

Área personal ► Mis cursos ► InfoC++ ► Clase 7 ► Guía interactiva 7

Pregunta 1

Correcta

Puntúa 1,00 sobre
1,00

En una determinada sección de un programa un programador escribió lo siguiente, determine cuales de las opciones son equivalentes.

```
const int N = 3;  
double h[N] = {12.23, -54.2, 120.2};
```

Seleccione una o más de una:

- ☐ a.

```
const int N = 5;  
double h[N] = {12.23, -54.2, 120.2};
```
- ☐ b.

```
const int N = 3;  
double h[N] = {12.23, 54.2, 120.2};
```
- ☒ c.

```
const int N = 3;  
double h[N];  
h[0] = 12.23;  
h[1] = -54.2;  
h[2] = 120.2;
```

 ✓
- ☒ d.

```
double h[] = {12.23, -54.2, 120.2};
```

 ✓
- ☒ e.

```
double h[3] = {12.23, -54.2, 120.2};
```

 ✓
- ☐ f.

```
double h[5] = {12.23, -54.2, 120.2};
```
- ☐ g.

```
double h[] = {12.23, -54.2, 120.2, 1.2};
```
- ☒ h.

```
double h[3];  
h[0] = 12.23;  
h[1] = -54.2;  
h[2] = 120.2;
```

 ✓

Respuesta correcta

Ha seleccionado correctamente 4.

Correcta

Puntos para este envío: 1,00/1,00.

A un programador le dieron la siguiente consigna:

Un supermercado aplica descuentos en las compras de sus clientes según estos sean estudiantes o jubilados:

- Los estudiantes tienen descuentos los días Lunes, Miércoles y Viernes del 10%, los jueves del 20% y el resto de los días del 5%
- Los jubilados tienen un 5% de descuentos los Lunes y Martes, 20% los Miércoles y un 8% el resto de los días

Escriba un programa que solicite 2 números enteros y un número flotante. Deberán ser interpretados como *tipo de cliente* (0 para estudiantes, 1 para jubilados, 2 para otro), *día de la semana* (0 es Lunes, 1 es Martes, ..., 6 es Domingo) y *monto de compra*. Al finalizar su programa deberá reportar el monto que debe abonar el cliente como indica el ejemplo de ejecución.

El programador escribió un código elegante usando arreglos, pero no funciona de manera correcta. Corrija la implementación.

Por ejemplo:

Input	Result
0 0 100	Monto a abonar: 90.00
0 1 100	Monto a abonar: 95.00

Respuesta: (penalty regime: 0 %)

```

1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main()
6  {
7      double desc_estudiantes[] = {0.1,0.05,0.1,0.2,0.1,0.05,0.05};
8      double desc_jubilados[] = {0.05,0.05,0.2,0.08,0.08,0.08,0.08,0.08};
9
10     int tipo_cliente, dia;
11     double monto;
12
13     cin>>tipo_cliente>>dia>>monto;
14
15     if (tipo_cliente==0)
16         monto *= 1 - desc_estudiantes[dia];
17     else if (tipo_cliente==1)
18         monto *= 1 - desc_jubilados[dia];

```

Comprobar

	Input	Expected	Got	
✓	0 0 100	Monto a abonar: 90.00	Monto a abonar: 90.00	✓
✓	0 1 100	Monto a abonar: 95.00	Monto a abonar: 95.00	✓
✓	0 1 150.5	Monto a abonar: 142.97	Monto a abonar: 142.97	✓
✓	0 2 100	Monto a abonar: 90.00	Monto a abonar: 90.00	✓
✓	1 2 100	Monto a abonar: 80.00	Monto a abonar: 80.00	✓
✓	0 3 100	Monto a abonar: 80.00	Monto a abonar: 80.00	✓
✓	1 3 100	Monto a abonar: 92.00	Monto a abonar: 92.00	✓
✓	0 4 125.38	Monto a abonar: 112.84	Monto a abonar: 112.84	✓
✓	1 4 125.38	Monto a abonar: 115.35	Monto a abonar: 115.35	✓
✓	0 5 100	Monto a abonar: 95.00	Monto a abonar: 95.00	✓
✓	1 5 100	Monto a abonar: 92.00	Monto a abonar: 92.00	✓
✓	0 6 2048.12	Monto a abonar: 1945.71	Monto a abonar: 1945.71	✓
✓	1 6 1513.26	Monto a abonar: 1392.20	Monto a abonar: 1392.20	✓
✓	0 5 123.33	Monto a abonar: 117.16	Monto a abonar: 117.16	✓

	Input	Expected	http://lev2.efn.uncor.edu/mod/quiz/attempt.php?...	
✓	1 2 29212.1 4	Monto a abonar: 23369.7 1	Monto a abonar: 23369. 71	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 3

Correcta

Puntúa 1,00 sobre 1,00

Escriba un programa en C++ que solicite se ingrese por teclado un arreglo de 5 enteros y luego por pantalla los valores del arreglo en el orden inverso al que fueron ingresados.

