

Área personal ► Mis cursos ► InfoC++ ► Clase 8 ► TP2

<b>Comenzado el</b>	jueves, 4 de octubre de 2018, 22:37
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	domingo, 14 de octubre de 2018, 20:36
<b>Tiempo empleado</b>	9 días 21 horas
<b>Calificación</b>	20 de 20 (100%)

**Pregunta 1**

Correcta

Puntúa 1 sobre 1

A partir de laño 1983 en Argentina se ha vivido bajo democracia. Los periodos presidenciales antes del año 1994 eran de 6 años pero en dicho año pasaron a ser de 4 años. El año 1983 y el año 1995 fueron de elecciones presidenciales.

Sabiendo esto, realice un programa que solicite que se ingrese por teclado un numero entero que representa a un año y en función del valor ingresado muestre por pantalla uno de los siguientes mensajes:

- "Elecciones presidenciales" si:

- El año ingresado es 1983 o 1995
- El año es mayor a 1983, menor a 1995 y la cantidad de años que pasaron desde 1983 es múltiplo de 6
- El año es mayor a 1995 y la cantidad de años que pasaron desde 1995 es múltiplo de 4

**Por ejemplo:**

Input	Resultado
1983	Elecciones presidenciales
1995	Elecciones presidenciales
1989	Elecciones presidenciales
2003	Elecciones presidenciales

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

Reiniciar respuesta

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a;
7
8      cin>>a;
9      if(((a==1995) || (a==1983)) || ((a>1983) && (a<1995) && ((a-1983)%6==
10         cout<<"Elecciones presidenciales"<<endl;
11
12         return 0;
13     }
```

	Input	Expected	Got	
✓	1983	Elecciones presidenciales	Elecciones presidenciales	✓

	Input	Expected	Got	
✓	1995	Elecciones presidenciales	Elecciones presidenciales	✓
✓	1989	Elecciones presidenciales	Elecciones presidenciales	✓
✓	2003	Elecciones presidenciales	Elecciones presidenciales	✓
✓	2011	Elecciones presidenciales	Elecciones presidenciales	✓
✓	2012			✓
✓	4000			✓
✓	12384			✓
✓	2011	Elecciones presidenciales	Elecciones presidenciales	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1/1.

Este ejercicio debe realizarse utilizando solamente sentencias if

El programa debe ordenar de mayor a menor 3 números ingresados por teclado y asignados a las variables x, y, z.

**IMPORTANTE:** La primera corrección de su programa es automática, por consiguiente su programa debe imprimir EXACTAMENTE lo que la consigna del ejercicio y los ejemplos de ejecución sugieren en caso contrario los test no pasarán. No olvide considerar espacios en blancos, saltos de línea, minúsculas y mayúsculas.

**Por ejemplo:**

Input	Resultado
3.1	7
7	3.1
2	2
10	10
2.1	5
5	2.1
3	11
5.1	5.1
11	3

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

```

2  using namespace std;
3
4  int main()
5  {
6      float x, y, z, max, min, med;
7
8      cin>>x;
9      cin>>y;
10     cin>>z;
11     max=x;
12     if(y>x&&y>z)
13         max=y;
14     if(z>x&&z>y)
15         max=z;
16     min=x;
17     if(x>y&&z>y)
18         min=y;
19     if(x>z&&y>z)
20         min=z;
21     med=x;
22     if(y<max&&y>min)
23         med=y;
24     if(z<max&&z>min)
25         med=z;
26     cout<<max<<endl;
27     cout<<med<<endl;
28     cout<<min<<endl;
29     return 0;
30 }
```

	Input	Expected	Got	
✓	3.1 7 2	7 3.1 2	7 3.1 2	✓
✓	10 2.1 5	10 5 2.1	10 5 2.1	✓
✓	3 5.1 11	11 5.1 3	11 5.1 3	✓

Todas las pruebas superadas. ✓

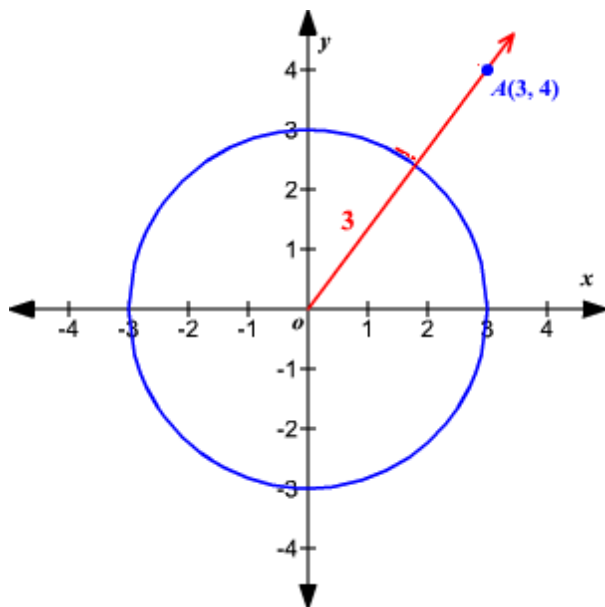
**Correcta**

Puntos para este envío: 1/1.

Se debe realizar un programa en C++ que determina si un punto se encuentra dentro del círculo de radio 3 y centro (0,0), respondiendo "Pertenece" si está dentro y "No pertenece" si está fuera.

El programa debe solicitar las coordenadas (x,y) del punto. Respetar este orden de ingreso de los datos.

**Pista:** Se debe trabajar con la distancia entre el centro y el punto. Hacer gráficos en papel de ser necesario.



Por ejemplo:

Test	Input	Resultado
// No Pertenece	4.0 3.0	No pertenece
// Pertenece	1.9 2.1	Pertenece

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

Reiniciar respuesta

