

Universidad Autónoma de Baja California



Ingeniería en Software y Tecnologías Emergentes.

### **Actividad 8. Arreglos en C.**

Materia: Programación Estructurada.

Maestro: Pedro Yepiz

Alumno: Diego Quiros 372688

Fecha: 10/01/2023

## Ejercicio 1.Llenar Vector 1 manualmente, Solo admite números dentro de cierto rango.

```
//Funcion Que llena un vector manualmente dentro de cierto rango de numeros
//QVDD_Act8_01_932.c
void llena_vectorManual (int vector1[10],int m,int ri,int rf)
{
    int num,aux;
    printf("Los datos del Vector deben estar entre los valores %d y %d\n",ri,rf);
    for (int i = 0; i < m; i++)
    {
        aux=0;
        scanf("%d",&num);
        if (num<ri || num>rf)
        {
            aux=1;

            do
            {
                printf("Valor fuera de limite\n");
                printf("El numero debe estar entre %d y %d\n",ri,rf);
                scanf("%d",&num);
                if (num>=ri && num<=rf)
                {
                    aux=0;
                }
            } while (aux==1);
        }
        vector1[i]=num;
    }
}
```

```

M E N U
1. Llenar Vector 1 (Manualtemnte)
2. Llenar vector 2 Aleatoriamente
3. Llenar vector 3 (Con Vector 1 y Vector 2)
4. Imprimir Vectores
5. Llena Matriz 4 x 4
6. Imprimir Matriz
0. Salir
Ingresa funcion a utilizar:
1
Los datos del Vector deben estar entre los valores 30 y 70
25
Valor fuera de limite
El numero debe estar entre 30 y 70
75
Valor fuera de limite
El numero debe estar entre 30 y 70
35
36
37
38
39
40
41
42
43
44
Press any key to continue . . . █

```

## Ejercicio 2. LLENAR VECTOR 2 ALEATORIAMENTE

```

//Funcion Que llena un vector aleatoriamente sin repetirse dentro de cierto rango de numeros
//QVDD_Act8_02_932.c
void llena_vectorRandom (int vector2[10],int m,int ri,int rf)
{
    srand(time(NULL));
    int aux,num,i,j;
    for (i = 0; i < m; i++)
    {
        aux=0;
        num=(rand()%(rf-ri))+ri;
        for (j = 0; j<m; j++)
        {
            if (num==vector2[j])
            {
                aux=1;
            }
        }

        if (aux==1)
        {
            i--;
        }
        else
        {
            vector2[i]=num;
        }
    }
    printf("\nEl Vector 2 ha sido llenado con valores del %d hasta %d\n",ri,rf);
}

```

```

      M E N U
1. Llenar Vector 1 (Manualtemnte)
2. Llenar vector 2 Aleatoriamente
3. Llenar vector 3 (Con Vector 1 y Vector 2)
4. Imprimir Vectores
5. Llena Matriz 4 x 4
6. Imprimir Matriz
0. Salir

```

Ingresa funcion a utilizar:

2

El Vector 2 ha sido llenado con valores del 1 hasta 20  
 Press any key to continue . . . █

### Ejercicio 3. LLENAR VECTOR 3 (CON VECTOR1 Y VECTOR2).

```
// FUNCION QUE LLENA VECTOR 3 CON DATOS DEL VECTOR1 Y VECTOR2.
//QVDD_Act8_03_932.c
void llena_vector3conV1yV2(int vector3[20],int vector1[10],int vector2[10],int m)
{
    int i,j;
    for (i = 0; i < m; i++)
    {
        vector3[i]=vector1[i];
    }
    j=0;
    for (i = 10; i < m; i++)
    {
        vector3[i]=vector2[j];
        j++;
    }
    printf("\nEl Vector ha sido llenado con los datos del Vector 1 y Vector 2");
}
```

```

M E N U
1. Llenar Vector 1 (Manualtemnte)
2. Llenar vector 2 Aleatoriamente
3. Llenar vector 3 (Con Vector 1 y Vector 2)
4. Imprimir Vectores
5. Llena Matriz 4 x 4
6. Imprimir Matriz
0. Salir
Ingresa funcion a utilizar:
3

El Vector ha sido llenado con los datos del Vector 1 y Vector 2
```