Por ejemplo:

Input	Result
1 -1 2 -2 3	3 -2 2 -1 1

Respuesta: (penalty regime: 0 %)

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      const int N=5;
7      int arreglo[N];
8
9      for(int i=0;i<N;i++)
10     {
11         cin>>arreglo[i];
12     }
13     for(int i=N-1;i>=0;i--)
14     {
15         cout<<arreglo[i]<<" ";
16     }
17
18 }
```

Comprobar

	Input	Expected	Got	
✓	1 -1 2 -2 3	3 -2 2 -1 1	3 -2 2 -1 1	✓
✓	54 23 7 -1 -100	-100 -1 7 23 54	-100 -1 7 23 54	✓
✓	703 6531 -123 2 2	2 2 -123 6531 703	2 2 -123 6531 703	✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 4

Correcta

Puntúa 1,00 sobre 1,00

Escriba un programa en C++ que solicite se ingrese por teclado un valor n entero mayor que cero y menor o igual que 15, en el caso de que el valor ingresado sea incorrecto se debe volver a solicitar tantas veces como sea necesario. Luego el programa debe solicitar que se ingresen n valores enteros y guardarlos en un arreglo de tamaño 15. Finalmente el programa debe mostrar por pantalla los valores del arreglo en el orden inverso al que fueron ingresados.

Por ejemplo:

Input	Result
0	6 -12312 21236 643
16	
4	
643	
21236	
-12312	
6	

Respuesta: (penalty regime: 0 %)

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      const int N=15;
7      int arreglo[N];
8      int i, n;
9
10     do
11     {
12         cin >> n ;
13     }
14     while(!(n>0 && n<=15)); //sale cuando es falso
15
16     for(i=0;i<n;i++)
17     {
18         cin>>arreglo[i];
19     }
20 }
```

Comprobar

	Input	Expected	Got	
✓	-1	5 4 3	5 4 3	✓
	0			
	16			
	21			
	100			
	-1			
	0			
	3			
	3			
	4			
	5			

	Input	Expected	Got	
✓	0 16 4 643 21236 -12312 6	6 -12312 21236 643	6 -12312 21236 643	✓
✓	5 703 6531 -123 2 2	2 2 -123 6531 703	2 2 -123 6531 703	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 5

Correcta

Puntúa 1,00 sobre 1,00

A un programador le dieron la siguiente consigna:

Escriba un programa que solicite se ingresen caracteres hasta que se ingrese una letra minúscula o se alcance una cantidad máxima de 10 caracteres. Al finalizar su programa deberá listar los caracteres ingresados tal como se muestra en el ejemplo de ejecución.

El programador dio la implementación que aparece a continuación. Sin embargo, esta no es correcta. Corrija la implementación.

Por ejemplo:

Input	Result
A B a	A B
A B C D E F m	A B C D E F

Respuesta: (penalty regime: 0 %)

```

1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main()
6  {
7      const int N=10;
8      char datos[N];
9
10     int cant = 0;
11     do
12     {
13         cin>>datos[cant];
14         cant++;
15     }while((int(datos[cant-1])>=97 && int(datos[cant-1])<=122)||cant
16
17     for(int i=0; i<cant; i++)
18         if ((int(datos[i])>64 && int(datos[i])<91))
    
```

Comprobar

	Input	Expected	Got	
✓	A B a	A B	A B	✓

	Input	Expected	Got	
✓	A B C D E F m	A B C D E F	A B C D E F	✓
✓	Q W E R T Y U O P Q	Q W E R T Y U O P Q	Q W E R T Y U O P Q	✓
✓	A Z z	A Z	A Z	✓
✓	A Z a	A Z	A Z	✓
✓	A B X Z n	A B X Z	A B X Z	✓
✓	a			✓
✓	A B C D E F G H I j	A B C D E F G H I	A B C D E F G H I	✓

	Input	Expected	Got	http://lev2.efn.uncor.edu/mod/quiz/attempt.php?...
✓	A B C D E F G H a	A B C D E F G H	A B C D E F G H	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 6

Correcta

Puntúa 1,00 sobre 1,00

A un programador le dieron la siguiente consigna:

Escriba un programa que solicite valores enteros al usuario. El programa debe permitir la carga de hasta 15 valores. La adquisición de datos finaliza ya sea por que se alcanzo la máxima cantidad de valores permitidos o el usuario ingreso 0. Al finalizar, el programa debe escribir la lista de datos ingresados o reportar que no se ingresaron datos, tal como se muestra en el ejemplo de ejecución.

El programador dio la implementación que aparece a continuación. Sin embargo, esta no es correcta. Corrija la implementación.