```

1 // Complete el código fuente
2
3 // No ponga texto en el cin
4 #include <iostream>
5 #include <cmath>
6 using namespace std;
7
8 int main () {
9     float x, y,d;
10
11     cin>>x;
12     cin>>y;
13
14     d= sqrt(pow(x,2)+pow(y, 2));
15     if(d<=3)
16         cout<<"Pertenece" <<endl;
17     else
18         cout<<"No pertenece" <<endl;
19     return 0;
20 }
```

	Test	Input	Expected	Got	
✓	// No Pertenece	4.0 3.0	No pertenece	No pertenece	✓
✓	// Pertenece	1.9 2.1	Pertenece	Pertenece	✓
✓	// Pertenece	-1.1 -2.2	Pertenece	Pertenece	✓
✓	// No Pertenece	-2.9 -2.7	No pertenece	No pertenece	✓

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Un número **cuadrado perfecto** es un número entero que es el cuadrado de algún otro; dicho de otro modo, es un número cuya raíz cuadrada es un número natural.

Un número es un cuadrado perfecto si se puede ordenar en una figura cuadrada. Por ejemplo, 9 es un número cuadrado perfecto ya que puede ser escrito como  $3 \times 3$ , y se puede ordenar del siguiente modo:

$$3^2 = 9$$


Realizar un programa que solicite que se ingrese un numero entero por teclado y muestre por pantalla alguno de los siguientes mensajes:

- **"El numero ingresado es cuadrado"** Si el numero ingresado es positivo y su raíz cuadrada es un numero entero.
- **"El numero ingresado no es cuadrado"** En caso contrario

Por ejemplo:

Input	Resultado
16	El numero ingresado es cuadrado
30	El numero ingresado no es cuadrado
25	El numero ingresado es cuadrado

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

Reiniciar respuesta

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main()
6  {
7      int num, r;
8
9      cin >> num;
10     r = int(sqrt(num));
11     if(num > 0 && pow(r, 2) == num)
12         cout << "El numero ingresado es cuadrado" << endl;
13     else
14         cout << "El numero ingresado no es cuadrado" << endl;
15
16     return 0;
17 }
```

	Input	Expected	Got
✓	16	El numero ingresado es cuadrado	El numero ingresado es



	Input	Expected	<a href="http://lev2.efn.unco.edu/mod/quiz/review.php?a...">http://lev2.efn.unco.edu/mod/quiz/review.php?a...</a>
✓	30	El numero ingresado no es cuadrado	El numero ingresado no
✓	25	El numero ingresado es cuadrado	El numero ingresado es
✓	4	El numero ingresado es cuadrado	El numero ingresado es
✓	5	El numero ingresado no es cuadrado	El numero ingresado no

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Resuelva el siguiente problema, usando estructuras de decisión compuestas y cualquier otro recurso de las unidades anteriores.

Se desea contar con un programa en C++ que le permita al usuario ingresar las coordenadas (positivas y distintas) de un vector tridimensional con la finalidad de escalarlo (dividir cada componente por la mayor). El despliegue por pantalla debe indicar la componente mayor y el vector escalado.

Ejemplo

Ingrese x1: 5.4

Ingrese x2: 3.8

Ingrese x3: 7.3

La coordenada de mayor valor es 3

Las componentes del vector escalado resultan: 0.739726 0.520548 1

**IMPORTANTE:** La primera corrección de su programa es automática, por consiguiente su programa debe imprimir EXACTAMENTE lo que la consigna del ejercicio y los ejemplos de ejecución sugieren en caso contrario los test no pasarán. No olvide considerar espacios en blancos, saltos de línea, minúsculas y mayúsculas.

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

Reiniciar respuesta

```

4  {
5      float x1, x2,x3, max ;
6
7      cin>>x1;
8      do
9      {
10         cin>>x2;
11     }
12     while(x2==x1); //si es falso sale, si es verdadero vuelve
13
14     do
15     {
16         cin>>x3;
17     }
18     while(x3==x1 || x3==x1);
19
20     max=x1;
21     if(x1<x2 && x2>x3)
22     {
23         max=x2;
24         cout<<"La coordenada de mayor valor es " << "2" << endl;
25     }
26     else if(x3>x1 && x3>x2)
27     {
28         max=x3;
29         cout<<"La coordenada de mayor valor es " << "3" << endl;
30     }
31     else
32     {
33         cout<<"La coordenada de mayor valor es " << "1" << endl;
34     }
35
36     cout<<"Las componentes del vector escalado resultan: " << x1/max << " " << x2/max << " " << x3/max << endl;
37     return 0;
38 }
```

	Input	Expected
✓	6.7 9.3 4.1	La coordenada de mayor valor es 2 Las componentes del vector escalado resultan: 0.72043 1
✓	13.9 24.8 79.3	La coordenada de mayor valor es 3 Las componentes del vector escalado resultan: 0.175284 0
✓	3 2 5	La coordenada de mayor valor es 3 Las componentes del vector escalado resultan: 0.6 0.4

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

**Pregunta 6**

Correcta

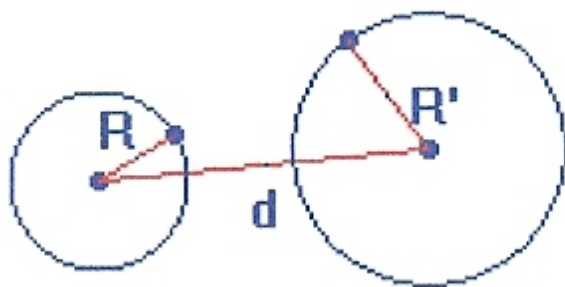
Puntúa 1 sobre 1

Se debe realizar un programa en C++ que determina si dos círculos cualesquiera cumplen las siguientes condiciones:

- No se tocan
- Tienen solapamiento
- El mayor contiene al menor
- Son concéntricos

El programa debe solicitar radio y coordenadas (x,y) del centro del primer círculo y luego del segundo. Respetar este orden de ingreso de los datos.

