

Área personal ► Mis cursos ► InfoC++ ► Clase 9 ► Guía interactiva 9

Comenzado el	viernes, 12 de octubre de 2018, 09:12
Estado	Finalizado
Finalizado en	sábado, 20 de octubre de 2018, 20:54
Tiempo empleado	8 días 11 horas
Puntos	20,00/20,00
Calificación	10,00 de 10,00 (100%)

Pregunta 1

Correcta

Puntúa 1,00 sobre 1,00

Dado el siguiente programa de C++, seleccione las opciones que son correctas.

```

1
2 #include <iostream>
3 using namespace std;
4
5 void bienvenida();
6
7 int main()
8 {
9     bienvenida();
10    bienvenida();
11    bienvenida();
12    cout << endl;
13    bienvenida();
14 }
15
16 void bienvenida()
17 {
18     cout << "Hola mundo!";
19 }
20
    
```

Seleccione una o más de una:

☐

a. Sale por la pantalla:

```

Hola mundo!
Hola mundo!
Hola mundo!
Hola mundo!
    
```

☒

b. La sentencia en la línea 13 corresponde a una llamada a la función bienvenida


☐

c. La sentencia en la línea 16 es la declaración de la función (prototipo)

☐

d. La sentencia en la línea 16 es la definición de la función bienvenida

☒

e. Sale por pantalla:

```

Hola mundo!Hola mundo!Hola mundo!
Hola mundo!
    
```


☐

f. Sale por pantalla:

```

Hola mundo!Hola mundo!Hola mundo!Hola mundo!
    
```

☒

g. La sentencia en la línea 5 es la declaración de la función (prototipo) ✓

☐

h. La función es llamada desde el programa principal 5 veces

☒

i. Las sentencias entre la línea 16 y 19 corresponden a la definición de la función bienvenida ✓

Respuesta correcta

Las respuestas correctas son: La sentencia en la línea 5 es la declaración de la función (prototipo), Las sentencias entre la línea 16 y 19 corresponden a la definición de la función bienvenida, La sentencia en la línea 13 corresponde a una llamada a la función bienvenida, Sale por pantalla:

```
Hola mundo!Hola mundo!Hola mundo!  
Hola mundo!
```

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

Correcta

Puntos para este envío: 1,00/1,00.

Correcta

Puntúa 1,00 sobre
1,00

Escribir en C++ la definición de una función llamada `vspace` de tipo `void` que no acepta ningún parámetro y que cada vez que se la llama debe dejar una línea en blanco a continuación de lo último que se haya sacado por pantalla. En la respuesta se debe completar únicamente la declaración de la función.

Por ejemplo:

Test	Resultado
<code>cout<<"Hola!";</code>	Hola!
<code>vspace();</code>	
<code>cout<<"Chau";</code>	Chau

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```
1 void vspace()  
2 {  
3     cout<<endl; //endl es como un enter  
4     cout<<endl;  
5 }
```

	Test	Expected	Got	
✓	cout<<"Hola!"; vspace(); cout<<"Chau";	Hola! Chau	Hola! Chau	✓
✓	cout<<"Uno"; vspace(); cout<<"Dos"; vspace(); cout<<"Tres"; vspace(); vspace(); cout<<"Cuatro";	Uno Dos Tres Cuatro	Uno Dos Tres Cuatro	✓

Todas las pruebas superadas. ✓

Question author's solution:

```
void vspace()
{
    cout<<"\n\n";
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 3

Correcta

Puntúa 1,00 sobre 1,00

Escribir en C++ la definición de una función llamada suma de tipo void y que no acepta ningún parámetro. La función debe solicitar que se ingrese por teclado dos valores flotantes y luego muestre por pantalla el resultado de sumarlos.

Por ejemplo:

Test	Input	Resultado
suma();	14.23 7.1	21.33

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```

1 void suma()
2 {
3     float a, b;
4     cin>>a;
5     cin>>b;
6     cout<<a+b;
7 }
```

	Test	Input	Expected	Got	
✓	suma();	14.23 7.1	21.33	21.33	✓
✓	suma(); cout << endl; suma(); cout << endl; suma();	5.12 3.12 4.64 23.4 8.2 1.23	8.24 28.04 9.43	8.24 28.04 9.43	✓

Todas las pruebas superadas. ✓

Question author's solution:

```
void suma()  
{  
    float a,b;  
    cin >> a;  
    cin >> b;  
    cout << a+b;  
  
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 4

Correcta

Puntúa 1,00 sobre 1,00

Defina una función en C++ llamada asteriscos de tipo void que recibe como argumento un número entero (num). La función debe mostrar por pantalla num asteriscos uno al lado del otro.

