

Área personal ► Mis cursos ► InfoC++ ► Clase 10 ► TP3

<b>Comenzado el</b>	jueves, 18 de octubre de 2018, 18:39
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	sábado, 10 de noviembre de 2018, 23:58
<b>Tiempo empleado</b>	23 días 5 horas
<b>Calificación</b>	15 de 20 (75%)

Escriba un programa que como primer paso pida con un "cin" que el usuario ingrese un número entero entre 1 y 10, el cual va a representar la cantidad de elementos de un vector de números enteros. Luego de ingresar este número, el programa debe pedir nuevamente mediante "cin" que se ingresen todos los números del vector. En ningún caso es necesario que el programa imprima un mensaje mediante "cout". Cuando el vector este completo, el programa debe calcular un nuevo vector de igual longitud que el vector ingresado por el usuario y en donde el elemento en la posición 'k' del nuevo vector es la suma de todos los números del vector que ingreso el usuario excepto el numero de la posición 'k' de dicho vector. Finalmente el programa debe imprimir por pantalla los números del nuevo vector separados por un espacio.

Ejemplo

Si los números de entrada al programa son:

4 1 2 3 4

Entonces el resultados del programa es:

9 8 7 6

Notar que el primer número de entrada '4' indica que el vector va a tener 4 elementos que en este caso es 1, 2, 3 y 4. Luego el vector resultante es 9, 8, 7 y 6 donde  $9 = 2+3+4$ ,  $8 = 1+3+4$ ,  $7 = 1+2+4$  y  $6 = 1+2+3$

**Respuesta:** (penalty regime: 0 %)

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main()
6  {
7      int suma;
8      int n;
9      int vector[10]; // puse el tamaño mas grande del vector
10
11      do
12      {
13          cin>>n;
14      }
15      while(n>10||n<0);
16
17
18      for(int i=0;i<n;i++)
19      {
20          cin>>vector[i];
21      }
22
23      for(int i=0;i<n;i++)
24      {
25          suma=0;
26          for(int k=0;k<n;k++)
27          {
28              if(k!=i)
29              {
30                  suma += vector[k];
31              }
32          }
33          cout<<suma<<" ";
34      }
35      return 0;
36  }
37

```

	Input	Expected	Got
✓	3 1 2 3	5 4 3	5 4 3
✓	7 2 3 5 4 9 7 3	31 30 28 29 24 26 30	31 30 28 29 24 26 30
✓	10 1 2 3 4 5 6 7 8 9 10	54 53 52 51 50 49 48 47 46 45	54 53 52 51 50 49 48 47 46 45

	Input	Expected	Got	http://lev2.efn.uncor.edu/mod/quiz/review.php?a...
✓	1 5	0	0	

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

## Pregunta 2

Correcta

Puntúa 1 sobre 1

Complete el siguiente programa para que almacene en el arreglo fibonacci[ ] una sucesión de fibonacci. La sucesión comienza con los números 0 y 1, y a partir de estos, cada término es la suma de los dos anteriores. Es decir, que el primer elemento (índice 0) deberá contener el valor 0, el segundo elemento el 1, y a partir del tercer elemento, la suma de los dos anteriores.

**Respuesta:** (penalty regime: 0 %)

Reiniciar respuesta

```

1 // Bibliotecas
2 #include<iostream>
3 #include<cmath>
4 using namespace std;
5
6 int main()
7 {
8     int N;
9     cin >> N;
10    int fibonacci[N];
11
12    //Completar con su codigo
13    fibonacci[0]=0;
14    fibonacci[1]=1;
15    for(int i=2;i<N;i++)
16    {
17        fibonacci[i]=fibonacci[i-1]+fibonacci[i-2];
18    }
19
20    for(int i=0; i<N; i++)
21        cout << fibonacci[i] << " ";
22
23    return 0;
24 }
25

```

	Input	Expected	Got	
✓	5	0 1 1 2 3	0 1 1 2 3	✓
✓	1	0	0	✓
✓	10	0 1 1 2 3 5 8 13 21 34	0 1 1 2 3 5 8 13 21 34	✓
✓	4	0 1 1 2	0 1 1 2	✓

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Realice un programa el cual solicite al usuario el ingreso de un valor entero  $t$  que será el tamaño de los arreglos a procesar.

Una vez ingresado el tamaño  $t$  de los arreglos, solicitar al usuario el ingreso de los  $t$  elementos enteros de DOS diferentes arreglos declarados como `array1` y `array2`. Los valores a ingresar por el usuario deben ser mayores o iguales a 0 y menores o iguales a 10. Caso contrario solicitar reingreso hasta que el valor esté dentro de rango.

Ej: Ingrese el tamaño de los arreglos a procesar: 4

Ahora ingrese los 4 elementos del arreglo 1:

1  
2  
3  
4

Ahora ingrese los 4 elementos del arreglo 2:

5  
6  
7  
8

Una vez cargados `array1` y `array2`; el programa deberá determinar un arreglo resultante de operaciones de potencia. Con `array1` utilizado como las bases y `array2` como los exponentes.

Por ejemplo con los **valores ingresados anteriormente** su programa debe determinar e imprimir en pantalla cada componente del arreglo exponente como:

`exponente[i]=array1[i]array2[i]`

**Ej. Salida:**

**1 64 2187 65536**

**Respuesta:** (penalty regime: 0 %)

```

1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5  int main()
6  {
7      int t;
8      cin >> t;
9      int array1[t], array2[t], exponente [t] ;    // declare exponente
10
11  for(int i=0;i<t;i++)
12  {
13      do
14      {
15          cin>>array1[i];
16      }
17      while(array1[i]<0||array1[i]>10);
18  }
19
20  for(int j=0;j<t;j++)
21  {
22      do
23      {
24          cin>>array2[j];
25      }
26      while(array2[j]<0||array2[j]>10);
27  }
28
29  for(int k=0;k<t;k++)
30  {
31      exponente[k]=pow(array1[k],array2[k]);
32      cout<<exponente[k]<<" ";
33  }
34
35
36  return 0;
37  }
```

	Input	Expected	Got	
--	-------	----------	-----	--

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Realice un programa el cual solicite al usuario el ingreso de un valor entero **t** que será el tamaño de los arreglos a procesar.

Una vez ingresado el tamaño **t** de los arreglos, solicitar al usuario el ingreso de los **t** elementos enteros de DOS diferentes arreglos declarados como **array1 y array2 (respete nombres)**.

Los valores a ingresar por el usuario deben ser enteros PARES mayores o iguales a 0 y menores o iguales a 50. Caso contrario solicitar reingreso hasta valor dentro de rango.

Ejemplo:

Ingrese cantidad de elementos: 4

Ingrese 4 elementos del array1:

2

6

24

12

Ingrese 4 elementos del array2:

52

Error, reingrese dentro de rango solicitado.

2

4

8

36

El programa deberá determinar e imprimir un mensaje indicando el **nombre del arreglo** que contenga la mayor cantidad de elementos mayores comparando elemento a elemento los arreglos (en caso de ser iguales los elementos no se consideran).

Si al finalizar la comparación ambos arreglos tienen igual cantidad de elementos mayores al otro arreglo imprimir mensaje **"EMPATE"**.

Ej. con los valores cargados anteriormente:

**SALIDA.**

**El arreglo con mayor cantidad de elementos mayores es el array1.**

Aclaración con los datos ingresados en el ejemplo el calculo:

Array1[0] = Array2[0]

Array1[1] > Array2[1]

Array1[2] > Array2[2]

Array1[3] < Array2[3]

El arreglo con más elementos mayores es el array1.

**Respuesta:** (penalty regime: 0 %)

