

1. DATOS INFORMATIVOS

Carrera: Electrónica y automatización

Asignatura: Fundamentos de programación

Tema del taller: Vectores

Docente: Jenny Ruiz

Integrantes: Omar Alquinga, Dylan Tutillo, Francisco Comina

Fecha: 1/12/2025 Paralelo: 29583

2. DESARROLLO

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 al introducido. 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 en diagrama de flujo (de 0 a N – 1 en C). Para la escritura se inicializa el índice a N y se decrementa en cada iteración hasta llegar a 1 (0 en C).

- **CÓDIGO DEL PROBLEMA:**

```
#include <stdio.h>

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

    printf("Introduce el numero de componentes (n <= 10): ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Introduce el componente %d: ", i + 1);
        scanf("%f", &vec[i]);
    }

    for (i = n - 1; i >= 0; i--) {
        res += vec[i];
    }

    printf("El resultado es: %f\n", res);
}
```

```
res = res + vec[i] * vec[i];  
}  
  
printf("\nLa norma al cuadrado del vector es: %.2f\n", res);  
  
return 0;  
}
```

PORGRAMA EN CODEBLOCKS:

```
#include <stdio.h>  
  
int main() {  
    int vec[10];  
    int i, N;  
  
    printf("Introduce el numero de componentes (maximo hasta el 10): ");  
    scanf("%d", &N);  
  
    if (N < 1 || N > 10) {  
        printf("Error: N debe estar entre 1 y 10.\n");  
        return 0;  
    }  
  
    for (i = 0; i < N; i++) {  
        printf("Introduce el valor %d: ", i + 1);  
        scanf("%d", &vec[i]);  
    }  
    //para inversa  
    printf("\nVector en orden inverso:\n");  
    for (i = N - 1; i >= 0; i--) {  
        printf("%d ", vec[i]);  
    }  
  
    return 0;  
}
```

- CAPTURA PROGRAMA FUNCIONANDO:

```

Introduce el numero de componentes (maximo hasta el 10): 7
Introduce el valor 1: 78
Introduce el valor 2: 34
Introduce el valor 3: 2
Introduce el valor 4: 7
Introduce el valor 5: 9
Introduce el valor 6: 10
Introduce el valor 7: 455

Vector en orden inverso:
455 10 9 7 2 34 78
Process returned 0 (0x0)    execution time : 15.250 s
Press any key to continue.

```

- LINK GDB ONLINE:

<https://onlinegdb.com/q9dwCxp4HQ>

PROBLEMA 2.1.2 NORMA DE UN VECTOR AL CUADRADO.

Desarrolle un programa que lea las n componentes reales de un vector vec y calcule:

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

El tamaño n se introduce por teclado y puede suponer 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.

- CÓDIGO DEL PROGRAMA:

```

#include <stdio.h>

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

    printf("Programa que lee las n componentes de un vector\n");
    printf("y calcula la norma al cuadrado del mismo\n\n");

    printf("Introduce n (<=10): ");

```

```

scanf("%d", &n);

if (n > 10 || n <= 0) {
    printf("Error: n debe ser entre 1 y 10.\n");
    return 1;
}

for(i = 0; i < n; i++)
{
    printf("Introduce componente %d: ", i + 1);
    scanf("%f", &vec[i]);
    res += vec[i] * vec[i];
}

printf("\nLa norma al cuadrado del vector es: %.2f\n\n", res);

return 0;
}

```

- **PROGRAMA EN CODE BLOCKS:**

```

1  #include <stdio.h>
2
3  int main(void)
4  {
5      float vec[10];
6      int i, n;
7      float res = 0;
8
9      printf("Programa que lee las n componentes de un vector\n");
10     printf("y calcula la norma al cuadrado del mismo\n\n");
11
12     printf("Introduce n (<=10): ");
13     scanf("%d", &n);
14
15     if (n > 10 || n <= 0) {
16         printf("Error: n debe ser entre 1 y 10.\n");
17         return 1;
18     }
19
20     for (i = 0; i < n; i++)
21     {
22         printf("Introduce componente %d: ", i + 1);
23         scanf("%f", &vec[i]);
24         res += vec[i] * vec[i];
25     }
26
27     printf("\nLa norma al cuadrado del vector es: %.2f\n\n", res);
28
29     return 0;
30 }
31

```

- **PROGRAMA FUNCIONANDO:**

```
C:\Users\G300\Documents\Ur X + ▾
Programa que lee las n componentes de un vector
y calcula la norma al cuadrado del mismo

Introduce n (<=10): 4
Introduce componente 1: 2
Introduce componente 2: 3
Introduce componente 3: 4
Introduce componente 4: 5

La norma al cuadrado del vector es: 54.00

Process returned 0 (0x0) execution time : 4.541 s
Press any key to continue.
|
```

- **LINK GDB ONLINE:**

<https://onlinegdb.com/J2em0yohF>

PROBLEMA 2.1.3 VECTOR CON TÉRMINO GENERAL DADO.

