



**Alumno:**  
**Ramírez Monjaraz Miguel Ángel**

**Profesor:**  
**Pedro Nunez Yepiz**

**Ingeniería en computación**  
**Grupo: 432**

**Materia:**  
**Programación - Estructurada**

**Tema:**  
**Actividad 8 Arreglos - Funciones**  
**ANEXOS**

**Fecha de entrega:**  
**03 de Octubre del 2023**



G+ MARM\_ACTV8\_432\_PE.cpp 1 ●

G+ JA.cpp

G+ MARM\_ACTV8\_432\_PE.cpp > ...

```
1 //Miguel Angel Ramirez Monjaraz 372205
2 //Ensenada, Baja California a 30 de Septiembre del 2023
3 //Programa que llena vectores y matriz, ademas las despliega
4 //Nombre de la actividad: MARM_ACT8_01_432
5
6 #include<stdio.h>
7 #include<stdlib.h>
8 #include<string.h>
9 #include <time.h>
10 #define M 10
11
12 //Funciones extras
13 void menu();
14 int valida_num(int ri, int rf);
15 bool no_repetir(int vect2[], int n);
16
17 //Funciones para llenar
18 void vect_manual(int vect1[], int m);
19 void vect_alea(int vect2[], int m, int ri, int rf);
20 void vector3(int vect1[], int vect2[], int vect3[], int m);
21 void matriz12(int vect1[], int vect2[], int matriz[4][4]);
22
23 //Funciones para imprimir
24 void imprimir_vects(int vect1[], int vect2[], int vect3[], int m);
25 void imprimir_mat(int matriz[4][4]);
26 int main()
27 {
28     menu();
29 }
```

```
1 // Funcion del menu//
2 void menu()
3 {
4     int res, m, vect1[M], vect2[M], vect3[M*2], matriz[4][4];
5     do{
6         printf("\n          MENU          ");
7         printf("\n Presiones el numero indicado para lo que quiera realizar");
8         printf("\n1 - Llenar vector 1 de manera manual");
9         printf("\n2 - Llenar vector 2 de forma aleatorea");
10        printf("\n3 - Llenar vector 3 con el vector 1 y 2");
11        printf("\n4 - Imprimir todos los vectores anteriores");
12        printf("\n5 - Llenar matriz 4x4 con vector 1 y 2");
13        printf("\n6 - Imprimir matriz");
14        printf("\n0 - PARA SALIR");
15        printf("\nQue accion deseas realizar? ");
16        scanf("%d", &res);
17        switch(res)
18        {
19            case 1:
20                printf("\nSe llenara el vector 1 de manera manual");
21                vect_manual(vect1, M);
22                break;
```



```
55         case 2:
56             printf("\nSe llenara de manera aleatoria el vector 2");
57             vect_alea(vect2, M, 1, 10);
58             break;
59
60         case 3:
61             printf("\n Has llenado el vector 3, con el vector 1 y 2");
62             vector3(vect1, vect2, vect3, M);
63             break;
64
65         case 4:
66             printf("\nSe mostraran los vectores llenados anteriormente ");
67             imprimir_vects(vect1, vect2, vect3, M);
68             break;
69
70         case 5:
71             printf("\nSe llenara la matriz con el vector 1 y 2");
72             matriz12(vect1, vect2, matriz);
73             break;
74
75         case 6:
76             printf("\nHas decidido mostrar la matriz");
77             imprimir_mat(matriz);
78             break;
79         case 0:
80             printf("Saliste");
81             res = 0;
82         default:
83             break;
```

```
84     }
85     }while(res != 0);
86 }
87
88 // Funcion que nos ayuda en la 2 funcion para evitar los numeros repetidos //
89 bool no_repetir(int vect2[],int n)
90 {
91     int i;
92     for (i = 0; i <10; i++)
93     {
94         if (n == vect2[i])
95         {
96             return true;
97         }
98     }
99     return false;
100 }
```



```
01
02 // Funcion para validar numeros que ingresan //
03 int valida_num(int ri, int rf)
04 {
05     int n;
06     char xnum[30];
07     printf("\nDame un numero entre el %d y %d: ", ri, rf);
08     fflush(stdin);
09     gets(xnum);
10     n = atoi(xnum);
11     if (n > rf)
12     {
13         printf("\n El numero esta fuera del rango, te pasaste");
14         n = 70;
15     }
16
17     if (n < ri)
18     {
19         printf("\n El numero esta fuera del rango, fue menor al rango propuesto");
20         n = 30;
21     }
22
23     return n;
24 }
```

```
26 //FUNCIONES DE LA PRACTICA//
27
28
29 void vect_manual(int vect1[], int m)
30 {
31     int i;
32     for(i = 0; i < m; i++)
33     {
34         vect1[i] = valida_num(30, 70);
35     }
36 }
```

```
40
41 void vect_alea(int vect2[], int m, int ri, int rf)
42 {
43     int rango, i, n;
44     rango = (rf - ri) + 1;
45     srand(time(NULL));
46     for(i = 0; i < m; i++)
47     {
48         do
49         {
50             n = (rand()%rango) + ri;
51         } while (no_repetir(vect2, n));
52         vect2[i] = n;
53     }
54 }
```



```
159
160 void vector3(int vect1[], int vect2[], int vect3[], int m)
161 {
162     int i;
163     for(i = 0; i < m; i++)
164     {
165         vect3[i] = vect1[i];
166     }
167
168     for(i = 0; i < m; i++)
169     {
170         vect3[i+10] = vect2[i];
171     }
172 }
```

```
178 void imprimir_vects(int vect1[], int vect2[], int vect3[], int m)
179 {
180     int i;
181     printf("\nVECTOR 1");
182     printf("\n[");
183     for (i = 0; i < m; i++)
184     {
185         printf("%d, ", vect1[i]);
186     }
187     printf("]");
188     printf("\n");
189
190     printf("\nVECTOR 2");
191     printf("\n[");
192     for (i = 0; i < m; i++)
193     {
194         printf("%d, ", vect2[i]);
195     }
196     printf("]");
197     printf("\n");
198
```

```
199     printf("\nVECTOR 3");
200     printf("\n[");
201     for (i = 0; i < m*2; i++)
202     {
203         printf("%d, ", vect3[i]);
204     }
205     printf("]");
206     printf("\n");
```



```
213
214 void matriz12(int vect1[], int vect2[], int matriz[4][4])
215 {
216     int j, i, k;
217     k = 0;
218     for (j = 0; j < 2; j++)
219     {
220         for (i = 0; i < 4; i++)
221         {
222             matriz[j][i] = vect1[k];
223             k = k + 1;
224         }
225     }
226
227     matriz[2][0] = vect1[8];
228     matriz[2][1] = vect1[9];
229     matriz[2][2] = vect2[0];
230     matriz[2][3] = vect2[1];
231
232     for (i = 0; i < 4; i++)
233     {
234         matriz[3][i] = vect2[i+2];
235     }
```

```
244 void imprimir_mat(int matriz[4][4])
245 {
246     int j, i;
247     printf("\n");
248     for (j = 0; j < 4; j++)
249     {
250         printf("\n[");
251         for(i = 0; i < 4; i++)
252         {
253             printf("%d, ", matriz[j][i]);
254         }
255         printf("]");
256     }
257     printf("\n");
258
259 }
```