```

1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5  int main()
6  {
7      int cuenta1=0, cuenta2=0;
8      int t;
9      cin >> t;
10     int array1[t], array2[t] ;
11
12     for(int i=0;i<t;i++)
13     {
14         do
15         {
16             cin>>array1[i];
17         }
18         while(array1[i]<0||array1[i]>50||array1[i]%2==1);
19     }
20

```

	Input	Expected	Got	
✓	4 2 6 24 12 52 2 4 8 36	array1	array1	✓
✓	3 0 0 0 2 4 6	array2	array2	✓

	Input	Expected	Got	
✓	2 2 -8 6 51 6 33 2	EMPATE	EMPATE	✓
✓	5 16 32 24 8 4 0 0 0 0 0  5 -8 1	array1	array1	✓
✓	4 10 10 10 10 10 10 10 50	array2	array2	✓

Todas las pruebas superadas. ✓

**Correcta**  
Puntos para este envío: 1/1.



TP3 **Pregunta 5**

Incorrecta

Puntúa 0 sobre 1

Codifique la definición de una función cuyo prototipo es: <http://lev2.efn.uncor.edu/mod/quiz/review.php?a...>

**void eliminarRepetidos(int conRepetir[], int N, int sinRepetir[], int& M);**

La función recibe como argumento un arreglo (conRepetir[]) de enteros de tamaño indeterminado y un entero N que representa la cantidad de elementos en el arreglo. A su vez recibe como argumento un arreglo de tamaño indeterminado y un entero por referencia M que representa la cantidad de elementos en el arreglo sinRepetir[].

La función debe copiar los elementos del arreglo conRepetir[] al arreglo sinRepetir[] pero sin repetir valores a su vez debe almacenar en el argumento pasado por referencia M la cantidad de elementos sin repetir.

**Respuesta:** (penalty regime: 0 %)

Reiniciar respuesta

```

1 void eliminarRepetidos(int conRepetir[], int N, int sinRepetir[], int &M)
2 {
3     int l;
4     sinRepetir[0]=conRepetir[0];
5     M=0;
6     for(int i=1;i<N;i++)
7     {
8         l=0;
9         for(int j=0;j<i;j++)
10        {
11            if(sinRepetir[j]==conRepetir[i])
12            {
13                l++;
14            }
15        }
16        if(l==0)
17        {
18            M++;
19            sinRepetir[M]=conRepetir[i];
20        }
21    }
22
23    return;
24 }
```

	Test	Expected	Got	
✗	<pre>const int N = 3; int A[N]={1,2,3}; int B[N]; int M; eliminarRepetidos(A,N,B,M); for(int i=0;i&lt;M;i++) {     cout&lt;&lt;B[i]&lt;&lt;endl; } cout&lt;&lt;M;</pre>	<pre>1 2 3 3</pre>	<pre>1 2 2</pre>	✗
✗	<pre>const int N = 4; int A[N]={1,2,3,2}; int B[N]; int M; eliminarRepetidos(A,N,B,M); for(int i=0;i&lt;M;i++) {     cout&lt;&lt;B[i]&lt;&lt;endl; } cout&lt;&lt;M;</pre>	<pre>1 2 3 3</pre>	<pre>1 2 2</pre>	✗

	Test	Expected	Got	
✖	<pre>const int N = 5; int A[N]={1,2,100,1,3}; int B[N]; int M; eliminarRepetidos(A,N,B,M); for(int i=0;i&lt;M;i++) {     cout&lt;&lt;B[i]&lt;&lt;endl; } cout&lt;&lt;M;</pre>	1 2 100 3 4	1 2 100 3	✖
✖	<pre>const int N = 10; int A[N]={1,2,3,5,1,3,5,8,1,2}; int B[N]; int M; eliminarRepetidos(A,N,B,M); for(int i=0;i&lt;M;i++) {     cout&lt;&lt;B[i]&lt;&lt;endl; } cout&lt;&lt;M;</pre>	1 2 3 5 8 5	1 2 3 5 4	✖

Some hidden test cases failed, too.  
Your code must pass all tests to earn any marks. Try again.

Mostrar diferencias

**Incorrecta**  
Puntos para este envío: 0/1.

Escriba un programa que como primer paso pida con un "cin" que el usuario ingrese un número entero entre 1 y 10, el cual va a representar la cantidad de elementos de un vector de números enteros. Luego de ingresar este número, el programa debe pedir nuevamente mediante "cin" que se ingresen todos los números del vector. En ningún caso es necesario que el programa imprima un mensaje mediante "cout". Este procedimiento se debe repetir dos veces para que el usuario pueda ingresar dos vectores. Cuando los dos vectores estén completos, el programa debe calcular un nuevo vector que contenga los números que tienen en común los dos vectores ingresados por el usuario (sin importar el orden en que se encuentren dichos números dentro de estos dos vectores). El nuevo vector no debe contener números repetidos y los números deben respetar en orden en que se encontraron en el primer vector ingresado. Finalmente el programa debe imprimir por pantalla los números del nuevo vector separados por un espacio.

Ejemplo

Si los números de entrada al programa son:

4 4 3 2 1

y

5 1 1 2 2 3

Entonces el resultados del programa es:

3 2 1

**Respuesta:** (penalty regime: 0 %)

```

1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5  int main()
6  {
7      int t1, t2, n;
8      int vector1[t1], vector2[t2], vector[n];
9
10     do
11     cin >> t1;
12     while(t1<0||t1>10);
13
14     for(int i=0;i<t1;i++)
15     cin>> vector1[i];
16
17     do
18     cin >> t2;
19     while(t2<0||t2>10);
20
21     for(int j=0;j<t2;j++)
22     cin>> vector2[j];
23
24     //no debe contener números repetidos y los números deben respetar el orden en que se
25     //encontraron en el primer vector ingresado
26
27     for(int i=0;i<t1;i++)
28     {
29         for(int j=0;j<t2;j++)
30         {
31             while(vector1[i]==vector2[j])
32             {
33                 for(int k=0;k<n;k++)
34                 {
35                     vector[k]=vector1[i];
36                 }
37             }
38         }
39     }
40
41     for(int k=0;k<n;k++)
42     {
43         cout<<vector[k]<<" ";
44         cout<<endl;
45     }

```

**Syntax Error(s)**

