



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Teresa Rivas Gómez

Matrícula: 372565

Maestro: Pedro Núñez Yépiz

Actividad No. : 8

Tema - Unidad : Arreglos en C

Ensenada, Baja California a 01 de Octubre del 2023



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Introducción:

En esta actividad continuamos trabajando con el uso de las funciones, pero esta vez agregando arreglos y parámetros para el desarrollo de cada ejercicio. Estaremos viendo como lleve a cabo la ejecución de cada uno y como el uso de arreglos/vectores y matrices ayuda a que estos se lleven a cabo de una mejor manera. También vemos ciclos, condiciones y estructuras anidadas.

Competencias

Se espera que podamos tener un dominio de este tema (Arreglos en C y funciones) y que con la práctica podamos desarrollar programas complejos que utilicen todas estas funciones sin necesidad de recurrir a internet o libro; esto nos ayudara a optimizar procesos y a que utilicemos la razón y la lógica en nuestros programas para que queden mejor estructurados.

Documentación

Para la teoría de esta actividad utilice mis apuntes en clase y los enlaces que dejo en la actividad y el de la presentación del otro profesor donde explica este tema, también un poco de ayuda de mis compañeros/hermana. En general, me parece que pude desarrollarlo correctamente sin la necesidad de recurrir mucho a internet.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Procedimiento

ACTIVIDAD 8

ARREGLOS EN C

INSTRUCCIONES

- 1.- Realiza un programa en C
(Funciones e Introduccion a Arreglos en C)
- 2.- Realiza reporte de práctica y anexos (Teoria Arreglos y Funciones)
- 3.- Sube a Blackboard: Programa , Reporte de practica y anexo con capturas y código (3 Archivos 1 cpp, 2 PDF)
- 4.- Sube a GitHub en tu repositorio los 3 documentos y poner enlace en BlackBoard

NOTA: No se te olvide poner portada en los documentos e informacion en el programa, recuerda que tus conclusion es muy importante y sobre todo saber si se cumple el objetivo del tema

ACTIVIDAD 8

Realiza programa en C el programa deberá tener el siguiente menú.

MENÚ

- 1.- LLENAR VECTOR 1 (MANUALMENTE)
- 2.- LLENAR VECTOR 2 ALEATORIAMENTE
- 3.- LLENAR VECTOR 3 (CON VECTOR1 Y VECTOR2)
- 4.- IMPRIMIR VECTORES
- 5.- LLENA MATRIZ 4 X 4
- 6.- IMPRIMIR MATRIZ
- 0.- SALIR

NOTA: EL PROGRAMA DEBERÁ REPETIRSE CUANTAS VECES LO DESEE EL USUARIO

NOTA 2: EL VECTOR 1 DE 10 POSICIONES, NÚMEROS DEL 30 AL 70

NOTA 3: EL VECTOR 2 DE 10 POSICIONES CON NÚMEROS GENERADOS ALEATORIAMENTE DEL 1 AL 20 (SIN REPETIR)

NOTA 4: EL VECTOR 3 DE 20 POSICIONES, CON LOS DATOS DEL ARREGLO1 Y ARREGLO2

NOTA 5: MATRIZ 4 X 4 LLENARLA CON LOS DATOS DEL VECTOR1 Y VECTOR2,

ENLACES DE AYUDA Y REFERENCIA

https://drive.google.com/drive/folders/13eq-7u7vxi_TC9ssK-Cqo3zt9xKTsKl?usp=sharing



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Resultados y Conclusiones

En esta parte agregare las partes mas relevantes de mi código con su respectiva documentación y comentarios.