Sea la sucesión:

$$vk = k^2 + 3$$

desarrolle un programa que lea el número n de componentes que se quieren calcular, y almacenarlas en un vector vec, tal que $\text{vec}(i) = v_i$.

Se mostrará el vector 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 i tomando valores de 1 a n en diagrama de flujo (de 0 a n-1 en C).

A la vez, se irá calculando la componente ($\text{vec}(i) = i^2 + 3$) y mostrándola por pantalla.

- **CÓDIGO DEL PROGRAMA:**

```
#include <stdio.h>

int main() {
    int vec[100]; // vector de 100 componentes
    int n, i;

    printf("Introduce el numero de componentes (n <= 100): ");
    scanf("%d", &n);
```

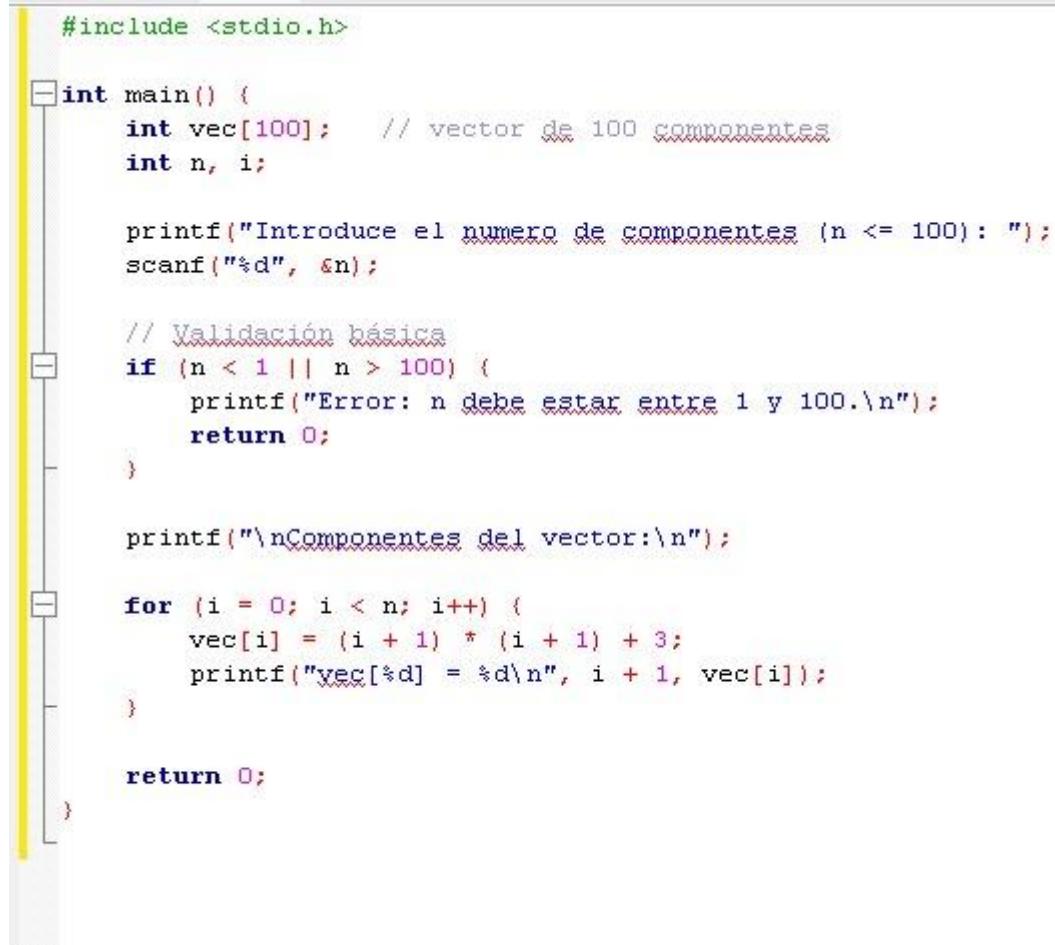
```
// Validación básica
if (n < 1 || n > 100) {
    printf("Error: n debe estar entre 1 y 100.\n");
    return 0;
}

printf("\nComponentes del vector:\n");

for (i = 0; i < n; i++) {
    vec[i] = (i + 1) * (i + 1) + 3; // vec(i) = i^2 + 3 usando índices desde 0
    printf("vec[%d] = %d\n", i + 1, vec[i]);
}

return 0;
}
```