```
prog.cpp: In function 'int main()':
prog.cpp:8:41: error: 'n' is used uninitialized in this function [-Werror=uninitialize
d]
    int vector1[t1], vector2[t2], vector[n];
                                   ^
cc1plus: all warnings being treated as errors
```

**Incorrecta**

Puntos para este envío: 0/1.

Realice un programa el cual declare una matriz de 3 filas x 3 columnas de valores de doble precisión.

Luego solicite al usuario el ingreso de los 9 valores de doble precisión, el programa debe cargárselos en la matriz por **COLUMNA**.

Ejemplo de ejecucion:

Ingresa los 9 valores de la matriz:

1.2

2.3

3.5

6.5

4.3

6.7

8.9

0.9

2.2

La Matriz debe quedar:

1.2 6.5 8.9

2.3 4.3 0.9

3.5 6.7 2.2

Una vez cargado los datos en la matriz, el programa deberá calcular el promedio de todos los valores de la matriz (Para el ejemplo el promedio es 4.05)

Una vez determinado el promedio de la matriz, el programa debe convertir la matriz ingresada en una matriz triangular superior.

Aclaracion: Matriz triangular superior: dada una matriz cuadrada, una matriz triangular superior tiene todos los elementos por debajo de la diagonal principal nulos.

Ej. matriz triangular superior de la matriz anterior cargada:

1.2 6.5 8.9

0.0 4.3 0.9

0.0 0.0 2.2

Finalmente el programa debe imprimir en pantalla cuantos valores de la matriz triangular superior obtenida son mayores al promedio.

**Ej. con los datos cargados:**

**SALIDA**

**3**

**Respuesta:** (penalty regime: 0 %)

```
1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5  int main()
6  {
7      int n=3;
8      double matriz[n][n];
9      double sum=0.0, prom;
10     int cuenta=0;
11
12     //el programa debe carga en la matriz por COLUMNA
13     for(int j=0;j<n;j++)
14     {
15         for(int i=0;i<n;i++)
16         {
17             cin>>matriz[i][j];
18         }
19     }
20
21     for(int j=0;j<n;j++)
22     {
23         for(int i=0;i<n;i++)
24         {
```

	Input	Expected	Got	
✓	1.2 2.3 3.5 6.5 4.3 6.7 8.9 0.9 2.2	3	3	✓
✓	0 0 0 0 0 0 0 0 0	0	0	✓
✓	1 20 30 2 3 40 5 6 7	0	0	✓

	Input	Expected	Got	
✓	20 -1 -5 50 40 -30 20 26 30	6	6	✓
✓	0 -1 -1 1 0 -1 1 1 0	3	3	✓

Todas las pruebas superadas. ✓

**Correcta**  
Puntos para este envío: 1/1.

Escriba un programa que como primer paso pida con un "cin" que el usuario ingrese un número entero entre 1 y 10, el cual va a representar la cantidad de filas y columnas de una matriz cuadrada. Luego de ingresar este número, el programa debe pedir nuevamente mediante "cin" que se ingresen todos los números de la matriz fila por fila. En ningún caso es necesario que el programa imprima un mensaje mediante "cout". Cuando la matriz este completa, el programa debe calcular una nueva matriz del mismo tamaño que la matriz ingresada por el usuario pero que en la fila 'k' y columna 'n' la nueva matriz tenga un '1' o un '0'. El valor '1' se asigna si el mayor elemento de la fila 'k' de la matriz ingresada por el usuario es mayor que el mayor elemento de la columna 'n' de dicha matriz. De lo contrario, la nueva matriz tiene un '0' en la posición mencionada. Luego de calcular la nueva matriz, el programa debe imprimir todos los números fila por fila separados por un espacio como se muestra en el ejemplo que esta a continuación.

### Ejemplo

Si la entrada de datos es:

```
2 1 2 3 4
```

El resultado de salida debe ser:

```
0 0 1 0
```

Notar que el primer número de la entrada indica que se trata de una matriz de 2 filas y dos columnas, los siguientes 4 números indican los valores de dicha matriz, esto es:

```
1 2
```

```
3 4
```

La matriz resultante es:

```
0 0
```

```
1 0
```

Por esta razón la salida del programa debe ser: 0 0 1 0

Notar que la posición [1][0] de la matriz resultante es la única que vale '1' porque el máximo de la fila 1 es 4 = max(3,4) mientras que el máximo de la columna 0 es 3 = max(1,3)

**Respuesta:** (penalty regime: 0 %)

```
1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4 int maxf(int,int,int);
5 int maxc(int,int,int);
6
7
8 int main()
9 {
10     int n;
11     int matriz[n][n];
12
13     do
14     {
15         cin>>n;
16     }
17     while(n<1||n>10);
18     //ingreso la matriz fila a fila
19     for(int i=0;i<n;i++)
20     {
21         for(int j=0;j<n;j++)
22         {
23             cin>> matriz[i][j];
24         }
25     }
26     //calculo una nueva matriz que en la fila 'k' y columna 'n' tiene un '1' o un '0'.El valor
27     // se asigna si el mayor elemento de la fila 'k' de la matriz ingresada es mayor que
28     //el mayor elemento de la columna 'n' de dicha matriz. De lo contrario, la nueva matriz tien
29     for(int m=0;m<n;m++)
30     {
31         for(int k=0;k<n;k++)
32         {
33             if(maxc(matriz[][m], n) < maxf( matriz[k][], k, n))
34                 matriz[k][m]=1;
35             else
36                 matriz[k][m]=0;
37         }
```



**Syntax Error(s)**

```

prog.cpp: In function 'int main()':
prog.cpp:33:35: error: expected primary-expression before ']' token
                if(maxc(matriz[][],m, n) < maxf( matriz[][], k, n))
                ^
prog.cpp:33:37: error: expected primary-expression before ']' token
                if(maxc(matriz[][],m, n) < maxf( matriz[][], k, n))
                ^
prog.cpp:33:60: error: expected primary-expression before ']' token
                if(maxc(matriz[][],m, n) < maxf( matriz[][], k, n))
                ^
prog.cpp:33:62: error: expected primary-expression before ']' token
                if(maxc(matriz[][],m, n) < maxf( matriz[][], k, n))
                ^
prog.cpp:34:28: error: name lookup of 'i' changed for ISO 'for' scoping [-fpermissive]
                matriz[i][j]=1;
                ^
prog.cpp:34:28: note: (if you use '-fpermissive' G++ will accept your code)
prog.cpp:34:31: error: 'j' was not declared in this scope
                matriz[i][j]=1;
                ^
prog.cpp:36:31: error: 'j' was not declared in this scope
                matriz[i][j]=0;
                ^

prog.cpp: At global scope:
prog.cpp:58:23: error: declaration of 'matriz' as multidimensional array must have bounds for all dimensions except the first
int maxf(int matriz[][],int k, int n)
                ^
prog.cpp:58:24: error: expected ')' before ',' token
int maxf(int matriz[][],int k, int n)
                ^
prog.cpp:58:25: error: expected unqualified-id before 'int'
int maxf(int matriz[][],int k, int n)
                ^

```

**Incorrecta**

Puntos para este envío: 0/1.

Codifique una función cuyo prototipo es:

```
void matrizNueva (float original[][N], float nueva[][N], float promedios[N]);
```

donde N es una constante definida globalmente. La función recibe como argumento dos matrices cuadradas y un arreglo de elementos float.

Primero la función debe calcular los promedios de cada columna de la matriz "original" y almacenarlos en el arreglo "promedios".

Ej:

Matriz original

1.1 2.2 3.3

4.4 5.5 6.6

7.7 8.8 9.9

Arreglo promedios

4.4 5.5 6.6

Una vez almacenado los promedios en el arreglo, la función debe cargar los datos en la matriz "nueva" según corresponda:

Los elementos de la primera columna de la matriz "original" se deben comparar con el primer elemento del arreglo "promedios", los elementos de la segunda columna de la matriz se debe comparar con el segundo elemento del arreglo y así sucesivamente.

Si al comparar el valor del elemento de la columna de la matriz "original" es mayor al elemento del arreglo "promedios", la matriz "nueva" en esa posición debe tener un 1; si es menor un -1 y si es igual un 0.

Ej:

Matriz original

1.1 2.2 3.3

4.4 5.5 6.6

7.7 8.8 9.9

Arreglo promedios (con promedios de columnas)

4.4 5.5 6.6

Matriz nueva

-1 -1 -1

0 0 0

1 1 1

**Respuesta:** (penalty regime: 0 %)

Reiniciar respuesta

```
1 void matrizNueva (float original[][N], float nueva[][N], float promedios[N])
2 {
3     float sum=0.0;
4
5     for(int j=0;j<N;j++)
6     {
7         sum=0.0;
8         for(int i=0;i<N;i++)
9         {
10             sum+=original[i][j];
11         }
12         promedios[j]=sum/N;
13     }
14
15     //los almaceno en el arreglo "promedios"
16
17     //comparo los elementos de la primera columna de la matriz "original" con el primer elemento
18     //arreglo "promedios", los elementos de la segunda columna de la matriz con el segundo elemento
19     //del arreglo y así sucesivamente. Si al comparar el valor del elemento de la columna de la matriz
20     //es mayor al elemento del arreglo, la matriz "nueva" en esa posición debe tener un 1;
21     //si es menor un -1 y si es igual un 0.
22
23     for(int j=0;j<N;j++)
```

	Test	Expected
✓	<pre>float original[N][N]={1.1,2.2,3.3,4.4,5.5,6.6,7.7,8.8,9.9}; float nueva[N][N]={0}; float promedios[N]={0};  matrizNueva(original,nueva,promedios); \t for(int i=0;i&lt;N;i++)     cout&lt;&lt;promedios[i]; \t for(int i=0;i&lt;N;i++)     for(int j=0;j&lt;N;j++)         cout&lt;&lt;nueva[i][j];</pre>	4.45.56.6-1-1-1000:
✓	<pre>float original[N][N]={0}; float nueva[N][N]={0}; float promedios[N]={0};  matrizNueva(original,nueva,promedios); \t for(int i=0;i&lt;N;i++)     cout&lt;&lt;promedios[i]; \t for(int i=0;i&lt;N;i++)     for(int j=0;j&lt;N;j++)         cout&lt;&lt;nueva[i][j];</pre>	000000000000
✓	<pre>float original[N][N]={20,20,20,40,40,40,50,50,50}; float nueva[N][N]={0}; float promedios[N]={0};  matrizNueva(original,nueva,promedios); \t for(int i=0;i&lt;N;i++)     cout&lt;&lt;promedios[i]; \t for(int i=0;i&lt;N;i++)     for(int j=0;j&lt;N;j++)         cout&lt;&lt;nueva[i][j];</pre>	36.666736.666736.6

	Test	http://lev2.efn.uncor.edu/mod/quiz/review.php?a...
✓	<pre>float original[N][N]={1.5,2.8,3.6,11.4,25.4,18.6,-5.4,-3.5,-11.6}; float nueva[N][N]={0}; float promedios[N]={0};  matrizNueva(original,nueva,promedios); \t for(int i=0;i&lt;N;i++)     cout&lt;&lt;promedios[i]; \t for(int i=0;i&lt;N;i++)     for(int j=0;j&lt;N;j++)         cout&lt;&lt;nueva[i][j];</pre>	2.58.233333.53333-
✓	<pre>float original[N][N]={1,1,1,1,1,1,1,1,1}; float nueva[N][N]={0}; float promedios[N]={0};  matrizNueva(original,nueva,promedios); \t for(int i=0;i&lt;N;i++)     cout&lt;&lt;promedios[i]; \t for(int i=0;i&lt;N;i++)     for(int j=0;j&lt;N;j++)         cout&lt;&lt;nueva[i][j];</pre>	111000000000

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Realice un programa el cual solicite al usuario el ingreso de un valor entero  $n$  que será la cantidad de productos a procesar.

Una vez ingresado  $n$ , se deberá crear una matriz de  $n$  Filas x 2 Columnas de valores tipo float.

Luego se debe solicitar al usuario el ingreso de los  $n$  x 2 valores floats de la matriz.

La columna 0, tendrá los precios de venta de los productos y la columna 1 el porcentaje de descuento por pago al contado.

Los productos se identificarán por su número de FILA (subíndice).

Al ingresar los datos se debe verificar SIEMPRE que el valor de la columna 1 sea mayor a 5 y menor a 50 (representa porcentaje de descuento por pago contado). Sino solicitar reingreso hasta que sea válido.

Una vez cargada la matriz, el programa debe determinar **el producto mas económico pagando de contado**. Debe imprimir por pantalla un mensaje indicando **el precio, con formato de dos decimales, del producto más económico**

Ejemplo de ejecución:

Cantidad de elementos a procesar 3

Ingrese los precios para los productos:

11.5

10

10.30

15

14.50

4

Error reingrese descuento (rango 5 a 50):

5

**SALIDA:**

**El producto más económico pagando contado cuesta \$ 8.76**

**Respuesta:** (penalty regime: 0 %)

```

1  #include<iostream>
2  #include<iomanip>
3  using namespace std;
4  int main()
5  {
6      int n;
7
8      float min;
9      //"n" es la cantidad de productos a procesar.
10     cin>>n;
11     float matriz[n][2];
12     //ingresar matriz de nx 2 verificando que el valor de la colum 1 sea mayor a 5 y menor a 50
13     for(int i=0;i<n;i++)
14     {
15         cin>> matriz[i][0];
16         do
17             cin>> matriz[i][1];
18         while(matriz[i][1]<5||matriz[i][1]>50); //porcentaje de descuento por pago contado
19     }
20
21     //La columna 0, tiene los precios de venta de los productos y la columna 1 el porcentaje de descuento
22     //por pago al contado. Los productos se identificarán por su número de FILA (subíndice).
23
24     //determina el producto mas económico pagando de contado.
25     min=matriz[0][0]-(matriz[0][0]*(matriz[0][1]/100.0));
26     for(int i=1;i<n;i++)
27     {
28         if(min > (matriz[i][0]-(matriz[i][0]*(matriz[i][1]/100.0))))
29         {
30             min=matriz[i][0]-(matriz[i][0]*(matriz[i][1]/100.0));
31         }
32     }
33     //imprimir por pantalla el precio, con formato de dos decimales, del producto más económico
34     cout<<setw(2)<<setprecision(2)<<fixed<<min<<endl;
35
36     return 0;

```

	Input	Expected	Got	
✓	3 11.50 10 10.30 15 14.50 8	8.76	8.76	✓
✓	1 100 51 50	50.00	50.00	✓
✓	3 10 10 10 10 10 10	9.00	9.00	✓
✓	4 10 9 11 10.5 12 13 14 15	9.10	9.10	✓
✓	3 10 4 6 9.9 5 10.5 11	9.35	9.35	✓

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Escriba un programa que como primer paso pida con un "cin" que el usuario ingrese un número entero entre 1 y 10, el cual va a representar la cantidad de filas y columnas de una matriz cuadrada. Luego de ingresar este número, el programa debe pedir nuevamente mediante "cin" que se ingresen todos los números de la matriz fila por fila (siendo una matriz de números enteros). En ningún caso es necesario que el programa imprima un mensaje mediante "cout". Cuando la matriz este completa, el programa debe calcular una nueva matriz que es el resultado de sumar la matriz que ingreso el usuario con una copia transpuesta de esa misma matriz. Luego de calcular la nueva matriz, el programa debe imprimir todos los números fila por fila separados por un espacio como se muestra en el ejemplo que esta a continuación.

### Ejemplo

Si la entrada de datos es:

```
2 1 2 3 4
```

El resultado de salida debe ser:

```
2 5 5 8
```

Notar que el primer número de la entrada indica que se trata de una matriz de 2 filas y dos columnas, los siguientes 4 números indican los valores de dicha matriz, esto es:

```
1 2
```

```
3 4
```

La matriz transpuesta es:

```
1 3
```

```
2 4
```

La matriz resultante es la suma de las dos anteriores, es decir:

```
2 5
```

```
5 8
```

Por esta razón la salida del programa debe ser: 2 5 5 8

**Respuesta:** (penalty regime: 0 %)

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int n;
6
7     //n representa la cantidad de filas y columnas de una matriz cuadrada.
8     do
9     cin>>n;
10    while(n<1||n>10);
11    int matriz[n][n], result[n][n], transp[n][n];
12    //ingreso de la matriz
13    for(int i=0;i<n;i++)
14    {
15        for(int j=0;j<n;j++)
16        {
17            cin>> matriz[i][j];
18        }
19    }
20
21    //la matriz transpuesta
22    for(int i=0;i<n;i++)
23    {
24        for(int j=0;j<n;j++)
25        {
26            transp[i][j]=matriz[j][i];
27        }
28    }
29    //matriz resultado de sumar la que se ingreso con una copia transpuesta de esa misma matriz.
30    for(int i=0;i<n;i++)
31    {
32        for(int j=0;j<n;j++)
33        {
34            result[i][j]=matriz[i][j]+transp[i][j];
35        }
36    }
```



	Input
✓	2 1 2 3 4
✓	3 1 2 3 4 5 6 7 8 9
✓	1 1
✓	9 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
✓	4 3 7 5 1 9 3 4 1 10000 2138 123987 2434 349857 234 49587 234

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Realice un programa el cual solicite al usuario el ingreso de un valor entero  $n$  que será la cantidad de productos a procesar.

Una vez ingresado  $n$ , se deberá crear un matriz de  $n$  Filas x 2 Columnas de valores tipo float.

Luego se debe solicitar al usuario el ingreso de los  $n$  x 2 valores floats de la matriz.

La columna 0, tendrá los precios de venta por menor de los productos y la columna 1 los precios de venta por mayor.

Los productos se identificarán por su número de FILA (subíndice).

Al ingresar los datos se debe verificar SIEMPRE que el valor de la columna 1 sea **menor** al de la columna 0 por ser precio por mayor. Sino solicitar reingreso hasta que sea válido.

Una vez cargada la matriz, el programa debe determinar **el mayor porcentaje de descuento**. Debe imprimir por pantalla un mensaje indicando **el porcentaje de descuento con Formato de dos decimales**.

Ejemplo de ejecución:

Cantidad de elementos a procesar 3

Ingrese los precios para los productos:

11.50

8.20

10.30

9.50

14.50

9.90

**SALIDA:**

**El producto con mayor descuento tiene un 31,72 %.**

Aclaración:

La matriz cargada debe quedar:

11.50 8.20

10.30 9.50

14.50 9.90

**Respuesta:** (penalty regime: 0 %)

```
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4  int main()
5  {
6      int n;// cantidad de productos a procesar.
7      float max;
8
9      cin>>n;
10     float matriz[n][n];
11     //ingreso de la matriz. La columna 0, tendrá los precios de venta por menor y la columna
12     //precios de por mayor. Los productos se identificarán por su número de FILA (subíndice).
13     //Al ingresar los datos verifico que el valor de la columna 1 sea menor al de la columna
14     //ser precio por mayor. Sino solicitar reingreso hasta que sea válido.
15
16     for(int i=0;i<n;i++)
17     {
18         cin>>matriz[i][0];
19         do
20         {
21             cin>>matriz[i][1];
22             while(matriz[i][1]>=matriz[i][0]);
23         }
24     }
25     // determinar el mayor porcentaje de descuento.
26     max=1-(matriz[0][1]/matriz[0][0]);
27
28     for(int i=0;i<n;i++)
29     {
30         if(max<1-(matriz[i][1]/matriz[i][0]))
31             max=1-(matriz[i][1]/matriz[i][0]);
32     }
```

	Input	Expected	Got	
✓	3 11.50 8.20 10.30 9.50 14.50 9.90	31.72	31.72	✓
✓	3 10 9 9 8 8 7	12.50	12.50	✓
✓	3 10 9 11 10 12 11	10.00	10.00	✓
✓	4 1 2 0.50 3 4 2.50 5 3 8 6.50	50.00	50.00	✓
✓	3 10 10 9.30 46.30 39.60 27.50 23.48	14.62	14.62	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1/1.

Codifique una función cuyo prototipo es:

```
void contarveces (double a, double b, int& resultado);
```

La función recibe como argumentos dos valores de precisión doble, un valor entero por referencia y un valor bool por referencia.

La función debe determinar si el mayor de los dos doubles recibidos es múltiplo del menor double.

Además la función debe determinar cuántas veces “cabe” el double de menor valor en el double de mayor valor.

Finalmente debe almacenar en el bool True si es múltiplo y False si no es múltiplo y almacenar en el valor entero enviado por referencia el resultado de cuantas veces cabe.

Ayuda:

Buscar información sobre las limitaciones del operador % y la función fmod().

Ejemplo 1:

a=2.3

b=18.4

En múltiplo se debe almacenar True

En resultado se debe almacenar 8

(salida ya que 18.4 es múltiplo de 2.3 y cabe 8 veces)

Ejemplo 2:

a=16.5

b=0.33

En múltiplo se debe almacenar False

En resultado se debe almacenar 50

(salida ya que 16.5 no es múltiplo de .33 y cabe 50 veces en a).

**Respuesta:** (penalty regime: 0 %)

Reiniciar respuesta

```
1 void contarveces (double a, double b, int& resultado, bool& multiplo)
2 {
3     double may,men;
4     //Determinar si el mayor de los dos doubles recibidos es múltiplo del menor double y cuántas
5     //“cabe” el double de menor valor en el double de mayor valor.
6     //Almacenar en el bool True si es múltiplo y False si no es múltiplo y almacenar en el valor
7     //enviado por referencia el resultado de cuantas veces cabe.
8     if(a>b)
9     {
10         if( fmod(a,b)==0)
11             multiplo=true;
12         else
13             multiplo=false;
14
15         may=a;
16         men=b;
17     }
18     else
19     {
20         if( fmod(b,a)==0)
21             multiplo=true;
22         else
23             multiplo=false;
24
25         may=b;
26         men=a;
27     }
28 }
29
30 do
31     resultado++;
32 while(double (resultado)*men<mav);
```

	Test	Expected	Got	
✓	<pre>double a=2.3,b=18.4; int resultado=0; bool multiplo;  contarveces(a,b,resultado,multiplo); cout&lt;&lt;resultado&lt;&lt;endl; cout&lt;&lt;multiplo&lt;&lt;endl;</pre>	<pre>8 1</pre>	<pre>8 1</pre>	✓
✓	<pre>double a=16.5,b=0.33; int resultado=0; bool multiplo;  contarveces(a,b,resultado,multiplo); cout&lt;&lt;resultado&lt;&lt;endl; cout&lt;&lt;multiplo&lt;&lt;endl;</pre>	<pre>50 0</pre>	<pre>50 0</pre>	✓
✓	<pre>double a=1.0,b=1.0; int resultado=0; bool multiplo;  contarveces(a,b,resultado,multiplo); cout&lt;&lt;resultado&lt;&lt;endl; cout&lt;&lt;multiplo&lt;&lt;endl;</pre>	<pre>1 1</pre>	<pre>1 1</pre>	✓
✓	<pre>double a=1000.0,b=1.0; int resultado=0; bool multiplo;  contarveces(a,b,resultado,multiplo); cout&lt;&lt;resultado&lt;&lt;endl; cout&lt;&lt;multiplo&lt;&lt;endl;</pre>	<pre>1000 1</pre>	<pre>1000 1</pre>	✓
✓	<pre>double a=1.0,b=1000.0; int resultado=0; bool multiplo;  contarveces(a,b,resultado,multiplo); cout&lt;&lt;resultado&lt;&lt;endl; cout&lt;&lt;multiplo&lt;&lt;endl;</pre>	<pre>1000 1</pre>	<pre>1000 1</pre>	✓

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Codifique una función cuyo prototipo es:

```
void contarveces (double a, double b, int& resultado);
```

La función recibe como argumentos dos valores de precisión doble, un valor entero por referencia y un valor bool por referencia.

La función debe determinar si el mayor de los dos doubles recibidos es múltiplo del menor double.

Además la función debe determinar cuántas veces "cabe" el double de menor valor en el double de mayor valor.

Finalmente debe almacenar en el bool True si es múltiplo y False si no es múltiplo y almacenar en el valor entero enviado por referencia el resultado de cuantas veces cabe.

Ayuda:

Buscar información sobre las limitaciones del operador % y la función fmod().

Ejemplo 1:

a=2.3

b=18.4

En múltiplo se debe almacenar True

En resultado se debe almacenar 8

(salida ya que 18.4 es múltiplo de 2.3 y cabe 8 veces)

Ejemplo 2:

a=16.5

b=0.33

En múltiplo se debe almacenar False

En resultado se debe almacenar 50

(salida ya que 16.5 no es múltiplo de .33 y cabe 50 veces en a).

**Respuesta:** (penalty regime: 0 %)

Reiniciar respuesta

```
1 void contarveces (double a, double b, int& resultado, bool& multiplo)
2 {
3     double may,men;
4     //Determinar si el mayor de los dos doubles recibidos es múltiplo del menor double y cuántas
5     //"cabe" el double de menor valor en el double de mayor valor.
6     //Almacenar en el bool True si es múltiplo y False si no es múltiplo y almacenar en el valor
7     //enviado por referencia el resultado de cuantas veces cabe.
8     if(a>b)
9     {
10         if( fmod(a,b)==0)
11             multiplo=true;
12         else
13             multiplo=false;
14
15         may=a;
16         men=b;
17     }
18     else
19     {
20         if( fmod(b,a)==0)
21             multiplo=true;
22         else
23             multiplo=false;
24
25         may=b;
26         men=a;
```

	Test	Expected	Got	
✓	<pre>double a=2.3,b=18.4; int resultado=0; bool multiplo;  contarveces(a,b,resultado,multiplo); cout&lt;&lt;resultado&lt;&lt;endl; cout&lt;&lt;multiplo&lt;&lt;endl;</pre>	<pre>8 1</pre>	<pre>8 1</pre>	✓
✓	<pre>double a=16.5,b=0.33; int resultado=0; bool multiplo;  contarveces(a,b,resultado,multiplo); cout&lt;&lt;resultado&lt;&lt;endl; cout&lt;&lt;multiplo&lt;&lt;endl;</pre>	<pre>50 0</pre>	<pre>50 0</pre>	✓
✓	<pre>double a=1.0,b=1.0; int resultado=0; bool multiplo;  contarveces(a,b,resultado,multiplo); cout&lt;&lt;resultado&lt;&lt;endl; cout&lt;&lt;multiplo&lt;&lt;endl;</pre>	<pre>1 1</pre>	<pre>1 1</pre>	✓
✓	<pre>double a=1000.0,b=1.0; int resultado=0; bool multiplo;  contarveces(a,b,resultado,multiplo); cout&lt;&lt;resultado&lt;&lt;endl; cout&lt;&lt;multiplo&lt;&lt;endl;</pre>	<pre>1000 1</pre>	<pre>1000 1</pre>	✓
✓	<pre>double a=1.0,b=1000.0; int resultado=0; bool multiplo;  contarveces(a,b,resultado,multiplo); cout&lt;&lt;resultado&lt;&lt;endl; cout&lt;&lt;multiplo&lt;&lt;endl;</pre>	<pre>1000 1</pre>	<pre>1000 1</pre>	✓

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Codifique la definición de una función cuyo prototipo es: <http://lev2.efn.uncor.edu/mod/quiz/review.php?a...>

**int masRepetido(int arreglo[], int N);**

La función recibe como argumento un arreglo de enteros de tamaño indeterminado y un entero N que representa la cantidad de elementos en el arreglo. .

La función debe devolver como valor de retorno el numero que más se repite dentro del arreglo. En caso de que existan dos o mas números que se repiten la mayor cantidad de veces, devolver el que aparezca primero en el arreglo (índice más bajo).

**Respuesta:** (penalty regime: 0 %)

Reiniciar respuesta

```

1  int masRepetido(int arreglo[], int N)
2  {
3      int max;
4      int contar[N],maxcontar;
5
6      for(int i=0;i<N;i++)
7      {
8          contar[i]=0;
9      }
10
11     for(int i=0;i<N;i++)
12     {
13         for(int j=0;j<N;j++)
14         {
15             if(arreglo[i]==arreglo[j])
16             {
17                 contar[i]++;
18             }
19         }
20     }
21
22     max=arreglo[0];
23     maxcontar=contar[0];
24     for(int i=0;i<N;i++)
25     {
26         for(int j=0;j<N;j++)
27         {
28             if(contar[i]>contar[j])
29             {
30                 maxcontar=contar[i];
31             }
32         }
33     }
34
35     for (int i=0;i<N;i++ )
36     {
37         if(contar[i]==maxcontar)
38         {
39             max=arreglo[i];
40             break;
41         }
42     }
43     return max;
44 }
45

```

	Test	Expected	Got	
✓	<pre> const int N = 3; int A[N]={1,2,3}; int M; M = masRepetido(A,N); cout &lt;&lt; M; </pre>	1	1	✓



	Test	Expected	Got	
✓	<pre>const int N = 4; int A[N]={1,2,3,2}; int M; M = masRepetido(A,N); cout &lt;&lt; M;</pre>	2	2	✓
✓	<pre>const int N = 5; int A[N]={1,2,100,1,3}; int M; M = masRepetido(A,N); cout &lt;&lt; M;</pre>	1	1	✓
✓	<pre>const int N = 10; int A[N]={1,2,3,2,3,2,3,2,3,1}; int M; M = masRepetido(A,N); cout &lt;&lt; M;</pre>	2	2	✓

Your code failed one or more hidden tests.

Your code must pass all tests to earn any marks. Try again.

**Incorrecta**

Puntos para este envío: 0/1.

Codifique la definición de una función cuyo prototipo es: <http://lev2.efn.uncor.edu/mod/quiz/review.php?a...>

**void histograma(int frecuencia[], int N);**

La función recibe como argumento un arreglo de enteros de tamaño indeterminado y un entero N que representa la cantidad de elementos en el arreglo.

La función debe solicitar al usuario ingresar valores enteros mayores o iguales que 0 pero menores que N hasta que se ingrese un valor fuera de dicho rango. La función debe cargar en el arreglo frecuencia[ ] la frecuencia (cantidad de veces) correspondiente a cada valor de 0 hasta N-1 considerando los valores válidos ingresados.

**Respuesta:** (penalty regime: 0 %)

Reiniciar respuesta

```

1 void histograma(int frecuencia[], int N)
2 {
3     // Complete su codigo aqui
4     int a;
5     for(int k=0;k<N;k++)
6     {
7         frecuencia[k]=0;
8     }
9     do
10    {
11        cin>>a;
12        if(a>=0&&a<N)
13            frecuencia[a]++;
14    }
15    while(a>=0&&a<N); // sigue en el do mientras sea verdadero
16
17    return;
18 }
19

```

	Test	Input	Expected	Got	
✓	const int N = 3; int A[N]; histograma(A,N); for (int i=0; i<N; i++) cout << endl << A[i];	1 1 3	0 2 0	0 2 0	✓
✓	const int N = 3; int A[N]; histograma(A,N); for (int i=0; i<N; i++) cout << endl << A[i];	1 1 -1		0 2 0	✓
✓	const int N = 3; int A[N]; histograma(A,N); for (int i=0; i<N; i++) cout << endl << A[i];	0 1 1 2 2 2 3	1 2 3	1 2 3	✓
✓	const int N = 2; int A[N]; histograma(A,N); for (int i=0; i<N; i++) cout << endl << A[i];	1 1 1 -1	0 3	0 3	✓
✓	const int N = 4; int A[N]; histograma(A,N); for (int i=0; i<N; i++) cout << endl << A[i];	1 3 4	0 1 0 1	0 1 0 1	✓

	Test	Input	Expected Output	Actual Output	
✓	<pre>const int N = 3; int A[N]; histograma(A,N); for (int i=0; i&lt;N; i++)     cout &lt;&lt; endl &lt;&lt; A[i];</pre>	-1	<pre>0 0 0</pre>	<pre>0 0 0</pre>	✓

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Codifique la definición de una función cuyo prototipo es: <http://lev2.efn.uncor.edu/mod/quiz/review.php?a...>

**void invierte(int org[][3], int dst[][3], int F);**

La función recibe como argumento un arreglo bidimensional de enteros con un numero indeterminado de filas, 3 columnas llamado org y un entero F que representa la cantidad de filas en el arreglo. Además recibe otro arreglo bidimensional de enteros con un numero indeterminado de filas, 3 columnas llamado dst.

La función debe copiar los elementos de org a dst pero en orden inverso de filas. Esto es, la primer fila de org debe almacenarse en la última de dst, la segunda fila de org debe almacenarse en la ante-última y así sucesivamente.

**Respuesta:** (penalty regime: 0 %)

Reiniciar respuesta

```

1 void invierte(int org[][3], int dst[][3], int F)
2 {
3     for(int i=0;i<F;i++)
4     {
5         for(int j=0;j<3;j++)
6         {
7             dst[i][j]=org[F-1-i][j];
8         }
9     }
10
11     return;
12 }
```

	Test	Expected	Got	
✓	const int N = 3; int A[N][3]={1,2,3,4,5,6,7,8,9}; int B[N][3]; invierte(A,B,N); for(int i=0; i<N; i++) cout<<B[i][0]<<" "<<B[i][1]<<" "<<B[i][2]<<endl;	7, 8, 9 4, 5, 6 1, 2, 3	7, 8, 9 4, 5, 6 1, 2, 3	✓
✓	const int N = 4; int A[N][3]={11,-2,3,34,5,6,27,8,9,6,90,2}; int B[N][3]; invierte(A,B,N); for(int i=0; i<N; i++) cout<<B[i][0]<<" "<<B[i][1]<<" "<<B[i][2]<<endl;	6, 90, 2 27, 8, 9 34, 5, 6 11, -2, 3	6, 90, 2 27, 8, 9 34, 5, 6 11, -2, 3	✓
✓	const int N = 5; int A[N][3]={-1,1232,43,4,25,6,7,8,9,12,4,12,32,45,12}; int B[N][3]; invierte(A,B,N); for(int i=0; i<N; i++) cout<<B[i][0]<<" "<<B[i][1]<<" "<<B[i][2]<<endl;	32, 45, 12 12, 4, 12 7, 8, 9 4, 25, 6 -1, 1232, 43	32, 45, 12 12, 4, 12 7, 8, 9 4, 25, 6 -1, 1232, 43	✓

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Codifique la definición de una función cuyo prototipo es: <http://lev2.efn.uncor.edu/mod/quiz/review.php?a...>

**void minimos(int org[], int N, int &minImpar, int &minPar);**

La función recibe como argumento un arreglo de enteros de tamaño indeterminado (org) y un entero N que representa la cantidad de elementos en el arreglo. Además recibe dos argumentos por referencia para almacenar los resultados.