Por ejemplo:

Test	Resultado
asteriscos(5)	*****

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```

1 void asteriscos(int num)
2 {
3     for(int i=0; i<num; i++)
4     {
5         cout<<"*";
6     }
7
8 }
```

	Test	Expected	Got	
✓	asteriscos(5)	*****	*****	✓
✓	asteriscos(15)	*****	*****	✓
✓	asteriscos(7)	*****	*****	✓

Todas las pruebas superadas. ✓

Question author's solution:

```

void asteriscos(int num)
{
    for(int i=0; i<num; i++)
    {
        cout << '*';
    }
}
```


Puntos para este envío: 1,00/1,00.

Pregunta 5

Correcta

Puntúa 1,00 sobre 1,00

Defina una función en C++ llamada valorInt de tipo void que acepta un número entero y muestre por pantalla alguno de los siguientes mensajes dependiendo del valor pasado cómo argumento.

- "Uno" si el valor es 1
- "Dos" si el valor es 2
- "Tres" si el valor es 3
- "Cuatro" si el valor es 4
- "Cinco" si el valor es 5
- "Seis" si el valor es 6
- "Siete" si el valor es 7
- "Ocho" si el valor es 8
- "Nueve" si el valor es 9
- "Fuera de rango" en cualquier otro caso

Por ejemplo:

Test	Resultado
valorInt(1);	Uno
cout << endl;	Dos
valorInt(2);	Tres
cout << endl;	Cuatro
valorInt(3);	Cinco
cout << endl;	Seis
valorInt(4);	Siete
cout << endl;	Ocho
valorInt(5);	Nueve
cout << endl;	
valorInt(6);	
cout << endl;	
valorInt(7);	
cout << endl;	
valorInt(8);	
cout << endl;	
valorInt(9);	

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```

2 ▼ {
3     switch(num)
4 ▼ {
5         case 1:
6             cout<<"Uno";
7             break;
8         case 2:
9             cout<<"Dos";

```

	Test	Expected	Got	
✓	<pre> valorInt(1); cout << endl; valorInt(2); cout << endl; valorInt(3); cout << endl; valorInt(4); cout << endl; valorInt(5); cout << endl; valorInt(6); cout << endl; valorInt(7); cout << endl; valorInt(8); cout << endl; valorInt(9); </pre>	<pre> Uno Dos Tres Cuatro Cinco Seis Siete Ocho Nueve </pre>	<pre> Uno Dos Tres Cuatro Cinco Seis Siete Ocho Nueve </pre>	✓
✓	<pre> valorInt(8); cout << endl; valorInt(9); cout << endl; valorInt(11); </pre>	<pre> Ocho Nueve Fuera de rango </pre>	<pre> Ocho Nueve Fuera de rango </pre>	✓

	Test	Expected	Got	
✓	valorInt(-1); cout << endl; valorInt(0); cout << endl; valorInt(11);	Fuera de rango Fuera de rango Fuera de rango	Fuera de rango Fuera de rango Fuera de rango	✓
✓	valorInt(-1); cout << endl; valorInt(1120); cout << endl; valorInt(5);	Fuera de rango Fuera de rango Cinco	Fuera de rango Fuera de rango Cinco	✓

Todas las pruebas superadas. ✓

Question author's solution:

```
void valorInt(int num)
{
    switch(num)
    {
        case 1:
            cout << "Uno";
            break;
        case 2:
            cout << "Dos";
            break;
        case 3:
            cout << "Tres";
            break;
        case 4:
            cout << "Cuatro";
            break;
        case 5:
            cout << "Cinco";
            break;
        case 6:
            cout << "Seis";
            break;
        case 7:
            cout << "Siete";
            break;
        case 8:
            cout << "Ocho";
            break;
        case 9:
            cout << "Nueve";
            break;
        default:
            cout << "Fuera de rango";
    }
}
```

Puntos para este envío: 1,00/1,00.

Pregunta 6

Correcta

Puntúa 1,00 sobre 1,00

Defina una función en C++ llamada entdec de tipo void que recibe como argumento un número double (val). La función debe mostrar por pantalla el siguiente mensaje:

"Parte entera: x, parte decimal: y" reemplazando x e y por la parte entera y la parte decimal de val.

Por ejemplo:

Test	Resultado
entdec(15.123)	Parte entera: 15, parte decimal: 0.123

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```

1 void entdec(double val)
2 {
3     int entval;
4     float decval;
5     entval=int(val);
6     decval=val-entval ;
7     cout<<"Parte entera: "<< entval <<" , parte decimal: "<< decval<<endl;
8 }
```

	Test	Expected	Got
✓	entdec(15.123)	Parte entera: 15, parte decimal: 0.123	Parte en
✓	entdec(15)	Parte entera: 15, parte decimal: 0	Parte en
✓	entdec(1624.4)	Parte entera: 1624, parte decimal: 0.4	Parte en

Todas las pruebas superadas. ✓

Question author's solution:

```

void entdec(double val)
{
    cout << "Parte entera: "<<int(val)<<" , parte decimal: "<<val-
int(val);
}
```

Correcta

Pregunta 7

Correcta

Puntúa 1,00 sobre 1,00

Dado el siguiente programa de C++ elija las opciones correctas:

```

1  #include <iostream>
2  using namespace std;
3
4  char opciones();
5
6  int main()
7  {
8      const double PI = 3.14159265359;
9      char op;
10     float r;
11
12     cout << "Ingrese el valor del radio: ";
13     cin >> r;
14     while(r<0)
15     {
16         cout << "EL radio debe ser mayor a cero";
17         cout << "Ingrese el valor del radio: ";
18         cin >> r;
19     }
20     while(1)
21     {
22         op = opciones();
23
24         if(op == 's')
25         {
26             cout << "La superficie de la circunferencia es: " << PI * r * r;
27         }
28         else if(op == 'p')
29         {
30             cout << "La perimetro de la circunferencia es: " << 2 * PI * r;
31         }
32         else if(op == 'q')
33         {
34             cout << "Adios";
35             break;
36         }
37     }
38 }
39
40 char opciones()
41 {
42     char opc = 't';
43
44     cout << endl << "Elija una de las siguientes opciones:" << endl;
45     while(opc!='s' && opc!='p' && opc!='q')
46     {
47         cout << "p ) para mostrar el perimetro"<<endl;
48         cout << "s ) para mostrar la superficie"<<endl;
49         cout << "q ) para salir"<<endl;
50         cin >> opc;
51     }
52     return opc;
53 }
54
55

```

Seleccione una o más de una:

- ☒ a. La función opciones únicamente devolverá uno de los siguientes tres valores: 's', 'p' o 'q' ✓
- ☒ b. La función opciones devuelve un valor de tipo char ✓
- ☒ c. El programa finalizará unicamente cuando se ingrese la opción 'q' ✓
- ☐ d. La función opciones no devuelve ningún valor
- ☒ e. Es posible mostrar en una única ejecución el valor de la superficie y del perímetro de la circunferencia ✓
- ☐ f. En el cuerpo de la función opciones se solicita que se ingrese por teclado un valor de tipo char y se valida que sea diferente a 's', 'p' o 'q'
- ☒ g. La sentencia de la línea 5 corresponde con la declaración de la función

- ☐ h. Si se desea conocer el valor de la superficie y del perímetro de una circunferencia se debe correr 2 veces el programa e ingresar dos veces el mismo valor para r.
- ☐ i. El programa solicitará el valor del radio tantas veces como opciones distintas de 'q' se ingresen
- ☐ j. La sentencia de la línea 42 corresponde con la declaración de la función opciones (prototipo)

Respuesta correcta

Las respuestas correctas son: La sentencia de la línea 5 corresponde con la declaración de la función opciones (prototipo), La función opciones devuelve un valor de tipo char, La función opciones únicamente devolverá uno de los siguientes tres valores: 's', 'p' o 'q', El programa finalizará únicamente cuando se ingrese la opción 'q', Es posible mostrar en una única ejecución el valor de la superficie y del perímetro de la circunferencia

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 8

Correcta

Puntúa 1,00 sobre 1,00

Defina una función en C++ de tipo int llamada enteroValidado que no acepte ningún argumento. La función debe solicitar que se ingrese por teclado un valor entero, valide que esté en el rango de 0 a 100 (ambos inclusive) y luego lo devuelva como valor de retorno. Si se ingresan valores fuera del rango se debe volver a solicitar el ingreso nuevamente, tantas veces como sea necesario.

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```
1 | int enteroValidado()
2 | {
3 |     int a;
4 |
5 |     do
6 |     {
7 |         cin>>a;
8 |         if(a>0 && a<100)
9 |         {
10 |             cout<<a;
11 |         }
12 |     }
13 |     while(a<0 || a>100);
14 |     return a;
15 |
16 | }
```

	Test	Input	Expected	Got	
✓	int x; x = enteroValidado(); cout << endl << x;	-1 -1 101 101 12324 -123123 4312 -6521 65	65	65 65	✓

	Test	Input	Expected	Got	
✓	<pre>int x; x = enteroValidado(); cout << endl << x;</pre>	<pre>-1 -1 -1 -123132 -1243 101 101 123 12465 1237 2312 0</pre>	0	0	✓
✓	<pre>int x; x = enteroValidado(); cout << endl << x;</pre>	<pre>-1 -1 101 101 1236 1293786 -123123 -123 -2 123 100</pre>	100	100	✓
✓	<pre>int x; x = enteroValidado(); cout << endl << x;</pre>	<pre>-1 -124 -2134 324 256 3134 32</pre>	32	32 32	✓

Todas las pruebas superadas. ✓

Question author's solution:

```
int enteroValidado()
{
    int a;
    cout << "Ingrese un valor entero en el rango [0,100]: ";
    cin >> a;
    while(a<0 || a>100)
    {
        cout << "Ingrese un valor entero en el rango [0,100]: ";
        cin >> a;
    }
    return a;
}
```

Correcta

Puntos para este envío: 1,00/1,00.

