

# Guía

## Arreglos

Resuelva los siguientes ejercicios en lenguaje C.

**1)** Dado un número **n** entero positivo y un **listado de n números reales**, almacenar los números en un vector según el orden de entrada. Informar el contenido del vector indicando la posición ocupada por cada número a partir de la primera posición. ( $1 \leq n \leq 100$ )

**2)** Dado un **listado de números reales** del cual **no se conoce la cantidad**, almacenar los números en un vector en el orden de entrada. Informar la cantidad de números y el contenido del vector indicando la posición ocupada por cada número a partir de la primera posición. Considerar una estructura de datos de tamaño físico máximo de 1000 posiciones.

**3)** Dado un listado de valores numéricos **i** y **x**, donde **i** es un **entero mayor que 0 y menor o igual 999**, y **x** un número real, almacenar **x** en la posición **i** de un vector. El listado no se ingresa ordenado por posición. Informar la cantidad de números y el contenido del vector indicando la posición ocupada por cada número a partir de la primera posición. Omitir las posiciones que no contengan valor. Considerar una estructura de datos de tamaño físico máximo de 1000 posiciones.

**4)** Cargar una serie de números reales positivos en un vector. No se conoce la cantidad exacta de datos, pero se sabe que no superan los 100. La serie finaliza cuando se ingresa cero. Utilizar una función que invierta el contenido del vector. Informar el contenido del vector indicando la posición ocupada por cada número a partir de la primera posición

**5)** Cargar una serie de números reales positivos en un vector. No se conoce la cantidad exacta de datos, pero se sabe que no superan los 100. La serie finaliza cuando se ingresa cero. Informar el valor máximo, la cantidad de veces que aparece y la/s posición/es que ocupa.

**6)** Escribir una función en C, que reciba:

- a) como primer parámetro un vector de números enteros;
- b) como segundo parámetro, la cantidad de elementos en el vector;
- c) como tercer parámetro deberá devolver la cantidad de valores negativos que hay en el vector recibido;
- d) como cuarto parámetro, la cantidad de elementos positivos que hay en el vector recibido.

Escribir el programa que incluya a la función y las invocaciones con los siguientes ejemplos, y la respectiva impresión de los valores devueltos:

Probar el programa con los siguiente casos:

<b>v1 = [2,8,1,-5,4]</b>	<b>-&gt; Negativos: 1   Positivos: 4</b>
<b>v2 = [-2,-15,-3]</b>	<b>-&gt; Negativos: 3   Positivos: 0</b>
<b>v3 = [0,0,10,12,23,55,1]</b>	<b>-&gt; Negativos: 0   Positivos: 5</b>

**7)** Dado un vector **a** ordenado **ascendente** de longitud **ml** y un elemento **p** del mismo tipo que los elementos del vector, insertar **p** en el vector **a** de modo que siga ordenado. Validar previamente que en el vector haya espacio libre para guardar el nuevo dato.

Se solicita resolver lo solicitado recorriendo una sola vez el vector y sin utilizar un arreglo auxiliar.

Ejemplo:

nuevo elemento **p=14**

3	6	9	16	21	45	...	...
---	---	---	----	----	----	-----	-----

3	6	9	14	16	21	45	...
---	---	---	----	----	----	----	-----

**8)** Dadas las siguientes declaraciones de constantes y tipos:

```
#define MAX 1000  
typedef int vector_t[MAX]
```

Desarrollar una función en lenguaje C que reciba como **parámetros** un vector de tipo **vector\_t** ya cargado y su **máximo lógico**, que devuelva:

- un valor de verdad que informe si el vector está ordenado ascendentemente
- la menor diferencia entre elementos contiguos.

Se solicita resolver lo pedido recorriendo una sola vez al vector.

Ejemplo:

Dado el siguiente vector cuyo máximo lógico es 7, la función debe devolver **true** y **1** (la menor diferencia entre 27 y 28).

0	1	2	3	4	5	6	7	8
15	18	20	27	28	46	70	-	-

**9)** Dados 2 vectores **a** y **b** de longitudes **n** y **m**, el primero ordenado ascendente y el segundo ordenado descendente, crear un nuevo vector **c** de longitud **n + m** elementos intercalando los elementos de a y b de modo que c quede ordenado ascendente.

**10)** Dado un vector **a** de longitud **n** y un elemento **p** del mismo tipo que los elementos del vector, buscar **p** en el vector a y devolver la posición que ocupa en caso de encontrarlo o una señal en caso contrario. Suponer que no hay repeticiones.

**11)** Idem problema anterior, pero el vector está ordenado ascendente.

# Matrices

1) Dada una matriz de **3x4**, escribir un programa que permita cargar sus datos y luego mostrarlos.

