

COMPUTACION

[Área personal](#) / [Mis cursos](#) / [Cursos Basicos](#) / [2o Semestre](#) / [Compu](#) / [Semana 21 - FINAL 1](#)
/ [Tema 3 - 1er Final - Fila 2](#)

Descripción

[Ver entrega](#)

Tema 3 - 1er Final - Fila 2

Disponible desde: miércoles, 2 de diciembre de 2020, 10:30

Límite de entrega: miércoles, 2 de diciembre de 2020, 11:50

Ficheros requeridos: t3F1F2.cpp ([Descargar](#))

Tipo de trabajo: Individual

Universidad Nacional de Asunción - Facultad de Ingeniería

CATEDRA DE COMPUTACIÓN - EXAMEN FINAL 1 - 02/dic/2020 - TEMA 3

Fila 2

Escribir una función en C++ llamada [contarApariciones](#) que:

- Lea (por teclado) las dimensiones de una matriz **A** (de **m** filas y **n** columnas) de números enteros.
- Lea (por teclado) los elementos de la matriz **A**.
- Lea (por teclado) las dimensiones de otra matriz **B** (de **p** filas y **q** columnas) de números enteros.
- Lea (por teclado) los elementos de la matriz **B**.
- Retorne la cantidad de veces que **B** está contenido en **A**. Si no existe la posibilidad de que **A** esté contenida en **B** (por las dimensiones de ambas), la función debe retornar **-1**.

Nota : Las dimensiones de las matrices **A** y **B** deben validarse (es decir, asegurarse de que sean números enteros y positivos). Esto se hará a través de la función [int validarDim\(\)](#), que deberá implementarse. No hace falta validar que los elementos ingresados para las matrices **A** y **B** sean números enteros.

Algunos ejemplos son los siguientes:



A

1	3	4
4	7	8
2	2	4
3	5	7

B

4	7
2	2

B está contenido 1 vez en A

A

5	3	1	2
4	5	3	2
4	4	5	9
4	5	3	2
3	4	5	4

B

5	3
4	5

B está contenido 3 veces en A

A

4	7
2	2

B

1	3	4
4	7	8
2	2	4
3	5	7

La función debe retornar **-1** en este caso

Una función útil para resolver este ejercicio (que también debe implementarse) es:

`int cantidadDeCoincidencias(int *b, int *c, int m, int n)` , que recibe la direcciones de los elementos en la posición `[0][0]` de dos matrices (ambas de tamaño `m * n` y contienen números enteros), y devuelve la cantidad de elementos (correspondientes) que son iguales. Por ejemplo:

A

4	7
2	2

B

4	7
3	2

La función debe retornar **3**

OTRAS DIRECTIVAS A SER NECESARIAMENTE TENIDAS EN CUENTA:

- **NO MODIFICAR LA FUNCIÓN PRINCIPAL** (`main`). **LA MODIFICACIÓN SERÁ PENALIZADA** , pues puede alterarse la secuencia de lecturas en la entrada y la salida esperada para el programa.
- **DEFINIR NECESARIAMENTE LA FUNCIÓN QUE REALICE LO SOLICITADO** . No se aceptará que la funcionalidad se desarrolle en el `main`.
- **SE PROPORCIONA la función** `void leerElementosMatriz(int *x, int y, int z)` , que carga los elementos de una matriz (de `y` filas y `z` columnas). El puntero `x` contiene la dirección del elemento inicial `[0][0]` de la matriz.
- **EN EL VPL**, en la función `contarNroApariciones` se encuentra una sección inicial de código que carga las dimensiones y elementos de las matrices `A` y `B` , utilizando la implementación de `validarDim` y la función auxiliar `leerElementosMatriz` . Se recomienda no modificar esa sección.
- **LA SOLUCIÓN DEBE SER LO SUFICIENTEMENTE GENERAL** para resolver cualquier entrada diferente a las propuestas en los ejemplos y los casos de prueba . Por lo tanto, la calificación mostrada por el sistema no necesariamente corresponderá con la calificación final.



• La distribución de puntajes para cada función es la siguiente:

- Función `validarDim` : **10%**
- Función `cantidadDeCoincidencias` : **40%**
- Función `contarApariciones` : **50%**

Ficheros requeridos

t3F1F2.cpp

```

1  #include<iostream>
2  using namespace std;
3
4  /*Función auxiliar que carga los elementos de una matriz (de "y" filas y "z" columnas).
5  El puntero x contiene la dirección del elemento inicial [0][0] */
6  void leerElementosMatriz(int *x, int y, int z){
7      int i,j;
8      for(i=0;i<y;i++) for(j=0;j<z;j++) cin>>*(x+i*z+j);
9  }
10
11 /*Esta función debe leer números por teclado, tantas veces sea necesario, de tal manera a asegurarse de que el mismo
12 sea entero y positivo. Retornar el número que cumpla con esta condición. */
13 int validarDim(){
14     //Comience aquí su implementación
15 }
16
17
18 /*Esta función recibe las direcciones de los elementos en [0][0] de dos matrices B y C (de las mismas dimensiones - m fila
19 y devuelve la cantidad de elementos (correspondientes) que son iguales.*/
20 int cantidadDeCoincidencias(int *b, int *c, int m, int n){
21     //Comience aquí su implementación
22 }
23
24 }
25
26 /*La sección inicial de código para esta función se encarga de llamar a validarDim() para cargar las dimensiones de las ma
27 y a leerElementosMatriz() para cargar sus elementos.
28
29 La función debe retornar la cantidad de apariciones de B en A.*/
30
31 int contarNroApariciones(){
32     /*---NO MODIFICAR ESTA SECCIÓN---*/
33     int m=validarDim(); //cantidad de filas de A
34     int n=validarDim(); // cantidad de columnas de A
35     int A[m][n]; // matriz A de m filas y n columnas
36     leerElementosMatriz(&A[0][0],m,n); //lectura de los elementos de A
37     int p=validarDim(); //cantidad de filas de B
38     int q=validarDim(); // cantidad de columnas de B
39     int B[p][q]; // matriz B de p filas y q columnas
40     leerElementosMatriz(&B[0][0],p,q); //lectura de los elementos de B
41     /*-----FIN DE LA SECCION-----*/
42
43     //Comience aquí su implementación
44 }
45
46 }
47
48 int main(){
49     /*---NO MODIFICAR EL MAIN*/
50     int opcion;
51     cin>>opcion;
52     if(opcion==1){
53         cout<<validarDim()<<endl;
54     }
55     if(opcion==2){
56         int m=validarDim(); //cantidad de filas de A
57         int n=validarDim(); // cantidad de columnas de A
58         int A[m][n]; // matriz A de m filas y n columnas
59         leerElementosMatriz(&A[0][0],m,n); //lectura de los elementos de A
60         int p=validarDim(); //cantidad de filas de B
61         int q=validarDim(); // cantidad de columnas de B
62         int B[p][q]; // matriz B de p filas y q columnas
63         leerElementosMatriz(&B[0][0],p,q); //lectura de los elementos de B
64         if(opcion==2&&m==p&&n==q) cout<<cantidadDeCoincidencias(&A[0][0],&B[0][0],p,q)<<endl;
65     }
66     if(opcion==3) cout<<contarNroApariciones()<<endl;
67     return 0;
68 }
```

◀ Final 1 - Tema 2 - Fila 2

Ir a...

Programa de estudio semana 1 ▶