Por ejemplo:

Input	Result
10 20 0	Dato 0: 10 Dato 1: 20
-10 -20 30 0	Dato 0: -10 Dato 1: -20 Dato 2: 30
0	No se ingresaron datos

Respuesta: (penalty regime: 0 %)

```

1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main()
6  {
7      const int N = 15;
8      int indice=0;
9      int datos[N];
10
11      do
12      {
13          cin>>datos[indice];
14          indice++;
15      }while( indice<15 || datos[indice-1] != 0);
16
17      if (datos [0]== 0)
18          cout<< "No se ingresaron datos"<<endl;

```

Comprobar

	Input	Expected	Got	
✓	10 20 0	Dato 0: 10 Dato 1: 20	Dato 0: 10 Dato 1: 20	✓

	Input	Expected	Got	
✓	-10 -20 30 0	Dato 0: -10 Dato 1: -20 Dato 2: 30	Dato 0: -10 Dato 1: -20 Dato 2: 30	✓
✓	0	No se ingresaron datos	No se ingresaron datos	✓
✓	1 15 28 -10101 3294 0	Dato 0: 1 Dato 1: 15 Dato 2: 28 Dato 3: -10101 Dato 4: 3294	Dato 0: 1 Dato 1: 15 Dato 2: 28 Dato 3: -10101 Dato 4: 3294	✓
✓	213 34 66 812 45 120 0	Dato 0: 213 Dato 1: 34 Dato 2: 66 Dato 3: 812 Dato 4: 45 Dato 5: 120	Dato 0: 213 Dato 1: 34 Dato 2: 66 Dato 3: 812 Dato 4: 45 Dato 5: 120	✓
✓	1 2 3 4 5 6 7 8 9 10 11 12 13 14 0	Dato 0: 1 Dato 1: 2 Dato 2: 3 Dato 3: 4 Dato 4: 5 Dato 5: 6 Dato 6: 7 Dato 7: 8 Dato 8: 9 Dato 9: 10 Dato 10: 11 Dato 11: 12 Dato 12: 13 Dato 13: 14	Dato 0: 1 Dato 1: 2 Dato 2: 3 Dato 3: 4 Dato 4: 5 Dato 5: 6 Dato 6: 7 Dato 7: 8 Dato 8: 9 Dato 9: 10 Dato 10: 11 Dato 11: 12 Dato 12: 13 Dato 13: 14	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 7

Correcta

Puntúa 1,00 sobre 1,00

A un programador le dieron la siguiente consigna:

Escriba un programa que solicite 10 caracteres al usuario. Al finalizar debe escribir una lista que contenga las letras vocales mayúsculas, en el orden inverso al ingresado por el usuario.

El programador dio la implementación que aparece a continuación. Sin embargo, esta no es correcta. Corrija la implementación.

Por ejemplo:

Input	Result
A	[I E A]
B	
C	
D	
E	
F	
G	
H	
I	
J	

Respuesta: (penalty regime: 0 %)

```

1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main()
6  {
7      const int N = 10;
8      char datos[N];
9
10     for(int i=0; i<N; i++)
11         cin>>datos[i];
12     cout<<"[ ";
13     for(int i=N; i>=0; i--)
14         if(datos[i]=='A' || datos[i]=='E' || datos[i]=='I' || datos[
15             cout<<datos[i]<<" ";
16     cout<<"]"<<endl;
17     return 0;
18 }
```

Comprobar

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

	Input	Expected	Got	
✓	A B C D E F G H I J	[I E A]	[I E A]	✓
✓	Q A P E R I B O V U	[U O I E A]	[U O I E A]	✓
✓	Q W R T Y P S D F G	[]	[]	✓
✓	P Q K A E K N S C	[E A]	[E A]	✓
✓	L M N P Q R S T Y O	[0]	[0]	✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 8

Correcta

Puntúa 1,00 sobre
1,00

En una determinada sección de un programa un programador escribió lo siguiente.

Asumiendo que A es un arreglo de enteros de longitud N determine cuales de las opciones son correctas:

```
int prod = 1;
for(int i=0; i<N; i++)
{
    if(i%2==0)
    {
        prod *= A[i];
    }
}
```

Seleccione una o más de una:



a. Es equivalente a:

```
int prod = 1;
for(int i=0; i<N; i+=2)
{
    prod *= A[i];
}
```



b. Es equivalente a:

```
int prod = 1;
for(int i=N; i>0; i--)
{
    if(i%2==0)
    {
        prod *= A[i];
    }
}
```



c. Es equivalente a:

```
int prod = 1;
for(int i=N-1; i>=0; i++)
{
    if(i%2==0)
    {
        prod *= A[i];
    }
}
```



d. La sección de código calcula la productoria de los elementos de A que se encuentran en posiciones cuyos índices son pares ✓



e. La sección de código calcula la productoria de los elementos de A que se encuentran en posiciones cuyos índices son impares



f. La sección de código calcula la productoria de los elementos de A que son pares



g. Si N es 1, entonces la productoria dará como resultado 1



h. Si N es 2, entonces la productoria será igual a A[0] ✓

Comprobar

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 9

Correcta

Puntúa 1,00 sobre 1,00

Escriba un programa en C++ que solicite se ingrese por teclado los 15 valores de un arreglo X de tipo double. Luego calcular la sumatoria de los 15 elementos y mostrarla por pantalla.