2) Dada una **matriz cuadrada de 5x5** cargada por el usuario con números enteros, escribir un programa modular que muestre:

- Todos los elementos de la matriz.
- La suma de todos los elementos de la matriz.
- La suma de la diagonal principal.
- La suma de la diagonal secundaria.
- La suma de los elementos del triángulo superior.
- La suma de los elementos del triángulo inferior.
- Si es una matriz diagonal (todos los elementos por fuera de la diagonal principal son igual a cero)

Cada módulo debe optimizar la cantidad de iteraciones a realizar.

Diagonal principal

f/c	0	1	2	3	4
0	54	34	76	21	87
1	32	45	67	54	21
2	13	46	68	34	6
3	12	34	56	78	90
4	46	64	32	16	79

Diagonal secundaria

f/c	0	1	2	3	4
0	54	34	76	21	87
1	32	45	67	54	21
2	13	46	68	34	6
3	12	34	56	78	90
4	46	64	32	16	79

f/c	0	1	2	3	4
0	54	34	76	21	87
1	32	45	67	54	21
2	13	46	68	34	6
3	12	34	56	78	90
4	46	64	32	16	79



Elementos del triángulo superior



Elementos del triángulo inferior

**3)** Escribir un programa que dada **una matriz** de números **reales**, de **n x m**, devuelva un vector que en cada posición **i** almacene el elemento mínimo de cada fila **i** de la matriz.

Ejemplo:

**n=3**

**m=4**

matriz (n x m)

f/c	0	1	2	3
0	23.5	45.7	2.34	50.3
1	32.1	6.78	10.11	42.1
2	23.5	4.6	34.21	76.4

  

vector de dimensión n

2.34	6.78	4.6
0	1	2

**4)** Escribir un programa modular que dada una matriz del tipo descrito a continuación, muestre el menor número que aparece y cuantas veces aparece dicho número en la matriz.

```
#define CANT_FILAS 3
#define CANT_COLS 3
typedef int matriz_t[CANT_FILAS][CANT_COLS];
```

Ejemplo:

5	10	14
5	5	7
8	6	12

El menor número es 5 y aparece 3 veces.

**5)** Dadas dos matrices A y B cuadradas de igual dimensión, tales que  $\mathbf{A} \in \mathbb{R}^{n \times n}$  y  $\mathbf{B} \in \mathbb{R}^{n \times n}$ , escribir un programa que permita cargar su dimensión y sus datos. El programa debe sumarlas y mostrar la matriz resultante. Para la estructura de datos considere un tamaño máximo de 10 ( $1 \leq n \leq 10$ ).

**6)** Dadas dos matrices A y B, tales que  $\mathbf{A} \in \mathbb{R}^{m \times n}$  y  $\mathbf{B} \in \mathbb{R}^{q \times p}$ , escribir un programa que permita cargar sus dimensiones y sus datos. El programa debe multiplicarlas ( $\mathbf{A} \cdot \mathbf{B}$ ) en caso de ser posible o devolver una señal si la operación no puede realizarse. Para la estructura de datos considere un tamaño máximo de 10 (**m**, **n** y **p** pueden tomar valores entre 1 y 10).

Aclaración: para multiplicar matrices, la primera matriz debe tener igual columnas que filas tenga la segunda ( $n = q$ ).

La matriz resultante será  $\mathbf{C} \in \mathbb{R}^{m \times p}$

Ejemplo:

Matrices	Resultado
<b>3 2 4    1 2</b>	<b>{29, 38}</b>
<b>5 6 7    3 4</b>	<b>{58, 76}</b>
<b>          5 6</b>	

**7)** Se tiene una colección de datos de la temperatura por hora (**0 a 23**) por día (**1 a 31**) del mes de marzo.

Se pide:

- a) Cargar el listado en una matriz e informarla en forma bidimensional.
- b) Informar además la máxima y la mínima temperatura por día, la máxima y la mínima del mes.

**8)** Se tiene un listado con los siguientes datos:

- número de alumno ( 1 a n )
- número de materia ( 1 a m )
- nota promedio ( 0 a 10 )

El mismo número de alumno y de materia puede aparecer más de una vez.

El listado no está ordenado, ni necesariamente completo. Esto último quiere decir que puede ser que un alumno no haya cursado una o más materias, y por lo tanto no existan los datos correspondientes en el listado.

Se pide :

- a) Crear una estructura bidimensional que almacene el promedio por materia de cada alumno e informarla asignándole en la impresión un guión al caso de falta de datos mencionado.
- b) Informar el porcentaje de alumnos que cursó cada materia y el promedio general por materia considerando los alumnos que la cursaron.
- c) Informar la cantidad de materias que cursó cada alumno y el promedio que obtuvo considerando las materias que cursó.