**Pista:** Se debe trabajar con la distancia entre los centros y los radios. Hacer gráficos en papel de ser necesario.



**Por ejemplo:**

Test	Input	Resultado
// Concéntricos	5.15 25.3 34.8 11.43 25.3 34.8	Son concéntricos

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

Reiniciar respuesta

```

6 using namespace std;
7
8 int main () {
9     float xc1, yc1, r1, xc2, yc2, r2, d ;
10
11     cin>>r1 ;
12     cin>>xc1 ;
13     cin>>yc1 ;
14     cin>>r2 ;
15     cin>>xc2 ;
16     cin>>yc2 ;
17
18     d=sqrt(pow((xc2-xc1),2)+ pow((yc2-yc1),2));
19
20     if(d>r1+r2)
21         cout<<"No se tocan"<<endl;
22     else if(d<=r1+r2&& d> abs(r2-r1) )
23         cout<<"Tienen solapamiento"<<endl;
24     else if((xc1!=xc2|| yc1!=yc2) && (d<abs(r1-r2)&& (d<r1+r2)))

```

	Test	Input	Expected	Got
✓	// Concentricos	5.15 25.3 34.8 11.43 25.3 34.8	Son concentricos	Son concentricos
✓	// No se tocan	3.5 -2.1 2.1 2.75 8.1 7.6	No se tocan	No se tocan
✓	// Solapamiento	4.3 1.5 -2.1 7.1 3.2 1.7	Tienen solapamiento	Tienen solapamiento

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Resuelva el siguiente problema, usando exclusivamente estructuras condicionales múltiples y cualquier otro recurso de las unidades anteriores.

Se desea contar con un programa en C++ que le permita al estudiante ingresar el porcentaje obtenido en un examen para que en la pantalla se despliegue su calificación con un mensaje alusivo con el siguiente criterio:

Si es menor que 45: "Tendra que reforzar sus estudios, su resultado es insuficiente"

Si es mayor o igual a 45 pero menor que 60: "Lo siento, su resultado es insuficiente para aprobar, aunque anduvo muy cerca"

Si es mayor o igual a 60 pero menor que 70: "Alcanzo la suficiencia y su calificacion es Buena"

Si es mayor o igual a 70 pero menor que 85: "Bien logrado!, su calificacion es Muy Buena"

Si es mayor o igual a 85 pero menor que 100: "Felicitaciones!, obtuvo una calificacion distinguida"

Si es igual a 100: "10, Felicitado, sobresaliente!";

### Ejemplo:

Ingrese el porcentaje obtenido entre 0 y 100: 55

Lo siento, su resultado es insuficiente para aprobar, aunque anduvo muy cerca

**IMPORTANTE:** La primera corrección de su programa es automática, por consiguiente su programa debe imprimir EXACTAMENTE lo que la consigna del ejercicio y los ejemplos de ejecución sugieren en caso contrario los test no pasarán. No olvide considerar espacios en blancos, saltos de línea, minúsculas y mayúsculas.

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

Reiniciar respuesta

```

7      // ...
8      cin >> nota;
9
10     if (nota == 100)
11     {
12         cout << "10, Felicitado, sobresaliente!";
13     }
14     else if (nota >= 85)
15     {
16         cout << "Felicitaciones!, obtuvo una calificacion distinguida";
17     }
18     else if (nota >= 70)
19     {
20         cout << "Bien logrado!, su calificacion es Muy Buena";
21     }
22     else if (nota >= 60)
23     {
24         cout << "Alcanzo la suficiencia y su calificacion es Buena";
25     }
26     else if (nota >= 45)
27     {
28         cout << "Lo siento, su resultado es insuficiente para aprobar";
29     }
30     else if (nota < 45 && nota > 0)
31     {
32         cout << "Tendra que reforzar sus estudios, su resultado es i

```

	Input	Expected
✓	10	Tendra que reforzar sus estudios, su resultado es insufici
✓	85	Felicitaciones!, obtuvo una calificacion distinguida
✓	76	Bien logrado!, su calificacion es Muy Buena

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

**Pregunta 8**

Correcta

Puntúa 1 sobre 1

Leer las longitudes de los lados de un triángulo y determinar si se trata de un triángulo isosceles, escaleno o equilatero.

**Por ejemplo:**

Input	Resultado
1 2 2	isosceles

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

```

1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      float l1, l2, l3;
6      cin>>l1;
7      cin>>l2;
8      cin>>l3;
9      if(l1==l2&&l2==l3)
10         cout<<"equilatero"<<endl;
11     else if(l1==l2||l2==l3||l1==l3)
12         cout<<"isosceles"<<endl;
13     else if((l1!=l2&&l1!=l3)&&l2!=l3)
14         cout<<"escaleno"<<endl;
15
16     return 0;
17 }
```

	Input	Expected	Got	
✓	1 2 2	isosceles	isosceles	✓
✓	2.0 2.0 2.0	equilatero	equilatero	✓
✓	2.3 4.5 1.2	escaleno	escaleno	✓
✓	2.1 2.1 6.7	isosceles	isosceles	✓

Todas las pruebas superadas. ✓

**Correcta**





Realice un programa **C++** que solicite al usuario el ingreso de 3 números enteros que deben ser interpretados como valores de horas, minutos y segundos respectivamente. Luego si la hora ingresada es válida debe calcular la hora un segundo después. En caso contrario su programa deberá imprimir "La hora ingresada no es valida" y terminar.