Por ejemplo:

Input	Result
-13.611	-163.504
-20.469	
-37.472	
-29.469	
-13.257	
10.88	
29.505	
42.19	
-26.423	
10.376	
17.932	
-46.911	
-36.964	
-27.157	
-22.654	

Respuesta: (penalty regime: 0 %)

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main()
6  {
7      const int N=15;
8      double X[N];
9      double suma;
10     for(int i=0;i<N;i++)
11     {
12         cin>> X[i];
13     }
14     suma=0;
15     for(int i=0;i<N;i++)
16     {
17         suma+= X[i];
18     }

```

Comprobar

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

	Input	Expected	Got	
✓	-13.611 -20.469 -37.472 -29.469 -13.257 10.88 29.505 42.19 -26.423 10.376 17.932 -46.911 -36.964 -27.157 -22.654	-163.504	-163.504	✓
✓	533.118 595.464 260.352 230.153 788.689 315.198 656.389 152.137 582.696 724.271 876.762 243.278 -33.035 463.001 61.818	6450.29	6450.29	✓
✓	5239.049 5091.247 5108.956 5205.611 5226.902 5057.665 5228.867 5103.051 5095.308 5218.905 5200.719 5061.425 5273.763 5068.973 5182.723	77363.2	77363.2	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 10

Correcta

Puntúa 1,00 sobre 1,00

Escriba un programa en C++ que solicite se ingrese por teclado los 15 valores de un arreglo X de tipo double. Luego se debe calcular y mostrar por pantalla el valor medio μ y la desviación estándar Σ de los 15 valores, se deben mostrar los dos valores separados por coma. Suponiendo que $n=15$ μ y Σ quedan definidos por:

$$\mu = \frac{1}{n} \sum_i^n x_i$$

$$\Sigma = \sqrt{\frac{1}{n} \sum_i^n (x_i - \mu)^2}$$

Por ejemplo:

Input	Result
-13.611	-10.9003, 25.7996
-20.469	
-37.472	
-29.469	
-13.257	
10.88	
29.505	
42.19	
-26.423	
10.376	
17.932	
-46.911	
-36.964	
-27.157	
-22.654	

Respuesta: (penalty regime: 0 %)

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main()
6  {
7      const int N=15;
8      double X[N];
9      double mu, sigma,sum2,sum; //mu=valor medio, sigma= desviacion e
10
11     for(int i=0; i<N; i++)
12     {
13         cin>> X[i];
14     }
15     for(int i=0; i<N; i++)
16         sum+=X[i];
17     mu= (1./N)* sum;
18     for(int i=0;i<N;i++)

```

Comprobar

	Input	Expected	Got	
✓	-13.611 -20.469 -37.472 -29.469 -13.257 10.88 29.505 42.19 -26.423 10.376 17.932 -46.911 -36.964 -27.157 -22.654	-10.9003, 25.7996	-10.9003, 25.7996	✓
✓	533.118 595.464 260.352 230.153 788.689 315.198 656.389 152.137 582.696 724.271 876.762 243.278 -33.035 463.001 61.818	430.019, 267.389	430.019, 267.389	✓
✓	5239.049 5091.247 5108.956 5205.611 5226.902 5057.665 5228.867 5103.051 5095.308 5218.905 5200.719 5061.425 5273.763 5068.973 5182.723	5157.54, 72.7001	5157.54, 72.7001	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 11

Correcta

Puntúa 1,00 sobre 1,00

Escriba un programa en C++ que solicite se ingrese por teclado los 5 valores flotantes de un arreglo llamado A y los 5 valores de un arreglo de flotantes llamado B. Luego se debe calcular la suma elemento a elemento de A con B. El resultado debe guardarse en un arreglo C y se debe mostrar por pantalla sus elementos separados por un espacio.

$$C = A + B$$

$$c_i = a_i + b_i$$

Por ejemplo:

Input	Result
1.12	7.24 2.46 6.67278 6.01 10.32
-1.21	
2.672	
-2.09	
3.123	
6.12	
3.67	
4.00078	
8.1	
7.2	

Respuesta: (penalty regime: 0 %)

