

## 1. DATOS INFORMATIVOS

Carrera: Electronica y Automatizacion

Asignatura: Fundamentos de Programacion

Tema del taller: Ejercicios de condiciones

Docente: Ing. Jenny Ruiz

Integrantes: Caisaguano Jefferson, Criollo Jahir, Mendoza Alexander

Fecha: 1/12/2025 Paralelo: 29583

## 2. DESARROLLO

Realizar los ejercicios enviados mediante via WhatsApp en lenguaje C

### Ejercicio 1

#### Problema 2.1.1 Vector al revés.

Desarrolle un programa que lea las N componentes de un vector y las escriba a continuación en orden inverso a la introducción. El valor de N se introduce por teclado y se supone que será igual o menor a 10.

Para la lectura del vector se utiliza una iteración con un índice que toma valores de 1 a N y se almacena en cada faceta. Para escribir se inicializa el índice a N y se decrementa en cada iteración hasta llegar en 0.

#### Codigo

```
#include <stdio.h>
```

```
int main() {  
    int N, i;  
    float vec[10];  
  
    printf("Ingrese el numero de componentes (maximo 10): ");  
    scanf("%d", &N);  
  
    if (N > 10 || N < 1) {  
        printf("Error N debe estar entre 1 y 10.\n");  
        return 1;  
    }  
}
```

```
for (i = 0; i < N; i++) {  
    printf("Ingrese el componente %d: ", i + 1);  
    scanf("%f", &vec[i]);  
}  
  
printf("Vector en orden inverso:\n");  
for (i = N - 1; i >= 0; i--) {  
    printf("%.2f ", vec[i]);  
}  
  
printf("\n");  
return 0;  
}
```

#### Corrida del programa

```
Ingrese el numero de componentes (maximo 10): 5  
Ingrese el componente 1: 2  
Ingrese el componente 2: 3  
Ingrese el componente 3: 6  
Ingrese el componente 4: 5  
Ingrese el componente 5: 8  
Vector en orden inverso:  
8.00 5.00 6.00 3.00 2.00  
  
Process returned 0 (0x0)   execution time : 9.515 s  
Press any key to continue.
```

#### Link de onlinegdb

<https://onlinegdb.com/UI2QHW8tf>

#### Ejercicio 2

##### Problema 2.1.2 Norma de un vector al cuadrado.

Desarrolle un programa que lea las **N componentes reales de un vector** y calcule:

$$m = \text{vec}^T \times \text{vec} = \sum_{i=1}^n \text{vec}(i)^2$$

El **tamaño N** se introduce por teclado y puede suponerse que es igual o menor que 10. Un bucle lee las componentes del vector y, simultáneamente, acumula la suma de sus cuadrados en la variable **res**, previamente inicializada a cero.

#### Codigo

```
#include <stdio.h>
```



```
#include <stdlib.h>
#include <conio.h>
#include <math.h>

int main()
{
    float vector[10];
    int n, i;
    float res = 0;

    printf("Escriba un numero de componenetes:");
    scanf("%d", &n);

    if (n < 1 || n > 10) {
        printf(" Error, ingrese numeros entre 1 y 10.\n");
        return 0;
    }

    for (i = 0; i < n; i++) {
        printf("Ingrese un componente %d: ", i + 1);
        scanf("%f", &vector[i]);
        res += pow(vector[i], 2);
    }

    printf("\nLa norma de un vector al cuadrado es: %.2f\n", res);
    getch();
    system("cls");
    return 0;
}
```

#### *Corrida del programa*

```
"C:\Users\G300\Desktop" X + - □ X
Escriba un numero de componenetes:3
Ingrese un componente 1: 2
Ingrese un componente 2: 3
Ingrese un componente 3: 4

La norma de un vector al cuadrado es: 29.00
|
```

#### *Link de onlinegdb*

<https://onlinegdb.com/ZNyWJdSJCj>

### Ejercicio 3

#### Problema 2.1.3 Vector con término general dado

Sea la sucesión:

$$v_k = k^2 + 3,$$

Desarrolle un programa que lea el número **n** de componentes que se quieren calcular de la sucesión, almacenándolas en un vector **vec**, tal que  $vec(i) = v_i$  mostrándolas por pantalla. Puede asumir que **n** será siempre menor o igual a 100.

Para calcular las componentes del vector se utilizará una iteración con un índice que toma valores de 1 a n en diagrama de flujo (de 0 a n-1 en C). A la vez, se irá calculando la componente ( $vec(i) = i^2 + 3$ ) y mostrándola por pantalla.

#### Código

```
#include <stdio.h>

int main() {
    int n;           // Número de componentes
    int vec[100];    // Vector con máximo 100 elementos

    printf("Ingrese el número de componentes (n <= 100): ");
    scanf("%d", &n);

    // Validar que n sea correcto
    if (n < 1 || n > 100) {
        printf("Error: n debe estar entre 1 y 100.\n");
        return 1; // Terminar el programa
    }

    // Calcular cada componente usando la fórmula  $v_k = k^2 + 3$ 
    for (int i = 0; i < n; i++) {
        vec[i] = i * i + 3; //  $i^2 + 3$ 
    }

    // Mostrar el vector
    printf("Vector generado:\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", vec[i]);
    }
    printf("\n");

    return 0;
}
```



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

#### *Corrida del programa*

```
Ingrese el numero de componentes (n <= 100): 12
Vector generado:
3 4 7 12 19 28 39 52 67 84 103 124

Process returned 0 (0x0)   execution time : 5.276 s
Press any key to continue.
|
```

#### *Link de onlinegdb*

<https://onlinegdb.com/4b28urBzmy>

### **3. CONCLUSIONES**

Mediante el uso de las condicionales if, swicht, for, while aprendimos a ejecutar e implementar ejercicios de algebra en CodeBlocks permitiendonos conocer mas a fondo los distintos usos y aplicaciones que tienen las condicionales.

### **4. RECOMENDACIONES**

Estudiar los distintos tipos de condicionales y operaciones logicas para poder aplicarlos en un lenguaje de programacion.

### **5. REFERENCIAS**