20/10/18 20:58

Pregunta 9

Correcta

Puntúa 1,00 sobre 1,00

Defina una función en C++ de tipo int llamada flotanteValidado que no acepta ningún argumento. La función debe solicitar que se ingrese por teclado un valor flotante, valide que esté en el rango de -10 a 0 (ambos inclusive) o del 20 al 30 (ambos inclusive) y luego lo devuelva como valor de retorno. Si se ingresan valores fuera del rango se debe volver a solicitar el ingreso nuevamente, tantas veces como sea necesario.

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```
1 float flotanteValidado()
2 {
3     float a;
4
5     do
6     {
7         cin>>a;
8         if(a>-10 && a<0)
9         {
10            cout<<a;
11        }
12        else if(a>20 && a<30)
13        {
14            cout<<a;
15        }
16    }
17    while((a<-10||a>0)&& (a<20||a>30) );
18    return a;
19
20 }
```

	Test	Input	Expected
✓	float x; x = flotanteValidado(); cout << endl << fixed << setprecision(2) << x;	-10.1 -10.1 -10.1 0.1 0.1 0.1 19.9 19.9 30.1 30.1 30.1 -2.52	-2.52

	Test	Input	Expected
✓	<pre>float x; x = flotanteValidado(); cout << endl << fixed << setprecision(2) << x;</pre>	-10.1 -10.1 -10.1 0.1 0.1 0.1 19.9 19.9 30.1 30.1 30.1 22.13	22.13
✓	<pre>float x; x = flotanteValidado(); cout << endl << fixed << setprecision(2) << x;</pre>	-10.1 -10.1 -10.1 0.1 0.1 0.1 19.9 19.9 30.1 30.1 30.1 -10.0	-10.00
✓	<pre>float x; x = flotanteValidado(); cout << endl << fixed << setprecision(2) << x;</pre>	-10.1 -10.1 -10.1 0.1 0.1 0.1 19.9 19.9 30.1 30.1 30.1 0.0	0.00
✓	<pre>float x; x = flotanteValidado(); cout << endl << fixed << setprecision(2) << x;</pre>	-10.1 -10.1 -10.1 0.1 0.1 0.1 19.9 19.9 30.1 30.1 30.1 20.0	20.00

	Test	Input	Expected
✓	<pre>float x; x = flotanteValidado(); cout << endl << fixed << setprecision(2) << x;</pre>	-10.1 -10.1 -10.1 0.1 0.1 0.1 19.9 19.9 30.1 30.1 30.1 30.0	30.00

Todas las pruebas superadas. ✓

Question author's solution:

```
float flotanteValidado()
{
    float a;
    cout << "Ingrese un valor flotante en el rango [-10,0] o [20,
30]: ";
    cin >> a;
    while(!((a>=-10 && a<=0)|| (a>=20 && a<=30)))
    {
        cout << "Ingrese un valor flotante en el rango [-10,0] o [20,
30]: ";
        cin >> a;
    }
    return a;
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 10

Correcta

Puntúa 1,00 sobre 1,00

Defina una función en C++ llamada esPar que acepta un valor entero como argumento y devuelve true(1) si el número es par y false(0) en caso contrario.

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```
1 bool esPar(int num)
2 {
3     bool a;
4     if(num%2==0)
5         a=true;
6     else if(num%2==1)
7         a=false;
8     return a;
9 }
```

	Test	Expected	Got	
✓	cout << esPar(2); cout << endl; cout << esPar(1); cout << endl;	1 0	1 0	✓
✓	cout << esPar(21232);	1	1	✓
✓	cout << esPar(-122);	1	1	✓
✓	cout << esPar(223423);	0	0	✓
✓	cout << esPar(-1234323);	0	0	✓

Todas las pruebas superadas. ✓

Question author's solution:

```
bool esPar(int num)
{
    return num%2==0;
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 11

Correcta

Puntúa 1,00 sobre
1,00

Defina una función en C++ llamada `es_primo` que acepte como argumento un número de tipo entero llamado `n` y devuelve `true` (1) si el número es primo y `false` (0) en caso contrario.

Un **número primo** es un numero natural mayor que 1 que tiene únicamente dos divisores distintos: él mismo y el 1.

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```

1  bool es_primo(int n)
2  {
3      int cuenta, vof;
4
5      cuenta=0;
6
7      for(int i=2;i<=n;i++)
8      {
9          if( n%i==0)
10             cuenta++;
11     }
12     if(n>=1&&(cuenta==1||cuenta==0))
13         vof=1;
14     else
15         vof=0;
16     return vof;
17 }
```

	Test	Expected	Got	
✓	<code>cout << es_primo(1);</code>	1	1	✓
✓	<code>cout << es_primo(2);</code>	1	1	✓
✓	<code>cout << es_primo(3);</code>	1	1	✓
✓	<code>cout << es_primo(4);</code>	0	0	✓
✓	<code>cout << es_primo(5);</code>	1	1	✓
✓	<code>cout << es_primo(7);</code>	1	1	✓
✓	<code>cout << es_primo(8);</code>	0	0	✓
✓	<code>cout << es_primo(9);</code>	0	0	✓
✓	<code>cout << es_primo(823);</code>	1	1	✓
✓	<code>cout << es_primo(12341);</code>	0	0	✓

Todas las pruebas superadas. ✓

Question author's solution:


```
bool es_primo(int n)
{
    for(int i=2; i<=(n/2); i++)
    {
        if(n%i==0)
        {
            return false;
        }
    }
    return true;
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 12

Correcta

Puntúa 1,00 sobre 1,00

Defina una función en C++ llamada polinom que acepta como argumento un número de tipo double llamado x y devuelve el resultado de evaluar x en la siguiente expresión:

$$4.12x^2 - 3.12x + 1.23$$

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```
1 double polinom(double x)
2 {
3     double p;
4     p=4.12*pow(x,2)-3.12*x+1.23;
5     return p;
6
7 }
```

	Test	Expected	Got	
✓	cout << polinom(4.12);	58.3101	58.3101	✓
✓	cout << polinom(-4.12);	84.0189	84.0189	✓
✓	cout << polinom(0.0);	1.23	1.23	✓
✓	cout << polinom(0.122);	0.910682	0.910682	✓
✓	cout << polinom(7.18);	191.224	191.224	✓

Todas las pruebas superadas. ✓

Question author's solution:

```
double polinom(double x)
{
    return 4.12 * x * x - 3.12 * x + 1.23;
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 13

Correcta

Puntúa 1,00 sobre 1,00

Definir una función en C++ de tipo double llamada mayorDeTres que acepta tres argumentos de tipo double y devuelve el valor del mayor de los tres como valor de retorno.

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```
1 double mayorDeTres(double a, double b, double c)
2 {
3     double max;
4     if(a>=b&& a>=c)
5         max= a;
6     else if(b>=a&& b>=c)
7         max= b;
8     else if(c>=a&& c>=b)
9         max=c;
10
11     return max;
12 }
```

	Test	Expected	Got
✓	cout << mayorDeTres(1.12, 2.23, 323.1);	323.1	323.1
✓	cout << mayorDeTres(1.12, 2.23, -323.1);	2.23	2.23
✓	cout << mayorDeTres(19.12, 2.23, -323.1);	19.12	19.12
✓	cout << mayorDeTres(1.12, 1.12, 1.12);	1.12	1.12
✓	cout << mayorDeTres(-12.12, 1.12, 1.12);	1.12	1.12
✓	cout << mayorDeTres(1.12, -11.12, 1.12);	1.12	1.12
✓	cout << mayorDeTres(1.12, 1.12, -1.12);	1.12	1.12
✓	cout << mayorDeTres(-231.12, 1231.12, -12321.12);	1231.12	1231.12

Todas las pruebas superadas. ✓

Question author's solution:

```
double mayorDeTres(double a, double b, double c)
{
    double maximo;
    if(a>=b && a>=c)
        maximo = a;
    else if(b>=a && b>=c)
        maximo = b;
    else
        maximo = c;
    return maximo;
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Definir una función en C++ de tipo double llamada mod, que acepta como parametro las tres componentes de un vector en R^3 y devuelve el valor del módulo del vector.

$$\|(x,y,z)\| = \sqrt{x^2 + y^2 + z^2}$$

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```
1 double mod(double x, double y, double z)
2 {
3     double modulo;
4     modulo= sqrt(pow(x,2)+ pow(y, 2)+ pow(z, 2));
5     return modulo;
6 }
7 }
```

	Test	Expected	Got	
✓	cout << mod(1.12,2.23,323.1);	323.11	323.11	✓
✓	cout << mod(1.12,2.23,-323.1);	323.11	323.11	✓
✓	cout << mod(19.12,2.23,-323.1);	323.673	323.673	✓
✓	cout << mod(1.12,1.12,1.12);	1.9399	1.9399	✓
✓	cout << mod(-12.12,1.12,1.12);	12.2231	12.2231	✓
✓	cout << mod(1.12,-11.12,1.12);	11.2322	11.2322	✓
✓	cout << mod(1.12,1.12,-1.12);	1.9399	1.9399	✓
✓	cout << mod(-231.12,1231.12,-12321.12);	12384.6	12384.6	✓

Todas las pruebas superadas. ✓

Question author's solution:

```
double mod(double x, double y, double z)
{
    return sqrt(pow(x,2.0) + pow(y,2.0) + pow(z,2.0));
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Correcta

Puntúa 1,00 sobre
1,00

Definir una función en C++ de tipo double llamada polinom que acepta 4 argumentos de tipo double, A, B, C y x. La función debe devolver el resultado de evaluar el polinomio:

$$Ax^2 + Bx + C$$

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```
1 double polinom(double A, double B, double C, double x)
2 {
3     double p;
4     p=A*pow(x,2)+B*x+C;
5
6     return p;
7 }
```

	Test	Expected	Got	
✓	cout << polinom(1.12,2.23,323.1,3.14);	341.145	341.145	✓
✓	cout << polinom(11.12,-2.23,323.1,-7.14);	905.915	905.915	✓
✓	cout << polinom(0.12,2.23,323.1,0.14);	323.415	323.415	✓
✓	cout << polinom(-1,1,0,123.24);	-15064.9	-15064.9	✓
✓	cout << polinom(-0,1,1,12.24);	13.24	13.24	✓
✓	cout << polinom(-5.12,1.5982,0.12,-1);	-6.5982	-6.5982	✓

Todas las pruebas superadas. ✓

Question author's solution:

```
double polinom(double A, double B, double C, double x)
{
    return A*x*x + B*x + C;
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 16

Correcta

Puntúa 1,00 sobre
1,00

A un programador le dieron la siguiente consigna:

Escriba un programa que solicite un numero entero al usuario.

- Si el usuario ingresa 1, deberá solicitarle otro numero entero e interpretarlo como un año e informar si este es o no bisiesto.
- Si el usuario ingresa 2, deberá solicitar otros 3 números enteros e interpretarlos como día, mes y año. Al finalizar debe informar si la fecha ingresada es o no válida.

El programador dio la implementación que aparece a continuación. La implementación tiene algunos problemas:

- 1) No válida de manera adecuada la fecha para el caso en el que el mes es Febrero
- 2) Omite imprimir un mensaje cuando la fecha es inválida. Corrija la implementación.

Nota: Para el error 1) considere reutilizar la función `es_bisiesto`.

Por ejemplo:

Input	Resultado
1 2004	El anio es bisiesto

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```

1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5
6
7  bool es_bisiesto(int anio)
8  {
9      return (anio % 4 == 0) && ((anio % 100 != 0) || (anio % 400 == 0));
10 }
11
12 bool es_fecha_valida(int d, int m, int y)
13 {
14     if(y > 0 && 0 < m && m<13)
15     {
16         switch(m)
17         {
18             case 1:
19             case 3:
20             case 5:
21             case 7:
22             case 8:
23             case 10:
24             case 12:
25                 return d <= 31;
26             case 4:
27             case 6:
28             case 9:
29             case 11:
30                 return d <= 30;
```

	Input	Expected	Got
✓	1 2004	El anio es bisiestro	El anio es bisiestro
✓	1 2005	El anio no es bisiestro	El anio no es bisiestro
✓	2 2 10 2018	La fecha ingresada es valida	La fecha ingresada es val

	Input	Expected	Got
✓	2 32 10 2018	La fecha ingresada no es valida	La fecha ingresada no es
✓	2 31 0 2018	La fecha ingresada no es valida	La fecha ingresada no es
✓	2 31 13 2018	La fecha ingresada no es valida	La fecha ingresada no es
✓	2 31 13 2018	La fecha ingresada no es valida	La fecha ingresada no es
✓	2 31 9 2018	La fecha ingresada no es valida	La fecha ingresada no es
✓	2 30 9 2018	La fecha ingresada es valida	La fecha ingresada es val
✓	2 28 2 2018	La fecha ingresada es valida	La fecha ingresada es val
✓	2 29 2 2018	La fecha ingresada no es valida	La fecha ingresada no es
✓	2 29 2 2016	La fecha ingresada es valida	La fecha ingresada es val
✓	2 30 2 2017	La fecha ingresada no es valida	La fecha ingresada no es

	Input	Expected	http://lev2.eft.uncor.edu/mod/quiz/review.php?a...
✓	-10 0 15 2 29 2 2004	La fecha ingresada es valida	La fecha ingresada es val

Todas las pruebas superadas. ✓

Question author's solution:

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

bool es_bisiesto(int anio)
{
    return (anio % 4 == 0) && ((anio % 100 != 0) || (anio % 400 == 0));
}

bool es_fecha_valida(int d, int m, int y)
{
    if(y > 0 && 0 < m && m<13)
    {
        switch(m)
        {
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                return d
                <= 31;
            case 4:
            case 6:
            case 9:
            case 11:
                return d
                <= 30;
            case 2:
                return d
                <= 28 + es_bisiesto(y);
        }
    }
    return false;
}

int main(){
    int opcion, dia, mes, anio;

    do
    {
        cin>>opcion;
    }while(opcion!=1 && opcion!=2);

    if(opcion == 1)
```

```
{
    cin>>anio;
    if(es_bisiesto(anio))
        cout<<"El anio es bisiesto"<<endl;
    else
        cout<<"El anio no es bisiesto"<<endl;
}else if(opcion == 2)
{
    cin>> dia >> mes >> anio;
    if (es_fecha_valida(dia, mes, anio))
        cout<<"La fecha ingresada es valida"<<endl;
    else
        cout<<"La fecha ingresada no es valida"<<endl;
}

return 0;
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 17

Correcta

Puntúa 1,00 sobre 1,00

A un programador le dieron la siguiente consigna:

Escriba un programa que solicite un numero entero entre 1 y 3. Si el numero ingresado es 1, el programa deberá solicitar otro valor entero l y escribir un cuadrado de lado l . Si el numero es 2, deberá solicitar otros 2 números entero b, h e imprimir un rectángulo de base b y altura h . Si el usuario ingresa 3, deberá solicitar un numero entero l e imprimir el cuadrado de lado l y el rectángulo de base l y altura $l+2$.

El programador dio la implementación que aparece a continuación. Se observa que definió una única función para imprimir tanto cuadrados como rectángulos, lo cual muestra una buena **modularización del código en procedimientos reutilizables**. Sin embargo, cometió algunos errores en la invocación a las funciones. Corrija la implementación.

Por ejemplo:

Input	Resultado
1	* * * *
4	* * * * * * * * * * * * * * * *

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```

1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5
6  void dibujar_rectangulo(int base, int altura)
7  {
8      for(int i=0; i<base; i++)
9      {
10         for(int j=0; j<altura; j++)
11             cout<<"* ";
12         cout<<endl; //escribe toda una fila luego baja
13     }
14 }
15
16
17 int main(){
18     int opcion, l, b, h;
19
20     do
21         cin>>opcion;
22     while (1>opcion || opcion>3);
23
24     switch(opcion)
25     {
26         case 1:
27             cin>>l;
28             dibujar_rectangulo(l,l);
29             break;
30         case 2:
31             cin>>b>>h;

