

[Dashboard](#) ► [My courses](#) ► [InfoC++](#) ► [Clase 10](#) ► [Guía interactiva 10](#)

<b>Started on</b>	Thursday, 18 October 2018, 5:35 PM
<b>State</b>	Finished
<b>Completed on</b>	Saturday, 27 October 2018, 1:23 PM
<b>Time taken</b>	8 days 19 hours
<b>Grade</b>	<b>20.00</b> out of 20.00 ( <b>100%</b> )

Question 1

Correct

Mark 1.00 out of 1.00

Escriba la definición de un a función en C++ llamada incremento, de tipo void que recibe por referencia una variable x de tipo entero y que incrementa su valor en 2 unidades.

Por ejemplo:

si se llama a la función de la siguiente manera:

```
int a = 4;
```

```
incremento(a);
```

```
cout << a;
```

saldrá por pantalla un 6

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

Reset answer

```
1 void incremento(int &x)
2 {
3     x=x+2;
4     return;
5 }
```

	Test	Expected	Got	
✓	int a=4; incremento(a); cout << a <<endl; incremento(a); cout << a <<endl; incremento(a); cout << a <<endl; incremento(a); cout << a <<endl;	6 8 10 12	6 8 10 12	✓
✓	int a=-124; incremento(a); cout << a;	-122	-122	✓

	Test	Expected	Got	
✓	int a=0; incremento(a); cout << a;	2	2	✓

Passed all tests! ✓

### Question author's solution:

```
void incremento(int &x)
{
    x+=2;
}
```

**Correct**

Marks for this submission: 1.00/1.00.

## Question 2

Correct

Mark 1.00 out of 1.00

Escriba la definición de un a función en C++ llamada ordenar de tipo void. La función acepta dos variables por referencia m y M de tipo entero. En caso de que el valor almacenado en m sea menor que M no se debe realizar nada. En caso de que el valor almacenado en m sea mayor a M se deben intercambiar los valores almacenados en m y M.

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

Reset answer

```
1 void ordenar(int &m, int &M)
2 {
3     if(m>M)
4     {
5         int x=m;
6         m=M;
7         M=x;
8     }
9     return;
10
11 }
```

	Test	Expected	Got	
✓	int a=4; int b=5; ordenar(a,b); cout << a << " " << b <<endl;	4 5	4 5	✓
✓	int a=5; int b=4; ordenar(a,b); cout << a << " " << b <<endl;	4 5	4 5	✓
✓	int a=4; int b=4; ordenar(a,b); cout << a << " " << b <<endl;	4 4	4 4	✓

Passed all tests! ✓

**Question author's solution:**

```
void ordenar(int &m, int &M)
{
    int t;
    if(m>M)
    {
        t=m;
        m=M;
        M=t;
    }
}
```

**Correct**

Marks for this submission: 1.00/1.00.

## Question 3

Correct

Mark 1.00 out of  
1.00

Complete el siguiente programa:

Escriba el cuerpo de la función menorMayor para que encuentre el menor y el mayor de cuatro valores (a, b, c y d) y los almacene en m y M respectivamente.

Complete el programa principal con la llamada a la función para encontrar el mayor y el menor de los cuatro valores ingresados por teclado

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

Reset answer

```

1  #include <iostream>
2  using namespace std;
3
4  void menorMayor(int a, int b, int c, int d, int &m, int &M);
5
6  int main()
7  {
8      int x1, x2, x3, x4, may, men;
9      cout << "Ingrese x1: " << endl;
10     cin >> x1;
11
12     cout << "Ingrese x2: " << endl;
13     cin >> x2;
14
15     cout << "Ingrese x3: " << endl;
16     cin >> x3;
17
18     cout << "Ingrese x4: " << endl;
19     cin >> x4;
20
21     //complete con la llamada a la funcion
22     menorMayor(x1, x2, x3, x4, men, may);
23     menorMayor(x1, x2, x3, x4, men, may);
24
25     cout << "El menor de los 4 valores es: " << men << endl;
26     cout << "El mayor de los 4 valores es: " << may << endl;
27
28     return 0;
29 }
30
31 void menorMayor(int a, int b, int c, int d, int &m, int &M)
32 {
33     //complete el cuerpo de la funcion
34     if(a<b&&a<c&&a<d)
35         m=a;
36     else if(b<a&&b<c&&b<d)
37         m=b;
38     else if(c<a&&c<b&&c<d)
39         m=c;
40     else if(d<a&&d<c&&d<b)
41         m=d;
42
43     if(a>b&&a>c&&a>d)
44         M=a;
45     else if(b>a&&b>c&&b>d)
46         M=b;
47     else if(c>a&&c>b&&c>d)
48         M=c;
49     else if(d>a&&d>b&&d>c)
50         M=d;

```

	Input	Expected	Got
✓	34 12 65 21	El menor de los 4 valores es: 12 El mayor de los 4 valores es: 65	Ingrese x1: Ingrese x2: Ingrese x3: Ingrese x4: El menor de los 4 va. El mayor de los 4 va.
✓	-12 -312 -46 -123	El menor de los 4 valores es: -312 El mayor de los 4 valores es: -12	Ingrese x1: Ingrese x2: Ingrese x3: Ingrese x4: El menor de los 4 va. El mayor de los 4 va.
✓	213 54321 1234 65	El menor de los 4 valores es: 65 El mayor de los 4 valores es: 54321	Ingrese x1: Ingrese x2: Ingrese x3: Ingrese x4: El menor de los 4 va. El mayor de los 4 va.

Passed all tests! ✓

**Question author's solution:**

```
#include <iostream>
using namespace std;

void menorMayor(int a, int b, int c, int d, int &m, int &M);

int main()
{
    int x1, x2, x3, x4, may, men;
    cout << "Ingrese x1: " << endl;
    cin >> x1;

    cout << "Ingrese x2: " << endl;
    cin >> x2;

    cout << "Ingrese x3: " << endl;
    cin >> x3;

    cout << "Ingrese x4: " << endl;
    cin >> x4;

    menorMayor(x1,x2,x3,x4,men,may);

    cout << "El menor de los 4 valores es: " << men << endl;
    cout << "El mayor de los 4 valores es: " << may << endl;

    return 0;
}

void menorMayor(int a, int b, int c, int d, int &m, int &M)
{
    if(a<=b && a<=c && a<=d)
    {
        m = a;
    }
    else if(b<=a && b<=c && b<=d)
    {
        m = b;
    }
    else if(c<=a && c<=b && c<=d)
    {
        m = c;
    }
    else
    {
        m = d;
    }

    if(a>=b && a>=c && a>=d)
    {
        M = a;
    }
    else if(b>=a && b>=c && b>=d)
```



```
{  
    M = b;  
}  
else if(c>=a && c>=b && c>=d)  
{  
    M = c;  
}  
else  
{  
    M = d;  
}  
}
```

**Correct**

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 1.00 out of 1.00

Dada la siguiente definición de función:

```
#include <iostream>
using namespace std;

void swap(int &a, int &b)
{
    int t = a;
    a = b;
    b = t;
}

int main()
{
```

Seleccione las opciones correctas:

Select one or more:

☐ a. La función swap es de tipo void, recibe dos argumentos por valor y los intercambia.

☐ b.

```
int main()
{
    int x, y;

    cout << "Ingrese x: ";
    cin >> x;

    cout << "Ingrese y: ";
    cin >> y;

    swap(y, x);

    cout << x << " " << y;

}
```

Luego de llamar a la función swap los valores de las variables x e y permanecen sin modificación.

☐ c. La función swap devuelve los valores de a y b como valor de retorno

☒ d. La función swap es de tipo void, recibe dos argumentos por referencia e intercambia los valores de las dos variables. ✓

☐ e. Asumiendo que se ingresan por teclado los valores -2 y 12 (en ese orden),

Sale por pantalla:

**-2 12**

```
int main()
{
    int x, y;

    cout << "Ingrese x: ";
    cin >> x;

    cout << "Ingrese y: ";
    cin >> y;

    swap(x, y);

    cout << x << " " << y;

}
```



f. Los dos programas principales que se muestran a continuación son

equivalentes:

```
int main()
{
    int x, y;

    cout << "Ingrese x: ";
    cin >> x;

    cout << "Ingrese y: ";
    cin >> y;

    swap(y, x);

    cout << x << " " << y;
}

int main()
{
    int x, y;

    cout << "Ingrese x: ";
    cin >> x;

    cout << "Ingrese y: ";
    cin >> y;

    swap(x, y);

    cout << x << " " << y;
}
```



g. Asumiendo que se ingresan por teclado los valores -2 y 12 (en ese orden),

Sale por pantalla:

**12 -2**

```
int main()
{
    int x, y;

    cout << "Ingrese x: ";
    cin >> x;

    cout << "Ingrese y: ";
    cin >> y;

    swap(x, y);

    cout << x << " " << y;
}
```



h.

## Respuesta correcta

The correct answers are: La función swap es de tipo void, recibe dos argumentos por referencia e intercambia los valores de las dos variables., Asumiendo que se ingresan por teclado los valores -2 y 12 (en ese orden), Sale por pantalla:

**12 -2**

```
int main()
{
    int x, y;

    cout << "Ingrese x: ";
    cin >> x;

    cout << "Ingrese y: ";
    cin >> y;

    swap(x, y);

    cout << x << " " << y;
}
```

, Los dos programas principales que se

muestran a continuación son equivalentes:

```
int main()
{
    int x, y;

    cout << "Ingrese x: ";
    cin >> x;

    cout << "Ingrese y: ";
    cin >> y;

    swap(y, x);

    cout << x << " " << y;
}
```

```
int main()
{
    int x, y;

    cout << "Ingrese x: ";
    cin >> x;

    cout << "Ingrese y: ";
    cin >> y;

    swap(x, y);

    cout << x << " " << y;
}
```

Los cambios realizados sobre las variables a y b dentro de la función swap se realizan a su vez sobre las variables y y x respectivamente.

**Correct**

Marks for this submission: 1.00/1.00.

Question 5

Correct

Mark 1.00 out of 1.00

Un programador escribió el siguiente programa. Sin embargo no logra pasar los casos de prueba (tests) provistos. Comprenda que debe hacer el programa y corrija el código sin modificar el programa principal.

**For example:**

Input	Result
2	3
3	2
4	5
5	4

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

Reset answer

```

1  #include <iostream>
2  using namespace std;
3
4  void intercambiar(int &n, int &m)
5  {
6      int temp = n;
7      n = m;
8      m = temp;
9  }
10 int main()
11 {
12     int a,b;
13     cin>>a>>b;
14
15     intercambiar(a,b);
16     cout<<a<<endl<<b<<endl;
17
18 }
```

	Input	Expected	Got	
✓	2	3	3	✓
	3	2	2	
✓	4	5	5	✓
	5	4	4	
✓	100	200	200	✓
	200	100	100	
✓	123	432	432	✓
	432	123	123	
✓	1024	2048	2048	✓
	2048	1024	1024	

	Input	Expected	Got	
✓	-100 34	34 -100	34 -100	✓

Passed all tests! ✓

### Question author's solution:

```
#include <iostream>
using namespace std;

void intercambiar(int& n, int& m)
{
    int temp = n;
    n = m;
    m = temp;
}

int main()
{
    int a,b;
    cin>>a>>b;

    intercambiar(a,b);
    cout<<a<<endl<<b<<endl;

}
```

**Correct**

Marks for this submission: 1.00/1.00.

## Question 6

Correct

Mark 1.00 out of  
1.00

Dada la siguiente definición de función seleccione las opciones correctas

```
#include <iostream>
using namespace std;

double num1 = 343.12;
double num2 = 200.43;
double otro = 3.1;

double suma(double num1, double b)
{
    double c = num1 + num2 + b;
    return c;
}
```

Select one or more:

- ☐ a. Las variables num1, num2 y otros se pueden alcanzar únicamente desde el cuerpo de la función main.
- ☐ b.

```
int main()
{
    double num1 = 13.1;
    double num2 = 123.1;
    double a, b, c;

    a = suma(num1, num2);
    cout << a << endl;

    b = suma(num1, ::num2);
    cout << b << endl;

    c = suma(::num1, num2);
    cout << c << endl;
}
```

Al ejecutar el programa sale por pantalla:

336.63

336.63

336.63

- ☐ c.

```
int main()
{
    double num1 = 13.1;
    double num2 = 123.1;
    double a, b, c;

    cout << num1 << endl;
    cout << num2 << endl;
    cout << ::num1 << endl;
    cout << ::num2 << endl;
    cout << otro << endl;
}
```

No es posible compilar el programa porque la variable otro no es accesible desde el main

- ☐ d.

```
int main()
{
    double num1 = 13.1;
    double num2 = 123.1;
    double a, b, c;

    cout << num1 << endl;
    cout << num2 << endl;
    cout << ::num1 << endl;
    cout << ::num2 << endl;
    cout << otro << endl;
}
```

Tras ejecutar el programa sale por pantalla:

13.1  
123.1  
13.1  
123.1  
3.1



e.

```
int main()
{
    double num1 = 13.1;
    double num2 = 123.1;
    double a, b, c;

    cout << num1 << endl;
    cout << num2 << endl;
    cout << ::num1 << endl;
    cout << ::num2 << endl;
    cout << otro << endl;
}
```

Tras ejecutar el programa sale por pantalla:

13.1  
123.1  
343.12  
200.43  
3.1 ✓



f.

```
int main()
{
    double num1 = 13.1;
    double num2 = 123.1;
    double a, b, c;

    a = suma(num1, num2);
    cout << a << endl;

    b = suma(num1, ::num2);
    cout << b << endl;

    c = suma(::num1, num2);
    cout << c << endl;
}
```

Al ejecutar el programa sale por pantalla:

666.65  
743.98  
666.65



g. Las variables num1, num2 y otros son variables globales ✓





h.

```
int main()
{
    double num1 = 13.1;
    double num2 = 123.1;
    double a, b, c;

    a = suma(num1, num2);
    cout << a << endl;

    b = suma(num1, ::num2);
    cout << b << endl;

    c = suma(::num1, num2);
    cout << c << endl;
}
```

Al ejecutar el programa sale por pantalla:

336.63

413.96

666.65 ✓



i.

```
int main()
{
    double num1 = 13.1;
    double num2 = 123.1;
    double a, b, c;

    cout << num1 << endl;
    cout << num2 << endl;
    cout << ::num1 << endl;
    cout << ::num2 << endl;
    cout << otro << endl;
}
```

Tras ejecutar el programa sale por pantalla:

343.12

200.43

343.12

200.43

3.1



j.

```
int main()
{
    double num1 = 13.1;
    double num2 = 123.1;
    double a, b, c;

    cout << num1 << endl;
    cout << num2 << endl;
    cout << ::num1 << endl;
    cout << ::num2 << endl;
    cout << otro << endl;
}
```

Tras ejecutar el programa sale por pantalla:

343.12

200.43

13.1

123.1

3.1

Respuesta correcta

The correct answers are: Las variables num1, num2 y otros son variables globales,

```
int main()
{
    double num1 = 13.1;
    double num2 = 123.1;
    double a, b, c;

    cout << num1 << endl;
    cout << num2 << endl;
    cout << ::num1 << endl;
    cout << ::num2 << endl;
    cout << otro << endl;
}
```

Tras ejecutar el programa sale por pantalla:

**13.1****123.1****343.12****200.43****3.1,**

```
int main()
{
    double num1 = 13.1;
    double num2 = 123.1;
    double a, b, c;

    a = suma(num1, num2);
    cout << a << endl;

    b = suma(num1, ::num2);
    cout << b << endl;

    c = suma(::num1, num2);
    cout << c << endl;
}
```

Al ejecutar el programa sale por pantalla:

**336.63****413.96****666.65****Correct**

Marks for this submission: 1.00/1.00.