Considere que una hora se considera válida si valen las siguientes condiciones:

- $0 \leq \text{hora} < 24$
- $0 \leq \text{minutos} < 60$
- $0 \leq \text{segundos} < 60$

**Por ejemplo:**

Input	Resultado
15 22 30	15:22:31
24 0 15	La hora ingresada no es valida

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

Reiniciar respuesta

```

7      int hora, min, seg;
8      cin>>hora;
9      cin>>min;
10     cin>>seg;
11     if((0<=hora&&hora<24) && (0<=min&&min<60) && (0<=seg&&seg<60))
12     {
13         seg++;
14         if(seg==60)
15         {
16             seg=0;
17             min++;
18         }
19         if(min==60)
20         {
21             min=0;
22             hora++;
23         }
24         if(hora==24)
25         {
26             hora=0;
27         }
28         cout<<hora<<":"<<min<<":"<<seg<<endl;
29     }
30     else
31     {
32         cout<<"La hora ingresada no es valida"<<endl;
33     }
34     return 0;
35 }
```

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

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Realizar un programa que solicite al usuario que ingrese por teclado un número entero que representa al día del mes y un número entero que representa al mes (1,2,3,4,5,6,7,8,9,10,11,12). Y a continuación muestre un mensaje en función del día y del mes ingresados:

- Si el mes es incorrecto se debe mostrar el mensaje **"Mes incorrecto"**
- Si el mes es correcto pero el día no está en el rango válido para el mes ingresado mostrar el mensaje **"Día incorrecto"**. Para los meses Enero, Marzo, Mayo, Julio, Agosto, Octubre y Diciembre el día debe ser mayor o igual a 1 y menor o igual a 31. Para los meses Abril, Junio, Septiembre 30 y Noviembre el día debe ser mayor o igual a 1 y menor o igual a 30. Para el mes de Febrero tomar como validos los días mayores o iguales a 1 y menores o iguales a 29 para tener en cuenta los años bisiestos.
- Si el mes y el día son correctos mostrar alguno de los siguientes mensajes:
  - **"Verano"** para el periodo que va del 21 de Diciembre al 20 de Marzo
  - **"Otonio"** para el periodo que va del 21 de Marzo al 20 de Junio
  - **"Invierno"** para el periodo que va del 21 de Junio al 20 de Septiembre
  - **"Primavera"** para el periodo que va del 21 de Septiembre al 20 de Diciembre

**Por ejemplo:**

Input	Resultado
15 3	Verano
30 4	Otonio
32 12	Dia incorrecto
2 22	Mes incorrecto
45 13	Mes incorrecto

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

Reiniciar respuesta

```

6      int dia, mes;
7      cin>>dia;
8      cin>>mes;
9
10     if(mes>12||mes<=0)
11     {
12         cout<<"Mes incorrecto"<<endl;
13     }
14     else if(mes==1 && (dia>=1 && dia<=31))
15     {
16         cout<< "Verano"<<endl;
17     }
18     else if(mes==2 && (dia>=1 && dia<=29))
19     {

```



	Input	Expected	Got	
✓	15 3	Verano	Verano	✓
✓	29 2	Verano	Verano	✓
✓	30 4	Otonio	Otonio	✓
✓	32 12	Dia incorrecto	Dia incorrecto	✓
✓	2 22	Mes incorrecto	Mes incorrecto	✓
✓	45 13	Mes incorrecto	Mes incorrecto	✓
✓	45 1	Dia incorrecto	Dia incorrecto	✓
✓	22 8	Invierno	Invierno	✓

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Realice un programa en C++ que determine si un número es perfecto

Un número es perfecto cuando la suma de todos sus divisores es igual al número.

Ejemplo:

Los divisores de 28 son: 1,2,4,7,14

$1+2+4+7+14 = 28$

28 es un número perfecto

Cuando el número es perfecto muestre en pantalla en mensaje "Numero perfecto!". En caso

contrario, el mensaje "El numero no es perfecto".

**Por ejemplo:**

Input	Resultado
28	Numero perfecto!

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

Reiniciar respuesta

```

10  int num, sum, divisor;
11  cin>>num;
12  sum=0;
13  for(int i=1; i<num; i++)
14  {
15      divisor=num%i;
16      if(divisor==0)
17      {
18          sum+= i;
19      }
20  }
21  if(sum==num)
22  {
23      cout<< "Numero perfecto!";
24  }
25  else
26  {
27      cout<< "No es numero perfecto";
28  }
29  return 0;
30  }
```

	Input	Expected	Got	
✓	28	Numero perfecto!	Numero perfecto!	✓
✓	3	No es numero perfecto	No es numero perfecto	✓
✓	496	Numero perfecto!	Numero perfecto!	✓

Todas las pruebas superadas. ✓

Puntos para este envío: 1/1.



Realice un programa en C++ con las siguientes características:

1. Pedir al usuario la cantidad de puntos a valorar (entero), el valor inicial y el valor final del intervalo (flotantes).
2. Dados los valores inicial y final y la cantidad de puntos, obtener el  $\Delta x$  entre los puntos intermedios e imprimirlo.
3. Evaluar para todos los puntos la función:

$$f(x) = \sin(x) + 2 \cos(x)$$

**Notas:**

- No colocar texto en los raw\_input()
- Respetar el formato de texto del ejemplo para que se pueda corregir correctamente

**Por ejemplo:**

Test	Input	Resultado
// Ejemplo	5 -3.25 3.25	Delta x = 1.6250 (-3.250, -1.880) (-1.625, -1.107) (0.000, 2.000) (1.625, 0.890) (3.250, -2.096) FIN

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

Reiniciar respuesta

```

6  #include <cmath>
7  using namespace std;
8  // el prototipo de la funcion
9  float f(float);
10 // declaracion de la funcion
11 float f(float x)
12 {
13     float y;
14     y=sin(x)+2*cos(x);
15     return y;
16 }
17
18 int main () {
19     int cantidad;
20     float vi, vf, deltax, x;
21
22     cin>> cantidad;
23     cin>> vi;
24     cin>> vf;
25
26     deltax=(vf-vi)/(cantidad-1);
27     cout<< "Delta x = "<<fixed<<setprecision(4)<< deltax <<endl;
28
29     for(int i=0; i<cantidad; i++)
30     {
31         x=vi+i*deltax;
32         cout<< "("<<fixed<<setprecision(3)<< x<<","<< f(x)<<")<< " ";
33     }
34     cout<<"FIN"<<endl;