```

	Input	Expected	Got	
✓	1 4	* * * * * * * * * * * * * * * *	* * * * * * * * * * * * * * * *	✓
✓	2 3 6	* * * * * * * * * * * * * * * * * *	* * * * * * * * * * * * * * * * * *	✓
✓	3 3	* * * * * * * * * * * * * * * * * * * * * * * *	* * * * * * * * * * * * * * * * * * * * * * * *	✓
✓	3 4	* * * * * * * * * * * * * * * * *	* * * * * * * * * * * * * * * * *	✓
✓	3 2	* * * * * * * * * * * *	* * * * * * * * * * * *	✓
✓	-1 -2 4 2 6 2	* * * * * * * * * * * *	* * * * * * * * * * * *	✓

Question author's solution:

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

void dibujar_rectangulo(int base, int altura)
{
    for(int i=0; i<base; i++)
    {
        for(int j=0; j<altura; j++)
            cout<<"* ";
        cout<<endl;
    }
}

int main(){
    int opcion, l, b, h;

    do
        cin>>opcion;
    while (1>opcion || opcion>3);

    switch(opcion)
    {
        case 1:
            cin>>l;
            dibujar_rectangulo(l,l);
            break;

        case 2:
            cin>>b>>h;
            dibujar_rectangulo(b,h);
            break;

        case 3:
            cin>>l;
            dibujar_rectangulo(l,l);
            cout<<endl;
            dibujar_rectangulo(l,l+2);
            break;

    }

    return 0;
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 18

Correcta

Puntúa 1,00 sobre 1,00

Un programador escribió el siguiente programa que utiliza una función que el mismo definió. Sin embargo, no logra hacerlo compilar. Resuelva los inconvenientes de tal manera que los casos de prueba sean superados.

Por ejemplo:

Input	Resultado
10 20 30	El maximo es 30

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```

1  #include <iostream>
2  using namespace std;
3  double max2(double, double);
4
5  int main(){
6      double a, b, c, max;
7
8      cin>>a>>b>>c;
9      max=max2(max2(a,b), c);
10     cout<<"El maximo es "<< max << endl;
11     return 0;
12 }
13
14 double max2(double a, double b)
15 {
16     //    int c;
17     if(a>b)
18     {
19         return a;
20     //    b=c;
21     //    max2=max2(a,b);
22     }
23     else
24     {
25         return b;
26     //    a=b;    los nombres de las variables en la funcion no import
27     //    b=c;    que en el principal solo imprta el orden en que se lo
28
29     }
30 }
31

```

	Input	Expected	Got	
✓	10 20 30	El maximo es 30	El maximo es 30	✓

	Input	Expected	Got	
✓	5 3 2	El maximo es 5	El maximo es 5	✓
✓	100 101.5 101.4	El maximo es 101.5	El maximo es 101.5	✓
✓	3.3 5.4 5.45	El maximo es 5.45	El maximo es 5.45	✓

Todas las pruebas superadas. ✓

Question author's solution:

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

double max2(double, double);

int main(){
    double a, b, c;

    cin>>a>>b>>c;
    cout<<"El maximo es "<< max2(max2(a,b), c) << endl;
    return 0;
}

double max2(double a, double b)
{
    if(a>b)
        return a;
    else
        return b;
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 19

Correcta

Puntúa 1,00 sobre
1,00

A un programador le dieron la siguiente consigna:

Escriba un programa que implemente la función: *div2_non_error(double a, double b)*. Esta función debe chequear que *b* no sea 0 y en tal caso retornar el resultado de la división. En caso contrario, la función deberá retornar -1. Luego, invoque la función en su programa principal.

Un programador escribió el siguiente programa. Sin embargo, no logra hacerlo compilar. Corrija la implementación.

Por ejemplo:

Input	Resultado
10 20	10 / 20 = 0.5

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  double div2_non_error(double a, double b)
6  {
7      if (b!=0)
8          return a/b;
9      else
10         return -1;
11 }
12
13
14 int main(){
15     double a, b;
16
17     cin>>a>>b;
18     cout<< setprecision(2) << a << " / " << b << " = "<< div2_non_er
19     return 0;
20 }
```