- **Declaración de funciones a utilizar:**

```
// Menu para el usuario
int msges();
void menu();
// Validacion
int validar(char mensj[], int ri, int rf) ;
// Funcion para cada ejercicio
void vector_1(int vector[10], int m); // LLENAR VECTOR 1 (MANUALMENTE)
void vector_2(int vector[10], int m, int ri, int rf); // LLENAR VECTOR 2 ALEATORIAMENTE
void vector_3(int vector1[10], int vector2[10], int vector3[20], int m1, int m2); // LLENAR VECTOR
3 (CON VECTOR1 Y VECTOR2)
void imp_vector(int vector1[10], int size1, int vector2[10], int size2, int vector3[20], int
size3); // IMPRIMIR VECTORES
void llen_matriz(int matriz[4][4], int m); // LLENA MATRIZ 4 X 4
void imp_matriz(int matriz[4][4], int m); // IMPRIMIR MATRIZ
```

- **Menus, para el usuario y el de los casos según la opción que elija:**

```
// Menu
int main()
{
    menu();
    return 0;
}
// Menu para el usuario
int msges()
{
    int op;
    system ("CLS");
    printf ("\n MENU DE LA ACTIVIDAD 8 \n");
    printf ("1.- LLENAR VECTOR 1 (MANUALMENTE) \n");
    printf ("2.- LLENAR VECTOR 2 ALEATORIAMENTE \n");
    printf ("3.- LLENAR VECTOR 3 (CON VECTOR1 Y VECTOR2) \n");
    printf ("4.- IMPRIMIR VECTORES \n");
    printf ("5.- LLENA MATRIZ 4 X 4 \n");
    printf ("6.- IMPRIMIR MATRIZ \n");
    printf ("0.- SALIR \n");
    // Matriz de caracteres para la conversion de la cadena
    char mensaje[] = "ESCOGE UNA OPCION: ";
    op = validar(mensaje, 0, 6);
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
// Retorno a opcion
return op;
}
// Casos para el menu segun la opcion que elijan
void menu()
{
    int op;
    int vector1[10];
    int vector2[10];
    int vector3[10];
    int matriz[4][4];
    // Ciclo
    do{
        op=msges();
        // Casos
        switch (op)
        {
            case 1:
                vector_1(vector1, 10);
                break;
            case 2:
                vector_2(vector2, 10, 1, 20);
                break;
            case 3:
```

- En esta sección del caso 3, agrego una condición para verificar que ya se hayan llenado los vectores 1 y 2 previamente:

```
        if (vector1[0] != 0)
        {
            if(vector2[0] != 0)
            {
                vector_3(vector1, vector2, vector3, 10, 10);
            }
        }
        else
        {
            printf("Primero llene los vectores 1 y 2.\n");
        }
        break;
    case 4:
        imp_vector(vector1, 10, vector2, 10, vector3, 20);
        break;
    case 5:
        llen_matriz(matriz, 4);
        break;
    case 6:
        imp_matriz(matriz, 4);
        break;
    case 0:
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        printf("Saliste del programa.\n");
        break;
    default:
        printf("Opcion Invalida.\n");
        break;
    }
}while (op != 0);
}
```

- Validación:

```
// Validacion
int validar(char mensj[], int ri, int rf)
{
    int num;
    // Cadena que va a leer el mensaje ingresado
    char cadena[50];
    // Proceso y desarrollo
    do
    {
        printf("%s", mensj);
        // Borrar basura
        fflush(stdin);
        gets(cadena);
        num = atoi(cadena);
    } while (num < ri || num > rf);
    // Retorna el valor que num, entre los rangos dados
    return num;
}
```

- Ejercicio 1:

```
// Ejercicio 1
// LLENAR VECTOR 1 (MANUALMENTE)
void vector_1(int vector[10], int m)
{
    system ("CLS");
    int i;
    // Ciclo para ingresar numeros
    for ( i = 0; i < m; i++)
    {
        // Numeros que ingresara el usuario manualmente
        int num;
        num = validar("Ingresa un numero entre el 30 y el 70: \n", 30, 70);
        // Guarda el numero como CHAR
        vector[i] = num;
        // El +1 es para que no imprima el 0
        printf ("Valor de la posicion %d : %d\n", i + 1, vector[i] );
    }
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
// Condicion para mostrar cuando el vector ya este lleno en su capacidad.
if ( i == m - 1 )
{
    printf ("El vector esta lleno\n");
    system ("PAUSE");
}
}
```