```

	Test	Input	Expected	Got
✓	// Ejemplo	5 -3.25 3.25	Delta x = 1.6250 (-3.250, -1.880) (-1.625, -1.107) (0.000, 2.000) (1.625, 0.890) (3.250, -2.096) FIN	Delta x = 1. (-3.250, -1.8 (-1.625, -1.1 (0.000, 2.000 (1.625, 0.890 (3.250, -2.09
✓	// Caso OK - Puntos par	8 -1.25 1.25	Delta x = 0.3571 (-1.250, -0.318) (-0.893, 0.476) (-0.536, 1.209) (-0.179, 1.791) (0.179, 2.146) (0.536, 2.230) (0.893, 2.033) (1.250, 1.580) FIN	Delta x = 0. (-1.250, -0.3 (-0.893, 0.47 (-0.536, 1.20 (-0.179, 1.79 (0.179, 2.146 (0.536, 2.230 (0.893, 2.033 (1.250, 1.580 FIN
✓	// Caso Ok - Puntos impar	7 -2.3 2.3	Delta x = 0.7667 (-2.300, -2.078) (-1.533, -0.924) (-0.767, 0.747) (0.000, 2.000) (0.767, 2.134) (1.533, 1.074) (2.300, -0.587) FIN	Delta x = 0. (-2.300, -2.0 (-1.533, -0.9 (-0.767, 0.74 (0.000, 2.000 (0.767, 2.134 (1.533, 1.074 (2.300, -0.58
✓	// Solo x positivos	6 3.7 9.2	Delta x = 1.1000 (3.700, -2.226) (4.800, -0.821) (5.900, 1.481) (7.000, 2.165) (8.100, 0.483) (9.200, -1.727) FIN	Delta x = 1. (3.700, -2.22 (4.800, -0.82 (5.900, 1.481 (7.000, 2.165 (8.100, 0.483 (9.200, -1.72 FIN
✓	// Solo x negativas	7 -14.42 -1.2	Delta x = 2.2033 (-14.420, -1.518) (-12.217, 2.222) (-10.013, -1.108) (-7.810, -0.911) (-5.607, 2.186) (-3.403, -1.673) (-1.200, -0.207) FIN	Delta x = 2. (-14.420, -1. (-12.217, 2.2 (-10.013, -1. (-7.810, -0.9 (-5.607, 2.18 (-3.403, -1.6 (-1.200, -0.2 FIN

Todas las pruebas superadas. ✓

<http://lev2.efn.uncor.edu/mod/quiz/review.php?a...>

**Correcta**

Puntos para este envío: 1/1.

**Pregunta 13**

Correcta

Puntúa 1 sobre 1

Dado un entero n ingresado por teclado calcular la cantidad de números múltiplos de 11 que hay en el rango comprendido entre 0 y n. En caso de que el valor de n ingresado sea menor a 1 volver a solicitarlo hasta que sea mayor o igual a 1. (No deben considerar al 0 como múltiplo de 11).

Por ejemplo, si se ingresa el valor 40 el se debe mostrar el mensaje "Existen 3 múltiplos de 11 en el rango (0,40]" ya que los múltiplos de 11 en dicho rango son 11, 22 y 33

**Por ejemplo:**

Input	Resultado
-1 -2 -123 0 1	Existen 0 multiplos de 11 en el rango (0,1]
-4 -6 0 222	Existen 20 multiplos de 11 en el rango (0,222]
500	Existen 45 multiplos de 11 en el rango (0,500]
-12 -32 -443 0 0 0 -12 -3 -5322 321	Existen 29 multiplos de 11 en el rango (0,321]
11	Existen 1 multiplos de 11 en el rango (0,11]

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

Reiniciar respuesta

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int n, cuenta, divisor;
7      do
8          cin>>n;
9      while(n<1);
10     cuenta= 0;
11     for(int i=1; i<=n; i++)
12     {
13         divisor=i%11;
14         if(divisor==0)
15         {
16             cuenta++;
17         }

```

	Input	Expected	Got
✓	-1 -2 -123 0 1	Existen 0 multiplos de 11 en el rango (0,1]	Existen 0 multiplos de 11 en el rango (0,1]
✓	-4 -6 0 222	Existen 20 multiplos de 11 en el rango (0,222]	Existen 20 multiplos de 11 en el rango (0,222]
✓	500	Existen 45 multiplos de 11 en el rango (0,500]	Existen 45 multiplos de 11 en el rango (0,500]
✓	-12 -32 -443 0 0 0 -12 -3 -5322 321	Existen 29 multiplos de 11 en el rango (0,321]	Existen 29 multiplos de 11 en el rango (0,321]

	Input	Expected	<a href="http://lev2.efn.uncor.edu/mod/quiz/review.php?a...">http://lev2.efn.uncor.edu/mod/quiz/review.php?a...</a>
✓	-123 -2 -3 -45 -2 -3 -3 0 0 0 0 0 0 0 0 60012	Existen 5455 multiplos de 11 en el rango (0,60012]	Existen 5455 multiplos de 11 en el rango (0,60012]
✓	11	Existen 1 multiplos de 11 en el rango (0,11]	Existen 1 multiplos de 11 en el rango (0,11]
✓	23	Existen 2 multiplos de 11 en el rango (0,23]	Existen 2 multiplos de 11 en el rango (0,23]
✓	4123	Existen 374 multiplos de 11 en el rango (0,4123]	Existen 374 multiplos de 11 en el rango (0,4123]
✓	-2 -3 -4 -1 -3 -3 67	Existen 6 multiplos de 11 en el rango (0,67]	Existen 6 multiplos de 11 en el rango (0,67]

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1/1.

Realice un programa en C++ que dado un número entero ingresado por teclado encuentre el primer número entero múltiplo de once mayor al número ingresado. Luego mostrar por pantalla el mensaje "El primer número múltiplo de 11 después de x es y" reemplazando x por el número ingresado e y por el número encontrado.

Ejemplo:

Si se ingresa 5 se debe mostrar el mensaje "El primer número múltiplo de 11 después de 5 es 11"

Si se ingresa 57 se debe mostrar el mensaje "El primer número múltiplo de 11 después de 57 es 66"

Si se ingresa 22 se debe mostrar el mensaje "El primer número múltiplo de 11 después de 22 es 33"

Por ejemplo:

Input	Resultado
1	El primer número múltiplo de 11 después de 1 es 11
45	El primer número múltiplo de 11 después de 45 es 55
-605	El primer número múltiplo de 11 después de -605 es -594
124	El primer número múltiplo de 11 después de 124 es 132

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

Reiniciar respuesta

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int n, divisor, i;
7      cin>>n;
8      i=n;
9      do
10     {
11         n++;
12         divisor=n%11;
13     }
14     while(divisor!=0);
15
16     cout<< "El primer número múltiplo de 11 después de "<< i << " es
17     return 0;
18 }
```

