     return mat[4][4];
21 }
2
3
4  void mostrar_vect(int vect[], int m)
5  {
6      for (int i = 0; i < m; i++)
7      {
8          printf("[");
9          printf("%d", vect[i]);
10         printf("]\n");
11     }
12 }
13
```

```
123
124 void mostrar_mat(int mat[][A], int m, int n)
125 {
126     for (int i = 0; i < m; i++)
127     {
128         for (int j = 0; j < n; j++)
129         {
130             printf("[");
131             printf("%d", mat[i][j]);
132             printf("]");
133         }
134         printf("\n");
135     }
136 }
137
```



Salida de pantalla

```
MENU
1 - Llenar vector aleatorio sin repetir
2 - Llenar matriz aleatorio sin repetir
3 - Mostrar vector llenado aleatoriamente
4 - Mostrar matriz llenado aleatoriamente
5 - Ordenar el vector
6 - Buscar valor en el vector
0 - Salir
Ingresa la opcion a realizar:
1
Presiona:
1 - Continuar
2 - Para salir
Deseas continuar?

1
MENU
1 - Llenar vector aleatorio sin repetir
2 - Llenar matriz aleatorio sin repetir
3 - Mostrar vector llenado aleatoriamente
4 - Mostrar matriz llenado aleatoriamente
5 - Ordenar el vector
6 - Buscar valor en el vector
0 - Salir
Ingresa la opcion a realizar:
2
Presiona:
1 - Continuar
2 - Para salir
Deseas continuar?
```

(EJECUTANDO LA OPCIÓN 1 Y 2)



Mostrando el vector y matriz(OPCIÓN 3 Y 4)

```
ingrese la opcion a realizar:
3
[3]
[9]
[4]
[13]
[11]
[2]
[7]
[12]
[16]
[10]
[15]
[1]
[8]
[14]
[5]
```

```
ingrese la opcion a realizar:
4
[3][9][4][13]
[11][2][7][12]
[16][10][15][1]
[8][14][5][6]
```

Ordenar el vector

```
ingrese la opcion a realizar:
3
[1]
[2]
[3]
[4]
[5]
[7]
[8]
[9]
[10]
[11]
[12]
[13]
[14]
[15]
[16]
```




```
[105]  
[120]  
[121]  
[122]  
[123]  
[124]  
[148]  
[158]  
[161]  
[162]  
[169]  
[171]  
[178]  
[191]  
[198]
```

Presiona:

1 - Continuar

2 - Para salir

Deseas continuar?

1

MENU

1 - Llenar vector aleatorio sin repetir

2 - Llenar matriz aleatorio sin repetir

3 - Mostrar vector llenado aleatoriamente

4 - Mostrar matriz llenado aleatoriamente

5 - Ordenar el vector

6 - Buscar valor en el vector

0 - Salir

Ingresa la opcion a realizar:

6

Que numero buscas?: 198

El numero si se encontro en la posicion 15 con indice de 14

Presiona: