

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Solano Meza Angel Daniel

Matrícula: 372453

Maestro: Pedro Núñez Yépiz

Actividad No. : 8

Tema - Unidad : Arreglos - Unidad 1

Ensenada Baja California a 30 de Septiembre del 2023

1.

```
case 1:
    printf("LLENAR VECTOR 1 MANUALMENTE\n");
    LlenarVector(vector1, 10, 30, 70, 0);
    system("CLS");
    printf("Vector 1 relleno\n");
    system("PAUSE");
    break;
```

```
void LlenarVector(int vect[], int m, int ri, int rf, int op)
{
    // VARIABLES LOCALES
    int i, j, num;
    system("CLS");
    // AQUI DESARROLLO PROGRAMA
    if (op == 1) // Opcion relleno automatico
    {
        for (i = 0; i < m; i++) // Le asigna un valor a cada indice del vector
        {
            vect[i] = (rand() % (rf - ri)) + ri;
            for (j = 0; j < i; j++) // Barre el vector hasta i comprobando que no se repita el numero
            {
                if (vect[i] == vect[j]) // Si se encuentra un numero repetido lo vuelve a generar
                {
                    vect[i] = (rand() % (rf - ri)) + ri;
                }
            }
        }
    }
    else // Opcion relleno manual
    {
        for (i = 0; i < m; i++)
        {
            printf("    Espacio [%d]\n", i + 1);
            vect[i] = Validar("Numero entre 30 y 70\n", 30, 70); // Valida la entrada manual
        }
    }
}
```

2.

```
case 2:
    printf("LLENAR VECTOR 2 AUTOMATICAMENTE\n");
    LlenarVector(vector2, 10, 1, 20, 1);
    system("CLS");
    printf("Vector 2 relleno\n");
    system("PAUSE");
    break;
```

```

void LlenarVector(int vect[], int m, int ri, int rf, int op)
{
    // VARIABLES LOCALES
    int i, j, num;
    system("CLS");
    // AQUI DESARROLLO PROGRAMA
    if (op == 1) // Opcion relleno automatico
    {
        for (i = 0; i < m; i++) // Le asigna un valor a cada indice del vector
        {
            vect[i] = (rand() % (ri - rf)) + ri;
            for (j = 0; j < i; j++) // Barre el vector hasta i comprobando que no se repita el numero
            {
                if (vect[i] == vect[j]) // Si se encuentra un numero repetido lo vuelve a generar
                {
                    vect[i] = (rand() % (ri - rf)) + ri;
                }
            }
        }
    }
    else // Opcion relleno manual
    {
        for (i = 0; i < m; i++)
        {
            printf("    Espacio [%d]\n", i + 1);
            vect[i] = Validar("Numero entre 30 y 70\n", 30, 70); // Valida la entrada manual
        }
    }
}

```

3.

```

case 3:
    printf("LLENAR VECTOR 3 \n");
    Vector3(vector3, vector1, vector2);
    system("CLS");
    printf("Vector 3 relleno\n");
    system("PAUSE");
    break;

```

```

void Vector3(int vect3[], int vect2[], int vect1[])
{
    // VARIABLES LOCALES
    int i;
    system("CLS");
    for (i = 0; i < 10; i++)
    {
        vect3[i] = vect2[i]; // Los primeros 10 espacios se llenan con el vector 2
        vect3[i + 10] = vect1[i]; // Del 10 al 20 con el vector 1
    }
    // AQUI DESARROLLO PROGRAMA
}

```

4.

```

        break;
    case 4:
        ImprimirVector("VECTOR 1", vector1, 10);
        ImprimirVector("VECTOR 2", vector2, 10);
        ImprimirVector("VECTOR 3", vector3, 20);
        break;
    case 5:

```

```

void ImprimirVector(char msg[], int vect[], int m)
{
    // VARIABLES LOCALES
    int i;
    system("CLS");
    // AQUI DESARROLLO PROGRAMA
    printf("%s\n", msg);
    for (i = 0; i < m; i++)
    {
        printf("[%d] --> %d\n", i + 1, vect[i]); //Imprime el valor del del arreglo en el indice
    }
    system("PAUSE");
}

```

5.

```

    case 5:
        LlenarMatriz(matriz, vector1, vector2, 4, 4);
        break;

```

```

void LlenarMatriz(int matriz[][4], int vect1[], int vect2[], int m, int n)
{
    // VARIABLES LOCALES
    int i, j, k;
    system("CLS");
    // AQUI DESARROLLO PROGRAMA
    for (i = 0, k=0; i < m; i++)
    {
        for (j = 0; j < n; j++, k++)
        {
            if (k < 10) //Llena la matriz con los 10 datos del vector 1
            {
                matriz[i][j] = vect1[k];
            }
            else //Cuando se termina el vector 1 continua con el vector 2
            {
                matriz[i][j] = vect2[k-10];
            }
        }
    }
    printf("Matriz rellena\n");
    system("PAUSE");
}
//*****

```

6.

```
case 6:  
    ImprimirMatriz(matriz, 4, 4);  
    break;  
}
```

```
void ImprimirMatriz(int matriz[][4], int m, int n)
```

```
{  
    // VARIABLES LOCALES  
    int i, j;  
    system("CLS");  
    // AQUI DESARROLLO PROGRAMA  
    printf(" IMPRIMIR MATRIZ\n");  
    for (i = 0; i < m; i++)  
    {  
        for (j = 0; j < n ; j++)  
        {  
            printf("Espacio [%d][%d] --> [%d]\n", i+1, j+1, matriz[i][j]); //Imprime los valores del arreglo bidimensional  
        }  
    }  
    system("PAUSE");  
}
```