



DEBER

Carrera: Electrónica y Automatización

Asignatura: Fundamentos de Programación

Docente: Jenny Ruiz

Estudiante: Jhosue Angamarca

Fecha: 08/12/2025

40 100 Problemas resueltos de programación en lenguaje C para ingeniería

2.2. Problemas de matrices en C

Problema 2.2.1 Suma de componentes de una matriz.
Halle la suma de las componentes de una matriz, cuya dimensión y componentes se han de leer por teclado y por filas. Suponga que el número de filas N y columnas M son menores o iguales a 10.

Para leer una matriz por filas lo más sencillo es usar un doble bucle anidado. El bucle interior lee todas las columnas para una fila dada. El bucle externo va cambiando la fila que hay que leer. A medida que se leen las componentes se irán acumulando en la variable *res*, la cual se debe iniciar a cero. Asimismo, observe que en el diagrama de flujo el índice de las filas se establece desde 1 al número de filas y el índice de las columnas desde 1 al número de columnas; sin embargo, en C los índices de vectores y matrices comienzan en 0.

Codificación en C del problema 2.2.1

<https://onlinegdb.com/w4lGsxkuz>

Problema 2.2.2 Escritura de matriz en sentido inverso.
Dada una matriz de $N \times N$ elementos, realice un algoritmo que recorra la matriz por filas desde la última a la primera y cada fila en sentido inverso, y de la primera columna a la primera, de modo que se vaya mostrando cada elemento.

La solución a este problema consiste en recorrer la matriz invirtiendo el sentido habitual de los bucles. Observe cómo, en este caso, los bucles de filas y columnas las variables *i* y *j* comienzan en la última fila/columna de la matriz. La condición de permanencia en los bucles es ahora $i \geq 1$ o $j \geq 1$ (en C $i \geq 0$ o $j \geq 0$) y las variables se decrementan en cada iteración.

A continuación se muestra el diagrama de flujo de la solución en la figura 2.2.2, así como la tabla de objetos y codificación en C³.

Codificación en C del problema 2.2.2

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

<https://onlinegdb.com/NJGyS4MFY>



Problema 2.2.3 Máximo de una fila.
Escriba un programa que lea una matriz de N filas y N columnas de valores enteros. A continuación, el programa debe pedir el número de una fila y mostrar por pantalla el valor de la mayor componente de esa fila.

Tal como ya se discutió en el problema 1.7, la dificultad de calcular el máximo valor de un vector (en este caso un vector fila de una matriz) reside en decidir qué valor inicial se le da a la variable que va a almacenar el máximo (max). Imagine que se asume que todos los números del vector son positivos y se inicializa $max = -1000$. Se procede entonces a comparar este valor con todas las componentes del vector y, si alguna es mayor, se actualiza el valor de max con el valor de esa componente. Podría ocurrir, sin embargo, que todas las componentes del vector sean menores que -1000 , en cuyo caso el valor del máximo calculado sería erróneamente -1000 .

Una forma sencilla de solucionar este problema es simplemente iniciar el valor de max con el valor de la primera componente del vector (cualquier componente del vector valdría en realidad para inicializar), y proceder a continuación con las comparaciones como se ha indicado. De este modo no se fuerza a ninguna suposición sobre el rango de valores donde se encuentran las componentes del vector.

Se muestra a continuación la solución planteada en el diagrama de flujo de la figura 2.12, junto con su tabla de objetos y codificación en C⁴.

<https://onlinegdb.com/MGyAclIw>

46

100 Problemas resueltos de programación en C

Problema 2.2.4 Intercambiar las filas i, j de una matriz.
Escriba un programa que intercambie las filas i y j de una matriz de enteros de $N \times N$ componentes, siendo i y j dos valores introducidos por teclado.

La solución se muestra en el diagrama de la figura 2.13, junto con su tabla de objetos y codificación⁵. La dificultad del problema reside en intercambiar las filas sin perder información, tal y como pasa aquí al intercambiar el elemento 7 de las filas a y b :
`mat[a][6]=mat[b][6]; //Sobrescribe mat[a][6] y se pierde su valor`
`mat[b][6]=mat[a][6]; // Queda mat[a][6] con el mismo valor que mat[b][6]`

Es necesario usar una variable auxiliar para no perder el valor original de `mat[a][6]`
`aux=mat[a][6];`
`mat[a][6]=mat[b][6];`
`mat[b][6]=aux;`

Codificación en C del problema 2.2.4

<https://onlinegdb.com/WEz65Ay5X>