	Input	Expected	C
--	-------	----------	---

	Input	Expected	<a href="http://lev2.efn.uncor.edu/mod/quiz/review.php?a...">http://lev2.efn.uncor.edu/mod/quiz/review.php?a...</a>
✓	1	El primer numero multiplo de 11 despues de 1 es 11	E
✓	45	El primer numero multiplo de 11 despues de 45 es 55	E
✓	-605	El primer numero multiplo de 11 despues de -605 es -594	E
✓	124	El primer numero multiplo de 11 despues de 124 es 132	E
✓	11	El primer numero multiplo de 11 despues de 11 es 22	E
✓	88	El primer numero multiplo de 11 despues de 88 es 99	E
✓	9854	El primer numero multiplo de 11 despues de 9854 es 9856	E

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.



Realice un programa en C++ con las siguientes características:

1. Pedir al usuario el valor inicial y el valor final del intervalo y el  $\Delta x$  (flotantes).
2. Evaluar para todos los puntos la función:  

$$f(x) = \cos(e^{\frac{x}{3}})$$
3. Al finalizar, imprimir la cantidad de puntos valuados.

#### Notas:

- No colocar texto en los cin
- Respetar el formato de texto del ejemplo para que se pueda corregir correctamente

#### Por ejemplo:

Test	Input	Resultado
// Ejemplo	-1.55	(-1.550,0.827)
	2.34	(-0.800,0.721)
	.75	(-0.050,0.554)
		(0.700,0.303)
		(1.450,-0.051)
		(2.200,-0.489)
		Se valuaron 6 puntos
		FIN

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

Reiniciar respuesta

```

8  using namespace std;
9  // el prototipo de la funcion
10 float f(float);
11 // declaracion de la funcion
12 float f(float x)
13 {
14     float y;
15     y=cos(exp(x/3));
16     return y;
17 }
18
19 int main () {
20     int cantidad;
21     float vi, vf, deltax, x;
22
23     cin>> vi;
24     cin>> vf;
25     cin>> deltax;
26     cantidad=((vf-vi)/deltax)+1;
27     for(int i=0; i<cantidad; i++)
28     {
29         x=vi+i*deltax;
30         cout<< "("<<fixed<<setprecision(3)<< x<<","<< f(x)<<")"<
31     }
32     cout<< "Se valuaron "<< cantidad<< " puntos" <<endl;
33     cout<<"FIN"<<endl;
34     return 0;
35 }
```

	Test	Input	Expected	Got
✓	// Ejemplo	-1.55 2.34 .75	(-1.550,0.827) (-0.800,0.721) (-0.050,0.554) (0.700,0.303) (1.450,-0.051) (2.200,-0.489) Se valoraron 6 puntos FIN	(-1.550,0.827) (-0.800,0.721) (-0.050,0.554) (0.700,0.303) (1.450,-0.051) (2.200,-0.489) Se valoraron 6 p FIN
✓	// Solo x negativos	-7.13 -6.8 0.125	(-7.130,0.996) (-7.005,0.995) (-6.880,0.995) Se valoraron 3 puntos FIN	(-7.130,0.996) (-7.005,0.995) (-6.880,0.995) Se valoraron 3 p FIN

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

**Pregunta 16**

Correcta

Puntúa 1 sobre 1

Realice un programa en C++ que implemente una calculadora con operaciones de suma, resta, multiplicación y división.

El programa debe solicitar la operación que desea realizar (+, -, \*, /) y a continuación los dos operandos, y mostrar el resultado de la operación por pantalla.

"Mientras" la operación indicada sea válida (es decir se ingrese +, -, \*, o /) el programa debe seguir solicitando un nuevo ingreso. Cuando la operación ingresada no es válida el programa termina mostrando el mensaje "FIN" por pantalla.

**Notas:**

- No colocar texto en los cin
- Respetar el formato de texto del ejemplo para que se pueda corregir correctamente

**Por ejemplo:**

Input	Resultado
+	2.1
2	FIN
0.1	
p	
2	
3	

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

Reiniciar respuesta

```

8
9 ▼ int main () {
10     char operador;
11     float operando1, operando2, resultado;
12     cin>>operador;
13     cin>>operando1;
14     cin>>operando2;
15     while(int(operador)==43 || int(operador)==45 || int(operador)==47 || int(operador)==42)
16 ▼     {
17         if(int(operador)==43)
18             resultado= operando1 + operando2;
19         else if(int(operador)==45)
20             resultado= operando1 - operando2;
21         else if(int(operador)==47)
22             resultado= operando1 / operando2;
23         else if(int(operador)==42)
24             resultado= operando1 * operando2;
25         cout<< resultado<<endl;
26         cin>>operador;
27         cin>>operando1;
28         cin>>operando2;
29     }
30
31     cout<<"FIN"<<endl;
32 }
```

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

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Realice un programa en C++ con las siguientes características:

1. Pedir al usuario valores enteros positivos hasta que ingrese un valor negativo.
2. Ir mostrando por pantalla numero de orden, valor ingresado y si es par o impar.  
(ver formato en el ejemplo)
3. Al finalizar la carga, mostrar la cantidad de pares ingresados.