Question 7

Correct

Mark 1.00 out of 1.00

Un programador escribió el siguiente programa pero no logra pasar los casos de prueba provistos. Comprenda que debe hacer el programa y corrija el código sin cambiar el perfil de la función de manera que los casos de pruebas sean superados.

**For example:**

Input	Result
2.5 1.5 5	El desvio estandard es: 1.47196

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

Reset answer

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  double calcular_desvio_standard(double, double, double);
6
7  int main()
8  {
9      double num1,num2,num3;
10     cin>>num1>>num2>>num3;
11
12     cout<<"El desvio estandard es: "<< calcular_desvio_standard(num1
13 }
14
15 double calcular_desvio_standard(double a, double b, double c)
16 {
17     double media = (a + b + c) / 3;
18     double varianza = (pow(a - media, 2) + pow(b - media, 2) + pow(c
19
20     return sqrt(varianza);
21 }
22

```

	Input	Expected	Got
✓	2.5 1.5 5	El desvio estandard es: 1.47196	El desvio estandard es:
✓	3 15.6 23.4	El desvio estandard es: 8.40476	El desvio estandard es:
✓	10.4 16.8 13.2	El desvio estandard es: 2.61958	El desvio estandard es:

	Input	Expected	Got
✓	-12 -873.3 -45	El desvio estandard es: 398.47	El desvio estandard es:
✓	0 18 23	El desvio estandard es: 9.87702	El desvio estandard es:
✓	-5 5 15	El desvio estandard es: 8.16497	El desvio estandard es:

Passed all tests! ✓

### Question author's solution:

```
#include <iostream>
#include <cmath>
using namespace std;

double calcular_desvio_standard(double, double, double);

int main()
{
    double num1,num2,num3;
    cin>>num1>>num2>>num3;

    cout<<"El desvio estandard es: "<< calcular_desvio_standard(n
um1, num2, num3)<<endl;
}

double calcular_desvio_standard(double a, double b, double c)
{
    double media = (a + b + c) / 3;
    double varianza = (pow(a - media, 2) + pow(b - media, 2) + po
w(c - media, 2)) / 3;

    return sqrt(varianza);
}
```

**Correct**

Marks for this submission: 1.00/1.00.

Question 8

Correct

Mark 1.00 out of 1.00

Escriba la definición de una función llamada min que acepta como argumento un arreglo unidimensional de 15 elementos enteros y devuelve el menor de ellos como valor de retorno.

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

Reset answer

```
1 int min(int A[15])
2 {
3     int menor;
4     menor=A[0];
5     for(int i=1;i<15;i++)
6         if(menor>A[i])
7             menor=A[i];
8         else if(menor<A[i])
9             menor=menor;
10    return menor;
11 }
```

	Test
✓	<pre>int A[15] = {10,24,32,1234,-5,6,72,38,-9,-10,-11,12,-13,14,-15}; int a; a = min(A); cout &lt;&lt; a;</pre>
✓	<pre>int A[15] = {-10,-24,-32,-1234,-5,-6,-72,-38,-9,-10,-11,12,-13,14,-15}; int a; a = min(A); cout &lt;&lt; a;</pre>
✓	<pre>int M[15] = {510,424,3122,31234,4512,6312,74122,45238,4129,1210,1210,1210,1210,1210,1210}; int a; a = min(M); cout &lt;&lt; a;</pre>

Passed all tests! ✓

**Question author's solution:**

```
int min(int A[15])
{
    int ret = A[0];
    for(int i=0; i<15; i++)
    {
        if(A[i]<ret)
        {
            ret = A[i];
        }
    }
    return ret;
}
```

**Correct**

Marks for this submission: 1.00/1.00.

Question 9

Correct

Mark 1.00 out of 1.00

Escriba la definición de una función llamada min que acepta como argumento un arreglo unidimensional A y un valor entero N. N representa la cantidad de elementos que tiene el arreglo. La función debe devolver el menor de los elementos como valor de retorno.

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

Reset answer

```
1 int min(int A[], int N)
2 {
3     int min=A[0];
4     for(int i=1;i<N;i++)
5     {
6         if (min>A[i])
7             min=A[i];
8         else if(min<A[i])
9             min=min;
10    }
11    return min;
12
13 }
```

	Test
✓	<pre>int A[15] = {10,24,32,1234,-5,6,72,38,-9,-10,-11,12,-13,14,-15}; int a; a = min(A, 15); cout &lt;&lt; a;</pre>
✓	<pre>int A[] = {-10,-24,-32,-1234,-5,-6,-72}; int a; a = min(A, 7); cout &lt;&lt; a;</pre>
✓	<pre>int M[15] = {510,424,3122,31234,4512,6312,74122,45238,4129,1210, int a; a = min(M, 10); cout &lt;&lt; a;</pre>

Passed all tests! ✓

**Question author's solution:**

```
int min(int A[], int N)
{
    int ret = A[0];
    for(int i=0; i<N; i++)
    {
        if(A[i]<ret)
        {
            ret = A[i];
        }
    }
    return ret;
}
```

**Correct**

Marks for this submission: 1.00/1.00.



Mark 1.00 out of 1.00

```

1 double promedio(double A[3][4])
2 {
3     double sum=0, prom;
4
5     for(int i=0;i<3;i++)
6     {
7         for(int j=0; j<4; j++)
8         {
9             sum+= A[i][j];
10        }
11    }
12    prom=sum/12;
13    return prom;
14
15 }

```

```
double promedio(double A[3][4])
{
    double ret = 0.0;
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<4; j++)
        {
            ret += A[i][j];
        }
    }
    return ret/(3*4);
}
```

**Correct**

Marks for this submission: 1.00/1.00.

Mark 1.00 out of 1.00

```

1 void fila(double M[4][3], double F[3], int fi)
2 {
3     for(int j=0;j<3;j++)
4         F[j]=M[fi][j];
5     return;
6 }

```

	Test
✓	<pre>double T[4][3] = {84.0, 137.58, 73.08, 112.95, 131.03, 45.55, 90.0, 10.0, 10.0}; double R[3]; fila(T,R,0); for(int i=0; i&lt;3; i++) {     cout &lt;&lt; R[i] &lt;&lt; " "; }</pre>
✓	<pre>double T[4][3] = {24.44, -5.74, -19.06, -44.99, 29.86, 12.57, 40.0, 10.0, 10.0}; double R[3]; fila(T,R,1); for(int i=0; i&lt;3; i++) {     cout &lt;&lt; R[i] &lt;&lt; " "; }</pre>
✓	<pre>double T[4][3] = {-357.43, -258.98, 429.76, 163.61, 489.02, 395.0, 10.0, 10.0, 10.0}; double R[3]; fila(T,R,2); for(int i=0; i&lt;3; i++) {     cout &lt;&lt; R[i] &lt;&lt; " "; }</pre>

27/10/18

**Question author's solution:**

```
void fila(double M[4][3], double F[3], int fi)
{
    for(int i=0; i<3; i++)
    {
        F[i] = M[fi][i];
    }
}
```

**Correct**

Marks for this submission: 1.00/1.00.

## Question 12

Correct

Mark 1.00 out of  
1.00

Complete el siguiente programa que permite calcular la suma de dos vectores en  $R^n$  utilizando la función `suma_v`.

La función `suma_v` debe almacenar en `s` la suma elemento a elemento de los arreglos `u` y `v`. `N` representa la cantidad de elementos en los arreglos.

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

Reset answer