## Ejercicio 4. Imprimir Vectores.

```
// FUNCION QUE IMPRIME LOS DATOS DE LOS 3 VECTORES.
//QVDD_Act8_04_932.c
void imprimir_vectores(int vect[],int m)
{
    for (int i = 0; i < m; i++)
    {
        printf("%d - [%d]\n",i,vect[i]);
    }
}
```

## M E N U

1. Llenar Vector 1 (Manualtemnte)
2. Llenar vector 2 Aleatoriamente
3. Llenar vector 3 (Con Vector 1 y Vector 2)
4. Imprimir Vectores
5. Llena Matriz 4 x 4
6. Imprimir Matriz
0. Salir

Ingresa funcion a utilizar:

4

Vector 1

- 0 - [35]
- 1 - [36]
- 2 - [37]
- 3 - [38]
- 4 - [39]
- 5 - [40]
- 6 - [41]
- 7 - [42]
- 8 - [43]
- 9 - [44]

Vector 2

- 0 - [8]
- 1 - [11]
- 2 - [5]
- 3 - [9]
- 4 - [13]
- 5 - [19]
- 6 - [6]
- 7 - [17]
- 8 - [10]
- 9 - [1]

Vector 3  
0 - [35]  
1 - [36]  
2 - [37]  
3 - [38]  
4 - [39]  
5 - [40]  
6 - [41]  
7 - [42]  
8 - [43]  
9 - [44]  
10 - [8]  
11 - [11]  
12 - [5]  
13 - [9]  
14 - [13]  
15 - [19]  
16 - [6]  
17 - [17]  
18 - [10]  
19 - [1]

## Ejercicio 5. Llena Matriz 4x4 con datos de los vectores 1 y 2.

```
// FUNCION QUE LLENA UNA MATRIZ 4X4 CON LOS DATOS DE LOS  
// VECTORES 1 Y 2.  
// QVDD_Act8_05_932.c  
void llena_matriz4x4(int matriz[4][4], int vector3[], int m, int n)  
{  
    int i, j;  
    int aux=0;  
    for (i = 0; i < m; i++)  
    {  
        for(j = 0; j < n ; j++)  
        {  
            matriz[i][j]=vector3[aux];  
            aux++;  
        }  
    }  
    printf("Se ha llenado la matriz con los valores de los vectores anteriores\n");  
}
```

You, 11 minutes ago • actividad 8

```

M E N U
1. Llenar Vector 1 (Manualtemnte)
2. Llenar vector 2 Aleatoriamente
3. Llenar vector 3 (Con Vector 1 y Vector 2)
4. Imprimir Vectores
5. Llena Matriz 4 x 4
6. Imprimir Matriz
0. Salir
Ingresa funcion a utilizar:
5
Se ha llenado la matriz con los valores de los vectores anteriores

```

## Ejercicio 6. Imprimir Matriz,

```

// FUNCION QUE IMPRIME UNA MATRIZ 4X4
//QVDD_Act8_06_932.c
void imprimir_matriz(int matriz[4][4],int m,int n)
{
    int i,j;

    for (i = 0; i < m; i++)
    {
        for(j = 0; j<n ; j++)
        {
            printf("[%d]",matriz[i][j]);
        }
        printf("\n");
    }
}

```



## M E N U

1. Llenar Vector 1 (Manualtemnte)
2. Llenar vector 2 Aleatoriamente
3. Llenar vector 3 (Con Vector 1 y Vector 2)
4. Imprimir Vectores
5. Llena Matriz 4 x 4
6. Imprimir Matriz
0. Salir

Ingresa funcion a utilizar:

6

[35][36][37][38]

[39][40][41][42]

[43][44][8][11]

[5][9][13][19]

Press any key to continue . . . █