**Por ejemplo:**

Test	Input	Resultado
// Ejemplo	5	Valor 1 = 5; impar
	10	Valor 2 = 10; par
	3	Valor 3 = 3; impar
	4	Valor 4 = 4; par
	-1	FIN: 2 pares

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

Reiniciar respuesta

```

7  int main()
8  {
9      int posit, pares;
10     int valor=0;
11     cin>>posit;
12     valor++;
13     pares=0;
14     if(posit>=0)
15     {
16         do
17         {
18             if(posit%2==0)
19             {
20                 cout<<"Valor "<< valor<<" = "<< posit<<"<<endl;
21                 pares++;
22             }
23             else
24             {
25                 cout<<"Valor "<< valor<<" = "<< posit<<"<<endl;
26             }
27             cin>>posit;
28             valor++;
29         }
30         while(posit>=0);
31     }
32
33     cout<<"FIN: "<< pares <<" pares"<< endl;
34
35     return 0;
36 }
```

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

	Test	Input	Expected	Got	
✓	// Ejemplo	5 10 3 4 -1	Valor 1 = 5; impar Valor 2 = 10; par Valor 3 = 3; impar Valor 4 = 4; par FIN: 2 pares	Valor 1 = 5; impar Valor 2 = 10; par Valor 3 = 3; impar Valor 4 = 4; par FIN: 2 pares	✓
✓	// Caso OK	61 52 43 39 24 -8	Valor 1 = 61; impar Valor 2 = 52; par Valor 3 = 43; impar Valor 4 = 39; impar Valor 5 = 24; par FIN: 2 pares	Valor 1 = 61; impar Valor 2 = 52; par Valor 3 = 43; impar Valor 4 = 39; impar Valor 5 = 24; par FIN: 2 pares	✓

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Realice un programa en C++ con las siguientes características:

1. Pedir al usuario valores enteros hasta que ingrese el valor 0 (cero).
2. Ir mostrando por pantalla numero de orden, valor ingresado y, de ser necesario, si es múltiplo de 3. (ver formato en el ejemplo)
3. Al finalizar la carga, mostrar la cantidad de múltiplos de 3 ingresados.

**Por ejemplo:**

Test	Input	Resultado
// Ejemplo	6	Valor 1 = 6; multiplo de 3
	10	Valor 2 = 10;
	-6	Valor 3 = -6; multiplo de 3
	-10	Valor 4 = -10;
	0	FIN: 2 multiplos de 3

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

Reiniciar respuesta

```

5  using namespace std;
6
7  int main()
8  {
9      int posit, multiplo;
10     int valor=0;
11     cin>>posit;
12     valor++;
13     multiplo=0;
14     if(posit!=0)
15     {
16         do
17         {
18             if(posit%3==0)
19             {
20                 cout<<"Valor "<< valor<<" = "<< posit<<"<<endl;
21                 multiplo++;
22             }
23             else
24             {
25                 cout<<"Valor "<< valor<<" = "<< posit<<"<<endl;
26             }
27             cin>>posit;
28             valor++;
29         }
30         while(posit!=0);
31     }
32
33     cout<<"FIN: "<< multiplo <<" multiplos de 3"<< endl;
34
35     return 0;
36 }
```

	Test	Input	Expected	Got
✓	// Ejemplo	6 10 -6 -10 0	Valor 1 = 6; multiplo de 3 Valor 2 = 10; Valor 3 = -6; multiplo de 3 Valor 4 = -10; FIN: 2 multiplos de 3	Valor 1 = 6; mult: Valor 2 = 10; Valor 3 = -6; multi Valor 4 = -10; FIN: 2 multiplos (

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1/1.



Un **número perfecto** es un número natural que es igual a la suma de sus divisores propios positivos.

Así, 6 es un número perfecto porque sus divisores propios son 1, 2 y 3; y  $6 = 1 + 2 + 3$ . Los siguientes números perfectos son 28, 496 y 8128.

$$28 = 1 + 2 + 4 + 7 + 14$$

$$496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248$$

$$8128 = 1 + 2 + 4 + 8 + 16 + 32 + 64 + 127 + 254 + 508 + 1016 + 2032 + 4064$$

Realice un programa en C++ que pida que se ingrese por teclado un numero entero, en caso de que el numero ingresado sea no positivo volver a solicitarlo hasta que se ingrese un numero positivo. Luego, si el número ingresado es perfecto imprimir el mensaje "El numero ingresado es perfecto" y en caso contrario imprimir el mensaje "El numero ingresado no es perfecto".

**Por ejemplo:**

Input	Resultado
-1 -2 -1231 0 6	El numero ingresado es perfecto
0 87	El numero ingresado no es perfecto
28	El numero ingresado es perfecto

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

Reiniciar respuesta

```

4  int main()
5  {
6      int num, sum, divisor;
7
8      do
9          cin>>num;
10         while(num<=0);
11         sum=0;
12         for(int i=1; i<num; i++)
13         {
14             divisor=num%i;
15             if(divisor==0)
16             {
17                 sum+= i;
18             }
19         }
20         if(sum==num)
21         {
22             cout<< "El numero ingresado es perfecto";
23         }

```

	Input	Expected	Got
✓	-1 -2 -1231 0 6	El numero ingresado es perfecto	El numero ingresado es
✓	0 87	El numero ingresado no es perfecto	El numero ingresado no
✓	28	El numero ingresado es perfecto	El numero ingresado es
✓	-1 -12 -123 -4 0 0 0 -23 496	El numero ingresado es perfecto	El numero ingresado es
✓	8128	El numero ingresado es perfecto	El numero ingresado es
✓	-12 -12 -12 -12 -4231 -123 -123 -32 -2 0 0 0 0 100	El numero ingresado no es perfecto	El numero ingresado no

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

Realice un programa **C++** que solicite al usuario el ingreso de números flotantes hasta que el usuario ingrese el 0. Sobre los valores ingresados deberá calcular los siguientes ítems:

- Egresos: Es la sumatoria de todos los valores < 0 ingresados.
- Ventas <= 1000: Es la sumatoria de todos los valores >0 y <1000 ingresados
- Ventas > 1000: Es la sumatoria de todos los valores >0 y >1000 ingresados

Estos valores deberá utilizarlos para imprimir el siguiente reporte:

Ingresos: X0

Egresos: X1

Ingresos Netos: X2

Además si las ventas son mayores a 0 deberá imprimir lo siguiente:

Proporcion de ventas <= 1000: X3

Proporcion de ventas > 1000: X4

Dónde X0 es la cantidad de ventas totales, X1 la cantidad de egresos, X3 es ingresos-egresos, X4 es ventas <= 1000 / ventas totales y X5 es ventas > 1000 / ventas totales.