```

1  #include <iostream>
2  using namespace std;
3
4  void suma_v(double u[], double v[], double s[], int N);
5
6  int main()
7  {
8      const int M = 3;
9      double x[M], y[M], r[M];
10
11     for(int i=0; i<M; i++)
12     {
13         cout << "Ingrese el valor de x["<<i<<"]: "<<endl;
14         cin >> x[i];
15     }
16
17     for(int i=0; i<M; i++)
18     {
19         cout << "Ingrese el valor de y["<<i<<"]: "<<endl;
20         cin >> y[i];
21     }
22
23     //complete la llamada a la funcion suma_v
24     suma_v(x, y, r, M);
25
26     cout << endl << "La suma es: (";
27     for(int i=0; i<M; i++)
28     {
29         cout << r[i];
30         if(i!=M-1)
31             cout << ",";
32         else
33             cout << ")";
34     }
35 }
36
37 void suma_v(double u[], double v[], double s[], int N)
38 {
39     //complete el cuerpo de la funcion
40     for (int j=0; j<N;j++)
41         s[j]=u[j]+v[j];
42     return;
43 }
44

```

	Input	Expected	Got

	Input	Expected	Got
✓	-44.39 38.75 -20.84 -29.55 -2.88 16.42	Ingrese el valor de x[0]: Ingrese el valor de x[1]: Ingrese el valor de x[2]: Ingrese el valor de y[0]: Ingrese el valor de y[1]: Ingrese el valor de y[2]:  La suma es: (-73.94,35.87,-4.42)	Ingrese el valor de : Ingrese el valor de : Ingrese el valor de : Ingrese el valor de : Ingrese el valor de : Ingrese el valor de :  La suma es: (-73.94,35.87,-4.42)
✓	-11.82 28.21 -17.48 -30.1 -27.66 -22.82	Ingrese el valor de x[0]: Ingrese el valor de x[1]: Ingrese el valor de x[2]: Ingrese el valor de y[0]: Ingrese el valor de y[1]: Ingrese el valor de y[2]:  La suma es: (-41.92,0.55,-40.3)	Ingrese el valor de : Ingrese el valor de : Ingrese el valor de : Ingrese el valor de : Ingrese el valor de : Ingrese el valor de :  La suma es: (-41.92,0.55,-40.3)
✓	-12.22 679.54 292.48 887.91 158.2 115.78	Ingrese el valor de x[0]: Ingrese el valor de x[1]: Ingrese el valor de x[2]: Ingrese el valor de y[0]: Ingrese el valor de y[1]: Ingrese el valor de y[2]:  La suma es: (875.69,837.74,408.26)	Ingrese el valor de : Ingrese el valor de : Ingrese el valor de : Ingrese el valor de : Ingrese el valor de : Ingrese el valor de :  La suma es: (875.69,837.74,408.26)

Passed all tests! ✓

## Question author's solution:

```
#include <iostream>
using namespace std;

void suma_v(double u[], double v[], double s[], int N);

int main()
{
    const int M = 3;
    double x[M], y[M], r[M];

    for(int i=0; i<M; i++)
    {
        cout << "Ingrese el valor de x["<<i<<"]: "<<endl;
        cin >> x[i];
    }

    for(int i=0; i<M; i++)
    {
        cout << "Ingrese el valor de y["<<i<<"]: "<<endl;
        cin >> y[i];
    }

    suma_v(x,y,r,M);

    cout << endl << "La suma es: (";
    for(int i=0; i<M; i++)
    {
        cout << r[i];
        if(i!=M-1)
            cout << ",";
        else
            cout << ")";
    }
}

void suma_v(double u[], double v[], double s[], int N)
{
    for(int i=0; i<N; i++)
    {
        s[i] = u[i] + v[i];
    }
}
```

**Correct**

Marks for this submission: 1.00/1.00.

Question 13

Correct

Mark 1.00 out of 1.00

Un programador escribió el siguiente programa, pero no logra hacerlo compilar. Corrija el programa, modificando el perfil de la función inicializar\_matriz\_identidad de tal manera que el programa compile y supere el caso de prueba propuesto.

**For example:**

Result
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

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

Reset answer

```

1  #include <iostream>
2  using namespace std;
3
4  const int N = 5;
5
6  void inicializar_matriz_identidad(int matriz[N][N])
7  {
8      for(int i=0; i<N; i++)
9          for(int j=0; j<N; j++)
10             if (i==j)
11                 matriz[i][j] = 1;
12             else
13                 matriz[i][j] = 0;
14  }
15
16  int main()
17  {
18      int mat[N][N];
19
20      inicializar_matriz_identidad(mat);
21      for(int i=0; i<N; i++)
22      {
23          for(int j=0; j<N; j++)
24              cout<< mat[i][j] <<" ";
25          cout<<endl;
26      }
27
28  }
```

	Expected	Got	
✓	1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1	1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1	✓



**Question author's solution:**

```
#include <iostream>
using namespace std;

const int N = 5;

void inicializar_matriz_identidad(int matriz[][N])
{
    for(int i=0; i<N; i++)
        for(int j=0; j<N; j++)
            if (i == j)
                matriz[i][j] = 1;
            else
                matriz[i][j] = 0;
}

int main()
{
    int mat[N][N];

    inicializar_matriz_identidad(mat);
    for(int i=0; i<N; i++)
    {
        for(int j=0; j<N; j++)
            cout<< mat[i][j] <<" ";
        cout<<endl;
    }
}
```

**Correct**

Marks for this submission: 1.00/1.00.

Question 14

Correct

Mark 1.00 out of 1.00

Definir una función en C++ llamada `mayor` de tipo `float` que recibe como argumento un arreglo unidimensional de tipo flotante llamado *valores*, un entero llamado *N* que representa el número de elementos en el arreglo y entero por referencia llamado *pos*. La función debe encontrar el mayor de los valores del arreglo y devolverlo como valor de retorno, a su vez debe guardar en *pos* el índice donde se encuentra el mayor. En caso de que el mayor valor se encuentre en más de una posición devolver la más chica.

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

Reset answer

```
1 float mayor(float valores[], int N, int &pos)
2 {
3     pos=0;
4     float may=valores[0];
5     for(int i=1;i<N;i++)
6     {
7         if (may<valores[i])
8         {
9             may=valores[i];
10            pos=i;
11        }
12    }
13
14    return may;
15 }
```

	Test
✓	<pre>float M[8] = {111.34, 55.01, 11.01, 48.37, 62.93, 21.89, 11.84, int ind; float may = mayor(M,8,ind); cout &lt;&lt; "El mayor es M[" &lt;&lt; ind &lt;&lt; "]: " &lt;&lt; may;</pre>
✓	<pre>float M[12] = {-44.63, -77.54, 107.71, 106.52, -77.38, -56.89, int ind; float may = mayor(M,12,ind); cout &lt;&lt; "El mayor es M[" &lt;&lt; ind &lt;&lt; "]: " &lt;&lt; may;</pre>
✓	<pre>float M[6] = {-79.02, -79.14, -79.41, -79.92, -79.42, -79.89}; int ind; float may = mayor(M,6,ind); cout &lt;&lt; "El mayor es M[" &lt;&lt; ind &lt;&lt; "]: " &lt;&lt; may;</pre>

Passed all tests! ✓

**Question author's solution:**

```
float mayor(float valores[], int N, int &pos)
{
    float m = valores[0];
    pos = 0;
    for(int i=0; i<N; i++)
    {
        if(valores[i]>m)
        {
            m = valores[i];
            pos = i;
        }
    }
    return m;
}
```

**Correct**

Marks for this submission: 1.00/1.00.

## Question 15

Correct

Mark 1.00 out of  
1.00

Escriba la definición de una función en C++ llamada minMax de tipo void, que acepta como argumento un arreglo unidimensional llamado A, un entero llamado N que representa la cantidad de elementos en el e arreglo A y dos enteros por referencia m y M. La función debe encontrar el mayor y el menor de los valores de A y los almacene en m y M.

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

Reset answer