● Ejercicio 2:

```
// Ejercicio 2
// LLENAR VECTOR 2 ALEATORIAMENTE
void vector_2(int vector[10], int m, int ri, int rf)
{
    system("CLS");
    // Declarar variables
    int i = 0;
    int valor;
    // Inicializar el vector con elementos no validos
    for (int j = 0; j < 10; j++)
    {
        vector[j] = -1;
    }
    // Ciclo para generar los valores aleatorios sin repetir
    while (i < m)
    {
        valor = (rand() % (rf - ri + 1)) + ri;
        bool valor_repetido = false;
        // Verificar si el valor ya existe en el vector
        for (int j = 0; j < 10; j++)
        {
            if (vector[j] == valor)
            {
                valor_repetido = true;
                break;
            }
        }
        if (!valor_repetido)
        {
            vector[i] = valor;
            i++;
        }
    }
    // Imprimir los valores
    printf("Valores generados aleatoriamente para el vector 2: \n");
    for (i = 0; i < m; i++)
    {
        // Imprimir solo los valores validos (diferentes de -1)
        if (vector[i] != -1)
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
    {  
        printf("%d\n", vector[i]);  
    }  
}  
system("PAUSE");  
getchar();  
}
```

● Ejercicio 3:

```
// Ejercicio 3  
// LLENAR VECTOR 3 (CON VECTOR1 Y VECTOR2)  
void vector_3(int vector1[10], int vector2[10], int vector3[20], int m1, int m2)  
{  
    // Declarar variables  
    int i, j;  
    // Copiar elementos del vector1 al vector3  
    for (i = 0; i < m1; i++)  
    {  
        vector3[i] = vector1[i];  
    }  
    // Copiar elementos del vector2 al vector3  
    for (j = 0; j < m2; j++)  
    {  
        vector3[i + j] = vector2[j];  
    }  
    // Mostrar el vector3  
    printf("Vector 3 con los elementos de vector 1 y vector 2:\n");  
    for (i = 0; i < m1 + m2; i++)  
    {  
        printf("%d\n", vector3[i]);  
    }  
    system("PAUSE");  
    getchar();  
}
```

● Ejercicio 4:

```
// Ejercicio 4  
// IMPRIMIR VECTORES  
void imp_vector(int vector1[10], int size1, int vector2[10], int size2, int vector3[20], int size3)  
{  
    // Vector 1  
    printf("Valores del vector 1:\n");  
    for (int i = 0; i < size1; i++)  
    {  
        printf("%d\n", vector1[i]);  
    }  
}
```




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
// Vector 2
printf("Valores del vector 2:\n");
for (int i = 0; i < size2; i++)
{
    printf("%d\n", vector2[i]);
}
// Vector 3
printf("Valores del vector 3:\n");
for (int i = 0; i < size3; i++)
{
    printf("%d\n", vector3[i]);
}
system("PAUSE");
getchar();
}
```

● Ejercicio 5:

```
// Ejercicio 5
// LLENA MATRIZ 4 X 4
void llen_matriz(int matriz[4][4], int m)
{
    // Ciclo para los renglones
    for(int i = 0; i < m; i++)
    {
        // Ciclo para las columnas
        for(int j = 0; j < 4; j++)
        {
            // Llena aleatoriamente
            matriz[i][j] = rand()%100;
        }
    }
    system("PAUSE");
    getchar();
}
```

● Ejercicio 6:

```
// Ejercicio 6
// IMPRIMIR MATRIZ
void imp_matriz(int matriz[4][4], int m)
{
    // Imprimir titulo
    printf("Matriz 4x4:\n");
    // Bucle para acomodar los numeros
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < m; j++)
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
{
    // El %3d acomoda los numeros con un ancho de 3 caracteres
    printf("%3d ", matriz[i][j]);
}
// Salta a la siguiente linea despues de cada fila
printf("\n");
}
system("PAUSE");
getchar();
}
```

En conclusion, este programa me dejo muchos conocimientos respecto al uso de arreglos/vectores y matrices; me parece fundamental aprender a dominar esta funcion ya que te permite agilizar y optimizar procesos.