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      const int N=5;
7      float A[N],B[N],C[N];
8
9      for(int i=0;i<5;i++)
10         cin>> A[i];
11
12     for(int i=0;i<5;i++)
13         cin>> B[i];
14
15     for(int i=0;i<5;i++)
16     {
17         C[i]=A[i]+B[i];
18         cout<< C[i]<< " ";

```

Comprobar

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

	Input	Expected	http://lev2-ofn.uncor.edu/mod/quiz/attempt.php?... Got	
✓	1.12 -1.21 2.672 -2.09 3.123 6.12 3.67 4.0007 8 8.1 7.2	7.24 2.46 6.67278 6.01 1 0.32	7.24 2.46 6.67278 6.01 10.323	✓
✓	54.54 23.243 7.1 -1.2 -100.3 336 1235.4 35 243 72.212 3 87.467 -122.7	1289.98 266.243 79.3123 86.267 -223.034	1289.98 266.243 79.312 3 86.267 -223.034	✓
✓	703.12 6531.6 34 -123.3 2.5 2 7.73 2.45 546 -123.2 34 4.76	710.85 6534.08 422.7 -12 0.734 6.76	710.85 6534.08 422.7 - 120.734 6.76	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

Escriba un programa en C++ que solicite se ingrese por teclado los 5 valores enteros de un arreglo llamado A. Luego se debe calcular la combinación lineal de A con el arreglo $B = \{1,2,3,4,5\}$ con los coeficientes $c1 = 3$ y $c2 = 1$. El resultado debe guardarse en un arreglo C y se debe mostrar por pantalla sus elementos separados por un espacio.

$$C = c1 \cdot A + c2 \cdot B$$

Por ejemplo:

Input	Result
1	4 -1 9 -2 14
-1	
2	
-2	
3	

Respuesta: (penalty regime: 0 %)

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      const int N=5;
7      int A[N],B[N]={1,2,3,4,5},C[N];
8      int c1 = 3, c2 = 1;
9
10     for(int i=0;i<5;i++)
11         cin>> A[i];
12
13     for(int i=0;i<5;i++)
14     {
15         C[i]=c1*A[i]+c2*B[i];
16         cout<< C[i]<< " ";
17     }
18     return 0;

```

Comprobar

	Input	Expected	Got	
✓	1 -1 2 -2 3	4 -1 9 -2 14	4 -1 9 -2 14	✓
✓	54 23 7 -1 -100	163 71 24 1 -295	163 71 24 1 -295	✓

	Input	Expected	Got	
✓	703 6531 -123 2 2	2110 19595 -366 10 11	2110 19595 -366 10 11	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

A un programador le solicitaron que escriba un programa que solicite 10 números reales que deberán ser interpretados como dos vectores de \mathbb{R}^5 y que calcule y reporte la distancia que existe entre ambos.

El programador dio la implementación que aparece a continuación. Sin embargo, esta no es correcta. Corrija la implementación.

Asumir que la distancia es la distancia euclideana, definida para dos vectores

$P = (p_1, p_2, \dots, p_n)$ y $Q = (q_1, q_2, \dots, q_n)$ en \mathbb{R}^n como:

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Por ejemplo:

Input	Result
1 2 3 4 5 -1 -2 -3 -4 -5	La distancia entre (1,2,3,4,5) y (-1,-2,-3,-4,-5) es 14.83
1 2 3 4 5 1 2 3 4 5	La distancia entre (1,2,3,4,5) y (1,2,3,4,5) es 0.00

Respuesta: (penalty regime: 0 %)

```

9      double dist = 0;
10
11     //ingreso v1
12     for(int i=0; i<5; i++)
13         cin>>v1[i];
14     //ingreso v2
15     for(int i=0; i<5; i++)
16         cin>>v2[i];
17     //calculo distancia
18     for(int i=0; i<5; i++)
19         dist += pow(v1[i]-v2[i],2);
20     //Imprimo mensaje de resultado
21     cout<<"La distancia entre (";
22     for(int i=0; i<4; i++)
23         cout<<v1[i]<<" ";
24
25     cout<<v1[4]<<" ) y (" ;
26

```

Comprobar

	Input	Expected	Got	
✓	1 2 3 4 5 -1 -2 -3 -4 -5	La distancia entre (1,2,3,4,5) y (-1,-2,-3,-4,-5) es 14.83	La distancia entre (1,2,3,4,5) y (-1,-2,-3,-4,-5) es 14.83	✓
✓	1 2 3 4 5 1 2 3 4 5	La distancia entre (1,2,3,4,5) y (1,2,3,4,5) es 0.00	La distancia entre (1,2,3,4,5) y (1,2,3,4,5) es 0.00	✓
✓	2.5 3.4 2.8 12.6 5 23.5 -1.5 -100 2.2 25.6 24.3 11.2 4	La distancia entre (2.5,3.4,2.8,12.6,5,23.5,-1.5,-100,2.2,25.6,24.3,11.2,4) y (-1.5,-100,2.2,25.6,24.3,11.2,4) es 1006.01	La distancia entre (2.5,3.4,2.8,12.6,5,23.5,-1.5,-100,2.2,25.6,24.3,11.2,4) y (-1.5,-100,2.2,25.6,24.3,11.2,4) es 1006.01	✓
✓	2 4 6 8 10 1 1 1 1 1 1	La distancia entre (2,4,6,8,10) y (1,1,1,1,1) es 12.85	La distancia entre (2,4,6,8,10) y (1,1,1,1,1) es 12.85	✓

			http://lev2.cfn.uncor.edu/mod/quiz/attempt.php?...	
	Input	Expected	Got	
✓	2.5 4.3 2.8 100. 2 105. 34 -1.4 -234 -3.4 5 -5 -8.7 7	La distancia entre (2.5,4.3,2.8,100.2,105.34) y (-1.4,-234,-3.45,-5,-8.77) es 284.48	La distancia entre (2.5,4.3,2.8,100.2,105.34) y (-1.4,-234,-3.45,-5,-8.77) es 284.48	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 14