```

1 void minMax(int A[], int N, int &m, int &M)
2 {
3     int min=A[0];
4     for(int i=1;i<N;i++)
5     {
6         if (min>A[i])
7             min=A[i];
8     }
9     m=min;
10
11     int may=A[0];
12     for(int i=1;i<N;i++)
13     {
14         if (may<A[i])
15             may=A[i];
16     }
17     M=may;
18     return;
19 }
20
21 }
```

	Test
✓	<pre>int T[12] = {990, -7, 1414, 1480, 1491, 1259, 308, 294, 1113, 90 int men, may; minMax(T,12,men,may); cout &lt;&lt; men &lt;&lt; " " &lt;&lt; may;</pre>
✓	<pre>int T[8] = {470, 1382, 1257, 1037, -422, -102, 333, 518}; int men, may; minMax(T,8,men,may); cout &lt;&lt; men &lt;&lt; " " &lt;&lt; may;</pre>
✓	<pre>int T[11] = {-498, -492, -498, -498, -493, -491, -492, -491, -49 int men, may; minMax(T,11,men,may); cout &lt;&lt; men &lt;&lt; " " &lt;&lt; may;</pre>

	Test
✓	<pre>int T[25] = {13, 45, -46, 45, 34, -42, 9, 0, -14, 12, -21, 34, ' int men, may; minMax(T,25,men,may); cout &lt;&lt; men &lt;&lt; " " &lt;&lt; may; minMax(T,12,men,may); cout &lt;&lt; men &lt;&lt; " " &lt;&lt; may; minMax(T,15,men,may); cout &lt;&lt; men &lt;&lt; " " &lt;&lt; may; minMax(T,5,men,may); cout &lt;&lt; men &lt;&lt; " " &lt;&lt; may;</pre>

Passed all tests! ✓

### Question author's solution:

```
void minMax(int A[], int N, int &m, int &M)
{
    m=M=A[0];
    for(int i=0; i<N; i++)
    {
        if(m>A[i])
        {
            m = A[i];
        }
        if(M<A[i])
        {
            M = A[i];
        }
    }
}
```

**Correct**

Marks for this submission: 1.00/1.00.

Question 16

Correct

Mark 1.00 out of 1.00

La **mediana** representa el valor del elemento de posición central en un conjunto de datos ordenados y dada una colección ordenada de datos  $X_0, X_1, \dots, X_n$  se define de la siguiente manera:

- Si  $N$  es impar, la mediana es  $X_{n/2}$

$$\frac{X_{\frac{n}{2} - 1} + X_{\frac{n}{2}}}{2}$$

- Si la colección es par, la mediana es el valor

A un estudiante de informática se le dio la siguiente consigna:

Escriba un programa que solicite se ingresen valores enteros, hasta que el usuario ingrese 0 o supere la cantidad máxima de 15 valores permitidos. Al finalizar su programa debe imprimir el valor de la mediana e informar si la misma es o no un elemento de la colección.

El estudiante dio la siguiente implementación. En la misma modularizó el código de manera correcta. Sin embargo, su instructor le marco los siguientes errores:

- La idea de su solución esta clara. Sin embargo, ¿la mediana de una colección de números enteros es siempre un numero entero? Considere modificar el tipo de retorno de la función calcular\_mediana y modifique los tipos de datos utilizados para operar de manera correcta. **No olvide que la operación de datos enteros devuelve valores enteros.**
- Usted dice que en el argumento pos\_mediana de la función calcular\_mediana almacena el valor de la posición en la que se encuentra la mediana o -1 en caso de que la mediana no sea un elemento de la colección. Le parece que será visible ese dato desde quién invoque a la función. **Considere utilizar pasaje de parámetros por referencia.**

Lea atentamente las sugerencias del instructor y corrija el programa.

Para reflexionar

En las últimas prácticas usted tuvo que leer códigos de cierta complejidad escritos por otro programador, ¿Que opina sobre la importancia de escribir comentarios y definir nombres significativos para las variables y funciones?

**For example:**

Input	Result
5 8 3 1 0	La mediana es: 4 pero no es un elemento de la coleccion.
19 13 8 0	La mediana es: 13 y es un elemento de la coleccion.

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

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  /*
6   Dado un arreglo de datos y la cantidad de elementos
7   a explorar, esta función calcula el valor de la mediana
8   y lo devuelve como valor de retorno. Además, en el
9   argumento pos_mediana, almacena la posición en la que
10  se encuentra el valor de la mediana si la mediana pertenece
11  a la colección o -1 en caso contrario.
12  */
13  float calcular_mediana(int datos[], int N, int &pos_mediana)
14  {
15      //ordeno el arreglo datos usando ordenamiento burbuja
16      //ordeno el arreglo datos usando ordenamiento burbuja
17      for (int i=0; i<N-1; i++)
18          for (int j=1; j < N - i; j++)
19              if (datos[j-1] > datos[j])
20              {
21                  int temp = datos[j];
22                  datos[j] = datos[j-1];
23                  datos[j-1] = temp;
24              }
25
26      //calculo mediana
27      if(N%2 == 1)
28      {
29          //Si la colección de datos es impar es el valor del medio
30          pos_mediana = N/2;
31          return datos[pos_mediana];
32      }else{
33          //Sino es el promedio de los dos valores del medio
34          float mediana = (datos[N/2 - 1] + datos[N/2]) / 2.0;
35          if (mediana == datos[N/2])
36              pos_mediana = N/2;
37          else
38              pos_mediana = -1;
39          return mediana;
40      }
41  }
42
43  int main()
44  {
45      const int N = 15;
46      int nums[N];
47      double mediana;
48      int i = 0, num;
49      int mediana_pos = 0;
50
51      //solicito datos al usuario
52      cin>>num;
53      while(num!=0 && i<N)
54      {
55          nums[i] = num;
56          i++;
57          cin>>num;
58      }
```

	Input	Expected
✓	9 3 5 8 1 7 2 6 4 0	La mediana es: 5 y es un elemento de la coleccion.
✓	0	No se ingresaron datos
✓	2 6 5 8 0	La mediana es: 5.5 pero no es un elemento de la coleccion.
✓	2 4 2 2 -8 -3 -10 -5 3 0	La mediana es: 2 y es un elemento de la coleccion.



	Input	Expected
✓	452 771 188 165 960 875 172 142 32 211 928 586 157 191 880	La mediana es: 211 y es un elemento de la coleccion.
✓	-789 61 579 858 -526 259 492 -471 860 -559 -421 104 296 432 -449	La mediana es: 104 y es un elemento de la coleccion.
✓	-531 829 -124 -950 512 -822 173 78 624 -749 675 -917 601 0	La mediana es: 78 y es un elemento de la coleccion.
✓	5 8 3 1 0	La mediana es: 4 pero no es un elemento de la coleccion.

	Input	Expected
✓	10 9 0	La mediana es: 9.5 pero no es un elemento de la coleccion.
✓	10 0	La mediana es: 10 y es un elemento de la coleccion.
✓	19 13 8 0	La mediana es: 13 y es un elemento de la coleccion.