- **PROGRAMA EN CODEBLOCKS:**



```
#include <stdio.h>

int main() {
    int vec[100]; // vector de 100 componentes
    int n, i;

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

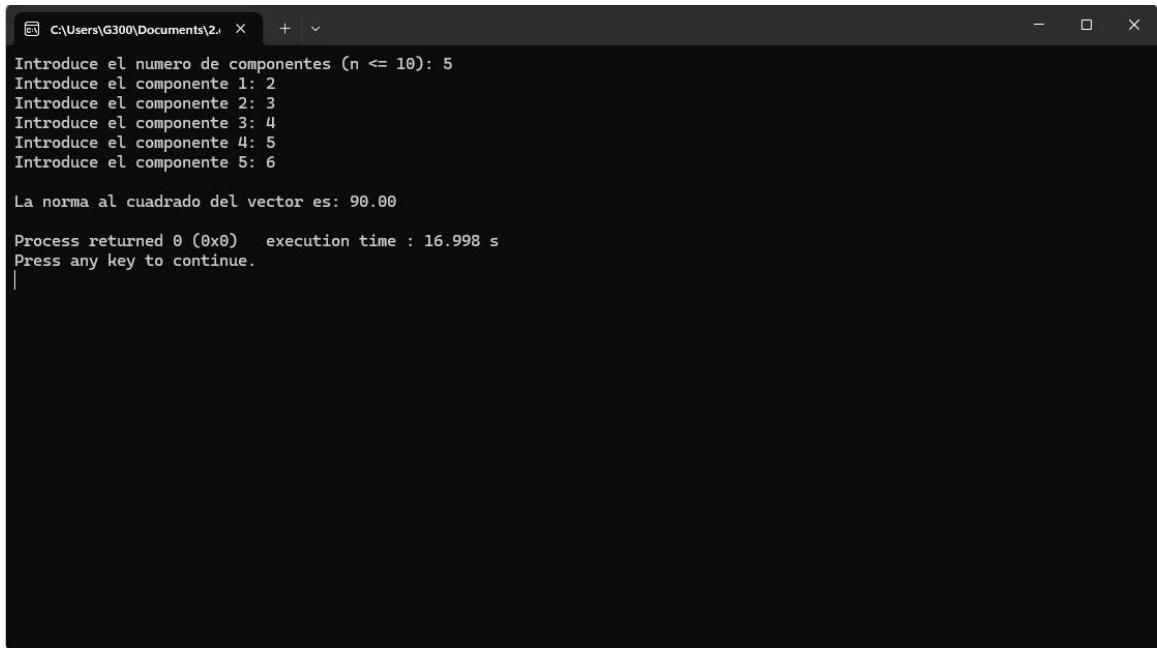
    // Validación básica
    if (n < 1 || n > 100) {
        printf("Error: n debe estar entre 1 y 100.\n");
        return 0;
    }

    printf ("\nComponentes del vector:\n");

    for (i = 0; i < n; i++) {
        vec[i] = (i + 1) * (i + 1) + 3;
        printf("vec[%d] = %d\n", i + 1, vec[i]);
    }

    return 0;
}
```

- **CAPTURA DEL PROGRAMA FUNCIONANDO:**



```
C:\Users\G300\Documents\2> Introduce el numero de componentes (n <= 10): 5
Introduce el componente 1: 2
Introduce el componente 2: 3
Introduce el componente 3: 4
Introduce el componente 4: 5
Introduce el componente 5: 6

La norma al cuadrado del vector es: 90.00

Process returned 0 (0x0)   execution time : 16.998 s
Press any key to continue.
```

- **LINK GDB ONLINE:**

https://onlinegdb.com/ZLZE_D9XN0