## Salida de pantalla:

```
MENU
Presiones el numero indicado para lo que quiera realizar
1 - Llenar vector 1 de manera manual
2 - Llenar vector 2 de forma aleatoria
3 - Llenar vector 3 con el vector 1 y 2
4 - Imprimir todos los vectores anteriores
5 - Llenar matriz 4x4 con vector 1 y 2
6 - Imprimir matriz
0 - PARA SALIR
Que accion deseas realizar? 1
```

```
Se llenara el vector 1 de manera manual
Dame un numero entre el 30 y 70: 34
```

```
Dame un numero entre el 30 y 70: 54
```

```
Dame un numero entre el 30 y 70: 56
```

```
Dame un numero entre el 30 y 70: 45
```

```
Dame un numero entre el 30 y 70: 65
```

```
Dame un numero entre el 30 y 70: 71
```

```
El numero esta fuera del rango, te pasaste
Dame un numero entre el 30 y 70: 31
```

```
Dame un numero entre el 30 y 70: 30
```

```
Dame un numero entre el 30 y 70: 50
```

```
Dame un numero entre el 30 y 70: 53
```

```
Que accion deseas realizar? 2
```

```
Se llenara de manera aleatoria el vector 2
```

```
Que accion deseas realizar? 3
```

```
Has llenado el vector 3, con el vector 1 y 2
```



Que accion deseas realizar? 4

Se mostraran los vectores llenados anteriormente

VECTOR 1

[34, 54, 56, 45, 65, 70, 31, 30, 50, 53, ]

VECTOR 2

[4, 10, 1, 3, 5, 2, 6, 9, 7, 8, ]

VECTOR 3

[34, 54, 56, 45, 65, 70, 31, 30, 50, 53, 4, 10, 1, 3, 5, 2, 6, 9, 7, 8, ]

MENU

Que accion deseas realizar? 6

Has decidido mostrar la matriz

[34, 54, 56, 45, ]

[65, 70, 31, 30, ]

[50, 53, 4, 10, ]

[1, 3, 5, 2, ]