Passed all tests! ✓

**Question author's solution:**

```

#include <iostream>
#include <cmath>
using namespace std;

/*
    Dado un arreglo de datos y la cantidad de elementos
    a explorar, esta función calcula el valor de la mediana
    y lo devuelve como valor de retorno. Además, en el
    argumento pos_mediana, almacena la posición en la que
    se encuentra el valor de la mediana si la mediana per
    tenece
    a la colección o -1 en caso contrario.
*/
double calcular_mediana(int datos[], int N, int& pos_mediana)
{
    //ordeno el arreglo datos usando ordenamiento burbuja
    //ordeno el arreglo datos usando ordenamiento bur
    buja
    for (int i=0; i<N-1; i++)
        for (int j=1 ; j < N - i; j++)
            if (datos[j-1] > datos[j])
            {
                int temp = datos[j];
                datos[j] = datos[j-1];
                datos[j-1] = temp;
            }

    //calculo mediana
    if(N%2 == 1)
    {
        //Si la coleccion de datos es impar es el valor d
        el medio
        pos_mediana = N/2;
        return datos[pos_mediana];
    }else{
        //Sino es el promedio de los dos valores del medi
        o
        double mediana = (datos[N/2 - 1] + datos[N/2]) / 2.0;
        if (mediana == datos[N/2])
            pos_mediana = N/2;
        else
            pos_mediana = -1;
        return mediana;
    }
}

int main()
{
    const int N = 15;
    int nums[N];
    double mediana;
    int i = 0, num;
    int mediana_pos;

```

```
//solicito datos al usuario
cin>>num;
while(num!=0 && i<N)
{
    nums[i] = num;
    i++;
    cin>>num;
}

if (i == 0)
    //El usuario no ingreso datos
    cout<<"No se ingresaron datos"<<endl;
else
{
    //Calculo mediana e informo resultados
    mediana = calcular_mediana(nums, i, mediana_pos);
    if (mediana_pos != -1)
        cout<<"La mediana es: "<< mediana <<" y es un e
lemento de la coleccion."<<endl;
    else
        cout<<"La mediana es: "<< mediana <<" pero no es un e
lemento de la coleccion."<<endl;
    }
}
```

**Correct**

Marks for this submission: 1.00/1.00.

## Question 17

Correct

Mark 1.00 out of  
1.00

Complete el siguiente programa. El programa solicita se ingrese por teclado la cantidad de filas y columnas de una matriz cuadrada y se almacena en n. n se valida para que sea mayor que cero y menor o igual que N=10. Luego se debe rellenar la matriz llamada mat con valores flotantes ingresados por teclado. Se muestra la matriz por pantalla en formato de matriz. Se extrae la diagonal principal y se la muestra por pantalla. Finalmente se calcula la media y la desviación estandar de los valores de la diagonal.

Complete el cuerpo de la función est, para que calcule la media y la desviación de los n valores almacenados en el arreglo y los almacene en las variables media y desviacion.

Complete el programa principal con las llamadas a función.

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

Reset answer

```

1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5
6  const int N = 10;
7
8  int tamaño()
9  {
10     int n;
11     cout << "Ingrese n: " << endl;
12     cin >> n;
13     while(n<1 || n>N)
14     {
15         cout << "Ingrese n: " << endl;
16         cin >> n;
17     }
18     return n;
19 }
20
21 void rellenar_matriz(float M[N][N], int n)
22 {
23     for(int i=0; i<n; i++)
24     {
25         for(int j=0; j<n; j++)
26         {
27             cout << "Ingrese el elemento "<< i << ", " << j << ": "
28             cin >> M[i][j];
29         }
30     }
31 }
32
33 void mostrar_matriz(float M[N][N], int n)
34 {
35     for(int i=0; i<n; i++)
36     {
37         for(int j=0; j<n; j++)
38         {
39             cout << " " << setw(6) << M[i][j];
40         }
41         cout << endl;
42     }
43     return;

```



	Input	Expected
✓	-1 -1 11 3 390.66 912.64 768.07 34.98 367.99 239.81 766.37 0.18 889.1	Ingrese n: Ingrese n: Ingrese n: Ingrese n: Ingrese el elemento 0, 0: Ingrese el elemento 0, 1: Ingrese el elemento 0, 2: Ingrese el elemento 1, 0: Ingrese el elemento 1, 1: Ingrese el elemento 1, 2: Ingrese el elemento 2, 0: Ingrese el elemento 2, 1: Ingrese el elemento 2, 2: <div>                         390.66      912.64      768.07                          34.98      367.99      239.81                          766.37      0.18      889.1                     </div> Diagonal: (390.66,367.99,889.1) La media es: 549.25 y la desviacion es: 240.488
✓	-1 -1 0 0 0 4 -20.89 24.53 23.13 43.51 -35.55 37.72 -39.77 7.03 -7.11 -34.36 -15.49 35.84 48.94 -46.47 12.08 -46.22	Ingrese n: Ingrese n: Ingrese n: Ingrese n: Ingrese n: Ingrese n: Ingrese el elemento 0, 0: Ingrese el elemento 0, 1: Ingrese el elemento 0, 2: Ingrese el elemento 0, 3: Ingrese el elemento 1, 0: Ingrese el elemento 1, 1: Ingrese el elemento 1, 2: Ingrese el elemento 1, 3: Ingrese el elemento 2, 0: Ingrese el elemento 2, 1: Ingrese el elemento 2, 2: Ingrese el elemento 2, 3: Ingrese el elemento 3, 0: Ingrese el elemento 3, 1: Ingrese el elemento 3, 2: Ingrese el elemento 3, 3: <div>                         -20.89      24.53      23.13      43.51                          -35.55      37.72      -39.77      7.03                          -7.11      -34.36      -15.49      35.84                          48.94      -46.47      12.08      -46.22                     </div> Diagonal: (-20.89,37.72,-15.49,-46.22) La media es: -11.22 y la desviacion es: 30.5445

	Input	Expected	http://lev2.efn.uncor.edu/mod/quiz/review.p			
✓	11	Ingrese n:				
	11	Ingrese n:				
	11	Ingrese n:				
	6	Ingrese n:				
	2.43	Ingrese el elemento 0, 0:				
	36.63	Ingrese el elemento 0, 1:				
	30.17	Ingrese el elemento 0, 2:				
	-28.91	Ingrese el elemento 0, 3:				
	23.65	Ingrese el elemento 0, 4:				
	-45.22	Ingrese el elemento 0, 5:				
	24.26	Ingrese el elemento 1, 0:				
	-29.49	Ingrese el elemento 1, 1:				
	-19.33	Ingrese el elemento 1, 2:				
	-15.59	Ingrese el elemento 1, 3:				
	40.74	Ingrese el elemento 1, 4:				
	1.57	Ingrese el elemento 1, 5:				
	-20.14	Ingrese el elemento 2, 0:				
	-36.03	Ingrese el elemento 2, 1:				
	0.15	Ingrese el elemento 2, 2:				
	38.16	Ingrese el elemento 2, 3:				
	-6.56	Ingrese el elemento 2, 4:				
	32.19	Ingrese el elemento 2, 5:				
	43.65	Ingrese el elemento 3, 0:				
	-31.24	Ingrese el elemento 3, 1:				
	-28.27	Ingrese el elemento 3, 2:				
	-49.65	Ingrese el elemento 3, 3:				
	-47.36	Ingrese el elemento 3, 4:				
	-7.31	Ingrese el elemento 3, 5:				
	-32.31	Ingrese el elemento 4, 0:				
	-0.05	Ingrese el elemento 4, 1:				
	-21.12	Ingrese el elemento 4, 2:				
	-27.91	Ingrese el elemento 4, 3:				
	22.9	Ingrese el elemento 4, 4:				
	41.25	Ingrese el elemento 4, 5:				
	-0.36	Ingrese el elemento 5, 0:				
	-32.7	Ingrese el elemento 5, 1:				
	-15.45	Ingrese el elemento 5, 2:				
	40.63	Ingrese el elemento 5, 3:				
	21.87	Ingrese el elemento 5, 4:				
	46.11	Ingrese el elemento 5, 5:				
			2.4336.6330.17-28.9123.65-4			
		24.26-29.49-19.33-15.5940.74				
		-20.14-36.030.1538.16-6.563				
		43.65-31.24-28.27-49.65-47.36-				
		-32.31-0.05-21.12-27.9122.94				
		-0.36-32.7-15.4540.6321.874				
		Diagonal: (2.43,-29.49,0.15,-49.65,22.9,46.11)				
		La media es: -1.25833 y la desviacion es: 31.5747				





```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;

const int N = 10;

int tamanio()
{
    int n;
    cout << "Ingrese n: " << endl;
    cin >> n;
    while(n<1 || n>N)
    {
        cout << "Ingrese n: " << endl;
        cin >> n;
    }
    return n;
}

void rellenar_matriz(float M[N][N], int n)
{
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<n; j++)
        {
            cout << "Ingrese el elemento " << i << ", " << j << "
: "<< endl;
            cin >> M[i][j];
        }
    }
}

void mostrar_matriz(float M[N][N], int n)
{
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<n; j++)
        {
            cout << setw(10) << M[i][j];
        }
        cout << endl;
    }
}

void extrae_diagonal(float M[N][N], float D[N], int n)
{
    for(int i=0; i<n; i++)
    {
        D[i] = M[i][i];
    }
}
```

```
void est(float A[N], int n, double &media, double &desviacion)
{
    media = 0.0;
    for(int i=0; i<n; i++)
    {
        media += A[i];
    }
    media /= n;

    desviacion = 0.0;
    for(int i=0; i<n; i++)
    {
        desviacion += pow(A[i] - media,2.0);
    }
    desviacion /= n;
    desviacion = sqrt(desviacion);
}

int main()
{
    float mat[N][N];
    float diag[N];
    double media, desv;
    int n;
    n = tamanio();

    rellenar_matriz(mat, n);
    mostrar_matriz(mat, n);
    extrae_diagonal(mat, diag, n);

    cout << endl << "Diagonal: (";
    for(int i=0; i<n; i++)
    {
        cout << diag[i];
        if(i!=n-1)
            cout << ",";
        else
            cout << ")" << endl;
    }

    est(diag, n, media, desv);

    cout << "La media es: " << media << " y la desviacion es: " <
< desv;

}
```