Correcta

Puntúa 1,00 sobre 1,00

Escriba un programa en C++ que solicite se ingrese por teclado los 5 valores enteros de un arreglo llamado A y los 5 valores enteros de un arreglo llamado B. Luego se debe calcular y mostrar por pantalla la cantidad de elementos iguales (mismo valor y misma posición).

Por ejemplo:

Input	Result
1	1
2	
3	
4	
5	
2	
3	
4	
4	
6	

Respuesta: (penalty regime: 0 %)

```

11     for(int i=0;i<5;i++)
12     {
13         cin>> A[i];
14     }
15     for(int i= 0;i<5;i++)
16     {
17         cin>> B[i];
18         if(A[i]==B[i])
19         {
20             cant+= 1;
21         }
22         else
23         {
24             cant+=0;
25         }
26     }
27     cout<<cant<<endl;
28     return 0;

```

Comprobar

	Input	Expected	Got	
✓	1	1	1	✓
	2			
	3			
	4			
	5			
	2			
	3			
	4			
	4			
	6			

	Input	Expected	Got	
✓	100 2 5 7 1 -12 2 5 54 12	2	2	✓
✓	-1 2 -3 4 -5 1 -2 -3 4 -5	3	3	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

En una determinada sección de un programa un programador escribió lo siguiente.

Asumiendo que A es un arreglo de enteros de longitud N determine cuales de las opciones son correctas:

```
bool hay_negativos = false;
for(int i=0; i<N; i++)
{
    if(A[i] < 0)
    {
        hay_negativos = true;
    }
}
if(hay_negativos)
    cout << "Hay negativos"
```

Seleccione una o más de una:

☐

a. Es equivalente a:

```
bool hay_negativos = false;
for(int i=0; i<N; i++)
{
    if( i < 0)
    {
        hay_negativos = true;
    }
}
if(hay_negativos)
    cout << "Hay negativos"
```

☒

b. Es equivalente a:

```
bool hay_negativos = false;
for(int i=0; i<N; i++)
{
    if(A[i] < 0)
    {
        hay_negativos = true;
        break;
    }
}
if(hay_negativos)
    cout << "Hay negativos"
```

☐

c. Es equivalente a:

```
bool hay_negativos = false;
for(int i=0; i<N; i++)
{
    if(A[i] <= 0)
    {
        hay_negativos = true;
        break;
    }
}
if(hay_negativos)
    cout << "Hay negativos"
```

☒

d. Es equivalente a:


```
int cant = 0;
for(int i=0; i<N; i++)
{
    if(A[i] < 0)
    {
        cant++;
    }
}
if(cant>0)
    cout << "Hay negativos"
```

☐

e. Es equivalente a:

```
int cant=0;
for(int i=0; i<N; i++)
{
    if(A[i] >= 0)
    {
        cant++;
    }
}
if(cant==N)
    cout << "Hay negativos"
```

☒

f. La sección encuentra si existen números negativos en el arreglo A. ✓

☐

g. Si N es 0 entonces se mostrará el mensaje "Hay negativos"

☒

h. Es suficiente que exista un número negativo para que se muestre el mensaje.



Comprobar

Respuesta correcta

Ha seleccionado correctamente 4.

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 16

Correcta

Puntúa 1,00 sobre 1,00

Escriba un programa en C++ que solicite se ingrese por teclado los 15 valores de un arreglo X de tipo double. Luego se debe encontrar y mostrar el mínimo valor del arreglo.

Por ejemplo:

Input	Result
-13.611	-46.911
-20.469	
-37.472	
-29.469	
-13.257	
10.88	
29.505	
42.19	
-26.423	
10.376	
17.932	
-46.911	
-36.964	
-27.157	
-22.654	

Respuesta: (penalty regime: 0 %)

```

5  {
6      const int N=15;
7      double X[N];
8      double min;
9
10     cin>>X[0];
11     min=X[0];
12     for(int i=1; i<15; i++)
13     {
14         cin>> X[i];
15         if(X[i]<=min)
16         {
17             min= X[i];
18         }
19     }
20     cout<<min<<endl;
21     return 0;
22 }
```