Ejemplo:

```
1000
200
250.35
2500.15
-150
-125.35
0
Ingresos: 3950.500000
Egresos: -275.350000
Ingresos netos: 3675.150000
Proporcion de ventas <= 1000: 0.367131
Proporcion de ventas > 1000: 0.632869
```

**Por ejemplo:**

Input	Resultado
1000 200 250.35 2500.15 -150 -125.35 0	Ingresos: 3950.500000 Egresos: -275.350000 Ingresos netos: 3675.150000 Proporcion de ventas <= 1000: 0.367131 Proporcion de ventas > 1000: 0.632869
0	Ingresos: 0.000000 Egresos: 0.000000 Ingresos netos: 0.000000
-25.5 -25.5 100.25 -49 100.42 0	Ingresos: 200.670000 Egresos: -100.000000 Ingresos netos: 100.670000 Proporcion de ventas <= 1000: 1.000000 Proporcion de ventas > 1000: 0.000000

Input	Resultado
2531.25 2468.75 -15.25 -75.75 0	Ingresos: 5000.000000 Egresos: -91.000000 Ingresos netos: 4909.000000 Proporcion de ventas <= 1000: 0.000000 Proporcion de ventas > 1000: 1.000000
150.32 203.25 300.085 1500.35 1589.23 0	Ingresos: 3743.235000 Egresos: 0.000000 Ingresos netos: 3743.235000 Proporcion de ventas <= 1000: 0.174623 Proporcion de ventas > 1000: 0.825377
-1500.35 -1499.65 0	Ingresos: 0.000000 Egresos: -3000.000000 Ingresos netos: -3000.000000

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

Reiniciar respuesta

```

0  {
7      double num, egresos, ventas1, ventas2, ventas;
8      egresos=0;
9      ventas1=0;
10     ventas2=0;
11
12     do
13     {
14         cin>>num;
15         if(num<0.0)
16         {
17             egresos+=num;
18         }
19         else if(num>0&&num<=1000)
20         {
21             ventas1+=num;
22         }
23         else if(num>1000)
24         {
25             ventas2+=num;
26         }
27     }
28     while(num!=0);
29
30     cout<<"Ingresos: "<<setprecision(6)<<fixed<< ventas1+ ventas2<<endl;
31     cout<<"Egresos: "<<fixed<<setprecision(6)<<egresos<<endl;
32     cout<<"Ingresos Netos: "<<fixed<<setprecision(6)<<((ventas1+ventas2)-egresos)<<endl;
33     ventas= ventas1+ventas2;
34     if(ventas>0)
35     {
36         cout<<"Proporcion de ventas <= 1000: " <<fixed <<setprecision(6)<<ventas1/ventas<<endl;
37         cout<<"Proporcion de ventas > 1000: " <<fixed <<setprecision(6)<<ventas2/ventas<<endl;
38     }
39
40     return 0;
41 }
```

	Input	Expected	Got
✓	1000 200 250.35 2500.15 -150 -125.35 0	Ingresos: 3950.500000 Egresos: -275.350000 Ingresos netos: 3675.150000 Proporcion de ventas <= 1000: 0.367131 Proporcion de ventas > 1000: 0.632869	Ingresos: 3950 Egresos: -275.35 Ingresos Netos: 3675.15 Proporcion de ventas <= 1000: 0.367131 Proporcion de ventas > 1000: 0.632869
✓	0	Ingresos: 0.000000 Egresos: 0.000000 Ingresos netos: 0.000000	Ingresos: 0.000000 Egresos: 0.000000 Ingresos Netos: 0.000000
✓	-25.5 -25.5 100.25 -49 100.42 0	Ingresos: 200.670000 Egresos: -100.000000 Ingresos netos: 100.670000 Proporcion de ventas <= 1000: 1.000000 Proporcion de ventas > 1000: 0.000000	Ingresos: 200.67 Egresos: -100.00 Ingresos Netos: 100.67 Proporcion de ventas <= 1000: 1.000000 Proporcion de ventas > 1000: 0.000000
✓	2531.25 2468.75 -15.25 -75.75 0	Ingresos: 5000.000000 Egresos: -91.000000 Ingresos netos: 4909.000000 Proporcion de ventas <= 1000: 0.000000 Proporcion de ventas > 1000: 1.000000	Ingresos: 5000.00 Egresos: -91.00 Ingresos Netos: 4909.00 Proporcion de ventas <= 1000: 0.000000 Proporcion de ventas > 1000: 1.000000
✓	150.32 203.25 300.085 1500.35 1589.23 0	Ingresos: 3743.235000 Egresos: 0.000000 Ingresos netos: 3743.235000 Proporcion de ventas <= 1000: 0.174623 Proporcion de ventas > 1000: 0.825377	Ingresos: 3743.235 Egresos: 0.000000 Ingresos Netos: 3743.235 Proporcion de ventas <= 1000: 0.174623 Proporcion de ventas > 1000: 0.825377
✓	-1500.35 -1499.65 0	Ingresos: 0.000000 Egresos: -3000.000000 Ingresos netos: -3000.000000	Ingresos: 0.000000 Egresos: -3000.00 Ingresos Netos: -3000.00

Todas las pruebas superadas. ✓

**Correcta**

Puntos para este envío: 1/1.

◀ REC123

Ir a...

Recursos Adicionales Clase 8 ▶