**Correct**

Marks for this submission: 1.00/1.00.

Question 18

Correct

Mark 1.00 out of 1.00

A un programador le dieron la siguiente consigna:

Escriba un programa que solicite **un numero entero** que indica la **cantidad de jugadores en un torneo** y luego solicite los **puntos alcanzados por cada jugador**. Los jugadores son identificados por un numero [0,...,n-1] donde n es la cantidad de jugadores y los puntos deben ser solicitados en ese orden. Es decir, primero deben solicitarse los puntos del jugador 0, luego los del 1 y al último los de n-1. Considere que como máximo podrá haber 10 jugadores.  
Al finalizar el programa deberá escribir una tabla de posiciones, dónde liste los jugadores y los puntos obtenidos tal como se muestra en el siguiente ejemplo de ejecución:

```
Ingrese cantidad de jugadores
4
Ingrese los puntos obtenidos por el jugador 0
1
Ingrese los puntos obtenidos por el jugador 1
2
Ingrese los puntos obtenidos por el jugador 2
3
Ingrese los puntos obtenidos por el jugador 3
4

Tabla de posiciones
1- Jugador 3  4 pts
2- Jugador 2  3 pts
3- Jugador 1  2 pts
4- Jugador 0  1 pts
```

Para cumplir con la consigna, el programador dividió el problema en sub-problemas y definió funciones que se encargan de resolver las distintas partes del mismo.

A continuación usted puede ver la solución que propuso el programador. Sin embargo solo se le provee un bosquejo del programa principal. Deberá completar el programa principal utilizando las funciones definidas para resolver el problema respetando la propuesta de solución del programador.

**For example:**

Input	Result
4	Ingrese cantidad de jugadores
1	Ingrese los puntos obtenidos por el jugador 0
2	Ingrese los puntos obtenidos por el jugador 1
3	Ingrese los puntos obtenidos por el jugador 2
4	Ingrese los puntos obtenidos por el jugador 3
	Tabla de posiciones
	1- Jugador 3  4 pts
	2- Jugador 2  3 pts
	3- Jugador 1  2 pts
	4- Jugador 0  1 pts

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

Reset answer

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
```



	Input	Expected	Got
✓	4 1 2 3 4	Ingrese cantidad de jugadores Ingrese los puntos obtenidor por el jugador 0 Ingrese los puntos obtenidor por el jugador 1 Ingrese los puntos obtenidor por el jugador 2 Ingrese los puntos obtenidor por el jugador 3	Ingrese can Ingrese los Ingrese los Ingrese los Ingrese los
		Tabla de posiciones 1- Jugador 3 4 pts 2- Jugador 2 3 pts 3- Jugador 1 2 pts 4- Jugador 0 1 pts	Tabla de po 1- Jugador 2- Jugador 3- Jugador 4- Jugador
✓	-10 0 1 23	Ingrese cantidad de jugadores Ingrese los puntos obtenidor por el jugador 0 Tabla de posiciones 1- Jugador 0 23 pts	Ingrese can Ingrese los Tabla de po 1- Jugador

	Input	Expected	Output
✓	10 10 20 5 45 53 8 7 13 16 25	Ingrese cantidad de jugadores Ingrese los puntos obtenidor por el jugador 0 Ingrese los puntos obtenidor por el jugador 1 Ingrese los puntos obtenidor por el jugador 2 Ingrese los puntos obtenidor por el jugador 3 Ingrese los puntos obtenidor por el jugador 4 Ingrese los puntos obtenidor por el jugador 5 Ingrese los puntos obtenidor por el jugador 6 Ingrese los puntos obtenidor por el jugador 7 Ingrese los puntos obtenidor por el jugador 8 Ingrese los puntos obtenidor por el jugador 9  Tabla de posiciones 1- Jugador 4 53 pts 2- Jugador 3 45 pts 3- Jugador 9 25 pts 4- Jugador 1 20 pts 5- Jugador 8 16 pts 6- Jugador 7 13 pts 7- Jugador 0 10 pts 8- Jugador 5 8 pts 9- Jugador 6 7 pts 10- Jugador 2 5 pts	Ingrese can Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los  Tabla de po 1- Jugador 2- Jugador 3- Jugador 4- Jugador 5- Jugador 6- Jugador 7- Jugador 8- Jugador 9- Jugador 10- Jugador
✓	3 -10 -1 0 8 5	Ingrese cantidad de jugadores Ingrese los puntos obtenidor por el jugador 0 Ingrese los puntos obtenidor por el jugador 0 Ingrese los puntos obtenidor por el jugador 0 Ingrese los puntos obtenidor por el jugador 1 Ingrese los puntos obtenidor por el jugador 2  Tabla de posiciones 1- Jugador 1 8 pts 2- Jugador 2 5 pts 3- Jugador 0 0 pts	Ingrese can Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los  Tabla de po 1- Jugador 2- Jugador 3- Jugador

	Input	Expected	Output
✓	11 10 33 23 46 60 10 12 18 21 23 0	Ingrese cantidad de jugadores Ingrese los puntos obtenidor por el jugador 0 Ingrese los puntos obtenidor por el jugador 1 Ingrese los puntos obtenidor por el jugador 2 Ingrese los puntos obtenidor por el jugador 3 Ingrese los puntos obtenidor por el jugador 4 Ingrese los puntos obtenidor por el jugador 5 Ingrese los puntos obtenidor por el jugador 6 Ingrese los puntos obtenidor por el jugador 7 Ingrese los puntos obtenidor por el jugador 8 Ingrese los puntos obtenidor por el jugador 9 Tabla de posiciones 1- Jugador 3 60 pts 2- Jugador 2 46 pts 3- Jugador 0 33 pts 4- Jugador 1 23 pts 5- Jugador 8 23 pts 6- Jugador 7 21 pts 7- Jugador 6 18 pts 8- Jugador 5 12 pts 9- Jugador 4 10 pts 10- Jugador 9 0 pts	Ingrese can Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Tabla de po 1- Jugador 2- Jugador 3- Jugador 4- Jugador 5- Jugador 6- Jugador 7- Jugador 8- Jugador 9- Jugador 10- Jugador
✓	11 -1 2 -1 -10 5 -1 -10 6	Ingrese cantidad de jugadores Ingrese los puntos obtenidor por el jugador 0 Ingrese los puntos obtenidor por el jugador 0 Ingrese los puntos obtenidor por el jugador 0 Ingrese los puntos obtenidor por el jugador 1 Ingrese los puntos obtenidor por el jugador 1 Ingrese los puntos obtenidor por el jugador 1 Tabla de posiciones 1- Jugador 1 6 pts 2- Jugador 0 5 pts	Ingrese can Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Ingrese los Tabla de po 1- Jugador 2- Jugador

Passed all tests! ✓

### Question author's solution:



```
#include <iostream>
#include <iomanip>
using namespace std;

const int CANT_MAX_JUGADORES = 10;

/*
Inicaliza el arreglo jugadores pasado como parámetro haciendo
jugadores[i] = i
*/
void inicializar_arreglo_jugadores(int jugadores[], int cant_juga
dores)
{
    for(int i=0; i<cant_jugadores; i++)
        jugadores[i] = i;
}

/*
Carga los puntos obtenidos por cada jugador
en el arreglo puntos
*/
void cargar_puntos(int puntos[], int cant_jugadores)
{
    for(int i=0; i<cant_jugadores; i++)
        do{
            cout<<"Ingrese los puntos obtenidor por el jugador "<
< i <<endl;
            cin>>puntos[i];
        }while(puntos[i]<0);
}

/*
Intercambiar valores entre n y m
*/
void intercambiar(int& n, int& m)
{
    int temp = n;
    n = m;
    m = temp;
}

/*
    Ordena el arreglo puntos de manera decreciente.
    Simultaneamente actualiza el arreglo jugadores,
    de manera tal que puntos[i] sean los puntos del
    jugador con identificador jugador[i]
*/
void ordenar_tabla_por_puntos(int puntos[],int jugadores[], int c
antidad)
{
    for (int i=0; i<cantidad-1; i++)
        for (int j=1 ; j < cantidad - i; j++)
            if (puntos[j-1] < puntos[j])
```

```
        {
            http://lev2.efn.uncor.edu/mod/quiz/review.php?a...
            intercambiar(puntos[j], puntos[j-1]);
            intercambiar(jugadores[j], jugadores[j-1]);
        }
    }

int main()
{
    int cant_jugadores;
    int jugadores[CANT_MAX_JUGADORES], puntos[CANT_MAX_JUGADORES]
;

    //Solicito cantidad de jugadores
    cout<<"Ingrese cantidad de jugadores "<<endl;
    do
        cin>>cant_jugadores;
    while(cant_jugadores<=0 || cant_jugadores>CANT_MAX_JUGADORES)
;

    //Incializo arreglo con identificador de jugadores
    inicializar_arreglo_jugadores(jugadores, cant_jugadores);

    //Pido al usuario que ingrese los puntos
    cargar_puntos(puntos, cant_jugadores);

    //Ordeno los jugadores de acuerdo a sus puntuaciones
    ordenar_tabla_por_puntos(puntos, jugadores, cant_jugadores);

    //Imprimimo la tabla de posiciones
    cout<<endl<<"Tabla de posiciones"<<endl;
    for(int i=0; i<cant_jugadores; i++)
        cout<< setw(2) << i+1 <<"- Jugador "<<jugadores[i]<<" "<
< puntos[i] <<" pts"<<endl;
}
```

**Correct**

Marks for this submission: 1.00/1.00.

Question 19

Correct

Mark 1.00 out of 1.00

```

1  #include <iostream>
2  using namespace std;
3
4
5  int serie()
6  {
7      static int i1=0;
8      static int i2=1;
9      int t;
10     t = i2;
11     i2 = i1 + i2;
12     i1 = t;
13     return i2;
14 }
15
16
17 int main()
18 {
19     for(int i=0; i<10; i++)
20     {
21         cout << serie() << " ";
22     }
23
24     cout << endl;
25     for(int i=0; i<10; i++)
26     {
27         cout << serie() << " ";
28     }
29
30     return 0;
31 }
32

```

Select one or more:

- ☐ a. La variable t conserva su valor entre llamadas sucesivas
- ☐ b. Las variables t, i1 e i2 de la función serie se pueden alcanzar desde el cuerpo de la función main
- ☒ c. Las variables i1 e i2 de la función serie se inicializan únicamente en la primera llamada a la función serie y luego conservan su valor en llamadas sucesivas. ✓
- ☒ d. Sale por pantalla:  
**1 2 3 5 8 13 21 34 55 89**  
**144 233 377 610 987 1597 2584 4181 6765 10946** ✓
- ☐ e. Sale por pantalla:  
**1 1 1 1 1 1 1 1 1 1**  
**1 1 1 1 1 1 1 1 1 1**
- ☐ f. Sale por pantalla:  
**1 2 3 5 8 13 21 34 55 89**  
**1 2 3 5 8 13 21 34 55 89**

Respuesta correcta

The correct answers are: Sale por pantalla:

**1 2 3 5 8 13 21 34 55 89**

**144 233 377 610 987 1597 2584 4181 6765 10946**, Las variables i1 e i2 de la función serie se inicializan únicamente en la primera llamada a la función serie y luego conservan su valor en llamadas sucesivas.

27/10/18 13:30

Marks for this submission: 1.00/1.00.

## Question 20

Correct

Mark 1.00 out of  
1.00

La **recursión** puede ser considerada una técnica de programación, la cual implica resolver un problema dando la solución para la/s instancias mas simples del problema (**caso/s base**) y un procedimiento por el cuál los casos generales se reducen a los casos bases (**caso/s inductivo/s**). Esta reducción implica que en el caso inductivo aparezcan invocaciones a la función que se esta definiendo. Esas invocaciones deberán resolver instancias mas pequeñas del problema inicial.

Un ejemplo trivial de recursión que resulta útil para comprender el concepto, es la definición recursiva de la función suma de números naturales. Un razonamiento recursivo para esta función sería:

- **Caso Base:** La suma entre 0 y cualquier otro numero m es m.
- **Caso Inductivo:** La suma entre n y m con  $n > 0$ , es la suma entre el predecesor de n y el sucesor de m.

Un error típico a la hora de codificar una solución recursiva, es el de definir de manera errónea el caso inductivo y generar una recursión infinita. Particularmente, este es el problema que se observa en el código de ejemplo. Corrija el código de ejemplo para que funcione de manera correcta.

**For example:**

Test	Result
<code>cout&lt;&lt;suma(0,5)&lt;&lt;endl;</code>	5
<code>cout&lt;&lt;suma(1,3)&lt;&lt;endl;</code>	4

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

Reset answer

```

1 | int suma(int n, int m)
2 | {
3 |     if (n==0)
4 |         return m;
5 |     else
6 |         return suma(n-1, m+1);
7 | }
```

	Test	Expected	Got	
✓	<code>cout&lt;&lt;suma(0,5)&lt;&lt;endl;</code>	5	5	✓

	Test	Expected	Got	
✓	cout<<suma(1,1)<<endl;	2	2	✓
✓	cout<<suma(1,3)<<endl;	4	4	✓
✓	cout<<suma(2,5)<<endl;	7	7	✓
✓	cout<<suma(3,2)<<endl;	5	5	✓

Passed all tests! ✓

### Question author's solution:

```
int suma(int n, int m)
{
    if (n==0)
        return m;
    else
        return suma(--n, ++m);
}
```

**Correct**

Marks for this submission: 1.00/1.00.

◀ Algoritmo de ordenamiento por selección

Jump to...

Guía de Ejercicios 10 ▶