Comprobar

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

	Input	Expected	Got	
✓	-13.611 -20.469 -37.472 -29.469 -13.257 10.88 29.505 42.19 -26.423 10.376 17.932 -46.911 -36.964 -27.157 -22.654	-46.911	-46.911	✓
✓	533.118 595.464 260.352 230.153 788.689 315.198 656.389 152.137 582.696 724.271 876.762 243.278 -33.035 463.001 61.818	-33.035	-33.035	✓
✓	5239.049 5091.247 5108.956 5205.611 5226.902 5057.665 5228.867 5103.051 5095.308 5218.905 5200.719 5061.425 5273.763 5068.973 5182.723	5057.66	5057.66	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 17

Correcta

Puntúa 1,00 sobre 1,00

Escriba un programa en C++ que solicite se ingrese por teclado los 15 valores de un arreglo X de tipo double. Luego se debe encontrar y mostrar el máximo valor del arreglo.

Por ejemplo:

Input	Result
-13.611	42.19
-20.469	
-37.472	
-29.469	
-13.257	
10.88	
29.505	
42.19	
-26.423	
10.376	
17.932	
-46.911	
-36.964	
-27.157	
-22.654	

Respuesta: (penalty regime: 0 %)

```

5 {
6     const int N=15;
7     double X[N];
8     double max;
9
10    cin>>X[0];
11    max=X[0];
12    for(int i=1; i<15; i++)
13    {
14        cin>> X[i];
15        if(X[i]>=max)
16        {
17            max= X[i];
18        }
19    }
20    cout<<max<<endl;
21    return 0;
22 }
```

Comprobar

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

	Input	Expected	Got	
✓	-13.611 -20.469 -37.472 -29.469 -13.257 10.88 29.505 42.19 -26.423 10.376 17.932 -46.911 -36.964 -27.157 -22.654	42.19	42.19	✓
✓	533.118 595.464 260.352 230.153 788.689 315.198 656.389 152.137 582.696 724.271 876.762 243.278 -33.035 463.001 61.818	876.762	876.762	✓
✓	5239.049 5091.247 5108.956 5205.611 5226.902 5057.665 5228.867 5103.051 5095.308 5218.905 5200.719 5061.425 5273.763 5068.973 5182.723	5273.76	5273.76	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 18

Correcta

Puntúa 1,00 sobre 1,00

Escriba un programa en C++ que solicite se ingrese por teclado los 15 valores de un arreglo X de tipo double. Luego se debe normalizar cada valor para que pertenezca al rango (0,1).

La normalización de cada valor $X[i]$ se puede calcular como $(X[i] - \min)/(max - \min)$.

Por ejemplo:

Input	Result
-13.6	0.373733
11	0.296764
-20.4	0.105936
69	0.195755
-37.4	0.377706
72	0.648601
-29.4	0.857
69	633
-13.2	1
57	0.229941
10.88	0.642945
29.50	0.727747
5	0
42.19	0.111637
-26.4	0.221703
23	0.2722
10.37	
6	
17.93	
2	
-46.9	
11	
-36.9	
64	
-27.1	
57	
-22.6	
54	

Respuesta: (penalty regime: 0 %)

```

13 min=X[0];
14 for(int i=1; i<15; i++)
15 {
16     cin>> X[i];
17     if(X[i]>=max)
18     {
19         max= X[i];
20     }
21
22     if(X[i]<=min)
23     {
24         min= X[i];
25     }
26 }
27
28 for(int i=0; i<15; i++)
29 {

```

Comprobar

	Input	Expected	Got	
✓	-13. 611 -20. 469 -37. 472 -29. 469 -13. 257 10.8 8 29.5 05 42.1 9 -26. 423 10.3 76 17.9 32 -46. 911 -36. 964 -27. 157 -22. 654	0.373733 0.296764 0.105936 0.195755 0.377706 0.648601 0.857633 1 0.229941 0.64294 5 0.727747 0 0.111637 0.221 703 0.272242	0.373733 0.296764 0.105 936 0.195755 0.377706 0 .648601 0.857633 1 0.22 9941 0.642945 0.727747 0 0.111637 0.221703 0.2 72242	✓

			http://lev2.cfm.uncor.edu/mod/quiz/attempt.php?...		
	Input	Expected	Got		
✓	533. 118 595. 464 260. 352 230. 153 788. 689 315. 198 656. 389 152. 137 582. 696 724. 271 876. 762 243. 278 -33. 035 463. 001 61.8 18	0.622285 0.690812 0.322475 0.289282 0.903195 0.382759 0.757778 0.203531 0.676778 0.83239 1 0.303708 0 0.5452 16 0.104257	0.622285 0.690812 0.322 475 0.289282 0.903195 0 .382759 0.757778 0.2035 31 0.676778 0.83239 1 0 .303708 0 0.545216 0.10 4257		✓