La función debe encontrar el mínimo numero par y el mínimo numero impar y debe almacenarlos en los argumentos pasados por referencia minImpar y minPar. Si no existieran números pares (impares) deberá almacenar el número cero en minPar(minImpar).

**Respuesta:** (penalty regime: 0 %)

Reiniciar respuesta

```

8      maxi=org[k];
9      k++;
10     }
11     while(maxi%2==0&&maxi<N);
12     do
13     {
14         maxp=org[l];
15         l++;
16     }
17     while(maxp%2==1&&l<N);
18
19     for(int i=1; i<N;i++)
20     {
21         if((org[i]%2)==0)
22         {
23             if(org[i]<minp)
24             {
25                 minp=org[i];
26             }
27         }
28     }
29     if(maxp%2==1)
30     {
31         maxp=0;
32     }
33
34     maxi=org[0];
35     for(int i=1; i<N;i++)
36     {
37         if(((org[i]%2)==0)&&(org[i]>maxi))
38         {
39             maxi=org[i];
40         }
41     }
42     if(maxi%2==0)
43     {
44         maxi=0;
45     }
46     return;
47 }
48

```

### Syntax Error(s)

```

prog.cpp:12:1: error: expected unqualified-id before '{' token
{
^

```

**Incorrecta**

Puntos para este envío: 0/1.

Codifique la definición de una función cuyo prototipo es: <http://lev2.efn.uncor.edu/mod/quiz/review.php?a...>

**void colMin(int org[][3], int dst[], int F);**

La función recibe como argumento un arreglo bidimensional de enteros con un número indeterminado de filas y 3 columnas (**org**) y un entero **F** que representa la cantidad de filas del arreglo. El arreglo **org** contiene valores enteros que ya han sido cargados antes (la función no debe cargar valor alguno en dicho arreglo).

La función debe almacenar la columna que contiene el mínimo del arreglo **org** en el arreglo unidimensional **dst** recibido como argumento.

**Respuesta:** (penalty regime: 0 %)

Reiniciar respuesta

```

1 void colMin(int org[][3], int dst[], int F)
2
3 {
4     int min, col;
5
6     min=org[0][0];
7     col=0;
8     for(int i=0;i<F;i++)
9     {
10         for(int j=0;j<3;j++)
11         {
12             if(min>org[i][j])
13             {
14                 min=org[i][j];
15                 col=j;
16             }
17         }
18     }
19     for(int i=0;i<F;i++)
20     {
21         dst[i]=org[i][col];
22     }
23     return;
24 }
25

```

	Test	Expected	Got	
✓	<pre> const int N = 3; int A[N][3]={1,2,3,4,5,6,7,8,9}; int B[N]; colMin(A,B,N); for(int i=0; i&lt;N; i++)     cout&lt;&lt;B[i]&lt;&lt;endl; </pre>	1 4 7	1 4 7	✓
✓	<pre> const int N = 4; int A[N][3]={12311,652,23,34,158,106,555,28,19,122,35443,123}; int B[N]; colMin(A,B,N); for(int i=0; i&lt;N; i++)     cout&lt;&lt;B[i]&lt;&lt;endl; </pre>	23 106 19 123	23 106 19 123	✓
✓	<pre> const int N = 1; int A[N][3]={1,2,-3}; int B[N]; colMin(A,B,N); for(int i=0; i&lt;N; i++)     cout&lt;&lt;B[i]&lt;&lt;endl; </pre>	-3	-3	✓

	Test	Expected	Got	
✓	<pre>const int N = 3; int A[N][3]={-11, -2, -33, -4, -5, -6, -7, -8, -9}; int B[N]; colMin(A,B,N); for(int i=0; i&lt;N; i++)     cout&lt;&lt;B[i]&lt;&lt;endl;</pre>	-33 -6 -9	-33 -6 -9	✓

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Codifique la definición de una función cuyo prototipo es: <http://lev2.efn.uncor.edu/mod/quiz/review.php?a...>

**void sumaFilas(int org[][3], int dst[], int F);**

La función recibe como argumento un arreglo bidimensional de enteros con un número indeterminado de filas y 3 columnas (org) y un entero F que representa la cantidad de filas del arreglo. Además recibe un arreglo unidimensional de tamaño indeterminado para almacenar el resultado.

La función debe almacenar en el arreglo dst la suma de los elementos de cada fila. El elemento 0 del arreglo dst debe almacenar la suma de los tres elementos de la fila 0, el elemento 1 del arreglo dst debe almacenar la suma de los tres elementos de la fila 1 y así para cada fila del arreglo org.

**Respuesta:** (penalty regime: 0 %)

Reiniciar respuesta

```

1 void sumaFilas(int org[][3], int dst[], int F)
2 {
3     int sum;
4     for(int i=0;i<F;i++)
5     {
6         sum=0;
7         for(int j=0;j<3;j++)
8         {
9             sum+=org[i][j];
10        }
11        dst[i]=sum;
12    }
13
14    return;
15 }
```

	Test	Expected	Got	
✓	<pre>const int N = 3; int A[N][3]={1,2,3,4,5,6,7,8,9}; int B[N]; sumaFilas(A,B,N); for(int i=0; i&lt;N; i++)     cout&lt;&lt;B[i]&lt;&lt;endl;</pre>	6 15 24	6 15 24	✓
✓	<pre>const int N = 4; int A[N][3]={12311,652,23,34,158,106,555,28,19,122,35443,123}; int B[N]; sumaFilas(A,B,N); for(int i=0; i&lt;N; i++)     cout&lt;&lt;B[i]&lt;&lt;endl;</pre>	12986 298 602 35688	12986 298 602 35688	✓
✓	<pre>const int N = 1; int A[N][3]={1,2,-3}; int B[N]; sumaFilas(A,B,N); for(int i=0; i&lt;N; i++)     cout&lt;&lt;B[i]&lt;&lt;endl;</pre>	0	0	✓
✓	<pre>const int N = 3; int A[N][3]={-11,-2,-33,-4,-5,-6,-7,-8,-9}; int B[N]; sumaFilas(A,B,N); for(int i=0; i&lt;N; i++)     cout&lt;&lt;B[i]&lt;&lt;endl;</pre>	-46 -15 -24	-46 -15 -24	✓

Todas las pruebas superadas. ✓

Correcta



[◀ Soluciones](#)[Ir a...](#)[EC4 ▶](#)