	Input	Expected	Got	
✓	5239 .049 5091 .247 5108 .956 5205 .611 5226 .902 5057 .665 5228 .867 5103 .051 5095 .308 5218 .905 5200 .719 5061 .425 5273 .763 5068 .973 5182 .723	0.83936 0.155402 0.237351 0 .684625 0.783149 0 0.792242 0.210025 0.174194 0.746143 0.661987 0.0173995 1 0.0523 281 0.57871	0.83936 0.155402 0.2373 51 0.684625 0.783149 0 0.792242 0.210025 0.174 194 0.746143 0.661987 0 .0173995 1 0.0523281 0. 57871	✓

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

A un programador le dieron la siguiente consigna:

Escriba un programa que solicite 10 números enteros en el rango [1,10]. Al finalizar debe escribir:

- "La colección tiene una única moda", si la colección ingresada tiene un único valor que se repite mas veces que el resto de los valores ingresados
- "La colección es polimodal", si la colección tiene 2 o mas valores que se repiten el número máximo de veces (considerando únicamente los valores ingresados)
- "La colección es amodal" , si todos los valores ingresados tienen la misma frecuencia

El programador dio la implementación que aparece a continuación. El esquema de la solución propuesta es correcto, pero contiene algunos errores. Corrija la implementación

Por ejemplo:

Input	Result
1 2 3 1 2 3 1 1 1 1 2	La coleccion es modal
5 5 5 4 4 4 2 1 9 8	La coleccion es polimodal
4 4 4 4 4 9 9 9 9 9	La coleccion es amodal

Respuesta: (penalty regime: 0 %)

```

8  ▾  /*
9      En coleccion[i] voy a contar
10     la cantidad de veces que el usuario

```

Comprobar

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

	Input	Expected	Got	
✓	1 2 3 4 5 6 7 8 9 10	La coleccion es amodal	La coleccion es amodal	✓
✓	1 2 3 1 2 3 1 1 1 2	La coleccion es modal	La coleccion es modal	✓
✓	5 5 5 4 4 4 2 1 9 8	La coleccion es polimodal	La coleccion es polimodal	✓
✓	2 4 6 8 10 1 1 1 1 1	La coleccion es modal	La coleccion es modal	✓

	Input	Expected	Got	
✓	4 4 4 4 4 9 9 9 9 9	La coleccion es amodal	La coleccion es amodal	✓
✓	6 2 5 5 10 8 5 10 3 6	La coleccion es modal	La coleccion es modal	✓
✓	10 5 10 9 1 9 8 6 5 2	La coleccion es polimodal	La coleccion es polimodal	✓
✓	10 5 5 7 8 4 5 7 2 5	La coleccion es modal	La coleccion es modal	✓

			http://lev2.efn.uncor.edu/mod/quiz/attempt.php?...	
	Input	Expected	Got	
✓	1	La coleccion es amodal	La coleccion es amodal	✓
	1			
	2			
	2			
	3			
	3			
	4			
	4			
	5			
	5			

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 20

Correcta

Puntúa 1,00 sobre 1,00

Escriba un programa en C++ que solicite se ingresen por teclado los 10 valores enteros de un arreglo. Se debe garantizar que el arreglo no tenga valores repetidos, por lo tanto si se ingresa un valor que ya fue ingresado previamente se debe solicitar nuevamente hasta que se ingrese un valor nuevo.

Por ejemplo:

Input	Result
0	0 16 7 23 5 87 2 4 643 21236
16	
16	
16	
16	
16	
16	
16	
16	
7	
23	
5	
87	
2	
4	
643	
2	
2	
2	
2	
21236	

Respuesta: (penalty regime: 0 %)

```

8   int i, j;
9   i=0;
10  while(i<10)
11  {
12      cin>>arreglo[i];
13      j=0;
14      for(int n=0;n<i;n++)
15      {
16          if(arreglo[i]==arreglo[n])
17              j++;
18      }
19      if (j==0)
20          i++;
21  }
22  for(int k=0;k<10;k++)
23      cout<<arreglo[k]<<" ";
24  return 0;
25 }
```

Comprobar

	Input	Expected	Got	
✓	-1 0 16 21 100 -1 0 3 3 4 5 5 64 1	-1 0 16 21 100 3 4 5 64 1	-1 0 16 21 100 3 4 5 64 1	✓
✓	0 16 16 16 16 16 16 7 23 5 87 2 4 643 2 2 2 2 2 2123 6	0 16 7 23 5 87 2 4 643 2 1236	0 16 7 23 5 87 2 4 643 2 1236	✓

	Input	Expected	Got	
✓	1	1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 10	✓
	1			
	1			
	2			
	2			
	2			
	2			
	2			
	2			
	3			
	3			
	3			
	3			
	3			
	3			
	3			
	4			
	4			
	5			
	6			
	6			
	6			
	6			
	6			
	7			
	7			
	7			
	8			
	8			
	9			
	9			
	9			
	10			

Todas las pruebas superadas. ✓

Correcta

Puntos para este envío: 1,00/1,00.