	Input	Expected	Got	
✓	10 20	10 / 20 = 0.5	10 / 20 = 0.5	✓
✓	7.5 2.3	7.5 / 2.3 = 3.3	7.5 / 2.3 = 3.3	✓
✓	0 1	0 / 1 = 0	0 / 1 = 0	✓
✓	10 0	10 / 0 = -1	10 / 0 = -1	✓

Todas las pruebas superadas. ✓

Question author's solution:

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

double div2_non_error(double a, double b)
{
    if (b!=0)
        return a/b;
    else
        return -1;
}

int main(){
    double a, b;

    cin>>a>>b;
    cout<< setprecision(2) << a << " / " << b << " = "<< div2_non
_error(a,b);
    return 0;
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 20

Correcta

Puntúa 1,00 sobre 1,00

El **Coefficiente Binomial**, comúnmente denotado por $\binom{n}{k}$ corresponde al número de

formas en que se puede extraer subconjuntos de k elementos a partir de un conjunto de n elementos. Este numero de combinaciones se calcula de la siguiente manera:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

El **Triángulo de Pascal** es una representación de los coeficientes binomiales ordenados en forma triangular, tal como se muestra a continuación:

$$\begin{array}{ccccccc} & & & & \binom{0}{0} & & & & \\ & & & & \binom{1}{0} & & \binom{1}{1} & & \\ & & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} & \\ & \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} \end{array}$$

A un programador le solicitaron que escriba un programa, que solicite un numero entero h y dibuje el Triángulo de Pascal de altura h . Tal como se muestra en el siguiente ejemplo de ejecución:

```
Ingrese la altura del triangulo: 8
El numero maximo que aparecera en el triangulo es: 35
Tiene 2 cifras.
El triangulo de Pascal de 8 niveles es:
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
```

El programador dio la implementación que aparece a continuación. La implementación muestra una correcta **división del problema en sub-problemas**, una adecuada **modularización del código en funciones nombradas de manera descriptiva** y un apropiado **uso de comentarios**. Sin embargo, el programador cometió un pequeño error en al invocación a una función y el resultado impreso en pantalla no es correcto. Corrija la implementación.

Por ejemplo:

Input	Resultado
5	<p>Ingrese la altura del triangulo:</p> <p>El numero maximo que aparecera en el triangulo es: 6</p> <p>Tiene 1 cifras.</p> <p>El triangulo de Pascal de 5 niveles es:</p> <pre> 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 </pre>

Input	Resultado
6	<p>Ingrese la altura del triangulo:</p> <p>El numero maximo que aparecera en el triangulo es: 10 Tiene 2 cifras. El triangulo de Pascal de 6 niveles es:</p> <pre> 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 </pre>

Respuesta: (penalty regime: 0 %)

Reiniciar respuesta

```

1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5
6  int fact(int n);
7  int coef_binomial(int n, int m);
8  int contar_cifras(int n);
9
10 /*
11  Calcula el factorial de un numero entero n.
12  Notar que no se realiza ningun control de overflow.
13  13! no es representable en 32 bits.
14  */
15 int fact(int n)
16 {
17     int res = 1;
18     for(int i=2; i<=abs(n); i++)
19         res *= i;
20
21     return res;
22 }
23
24 /*
25 Calcula el coeficiente binomial de n elementos tomados en conjuntos
26 */
27 int coef_binomial(int n, int k)
28 {
29     return fact(n) / (fact(k) * fact(n-k));
30 }
31
32 /*
33 Cuenta la cantidad de cifras decimales que tiene
34 el numero n.
35 Ejemplo:
36     contar_cifras(100) = 3
37 */
38 int contar_cifras(int n)

```


	Input	Expected	Got
✓	5	<p>Ingrese la altura del triangulo:</p> <p>El numero maximo que aparecera en el triangulo es: 6 Tiene 1 cifras. El triangulo de Pascal de 5 niveles es:</p> <pre> 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 </pre>	<p>Ing</p> <p>El Tie El</p> <p>1 1</p>
✓	6	<p>Ingrese la altura del triangulo:</p> <p>El numero maximo que aparecera en el triangulo es: 10 Tiene 2 cifras. El triangulo de Pascal de 6 niveles es:</p> <pre> 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 </pre>	<p>Ing</p> <p>El Tie El</p>
✓	1	<p>Ingrese la altura del triangulo:</p> <p>El numero maximo que aparecera en el triangulo es: 1 Tiene 1 cifras. El triangulo de Pascal de 1 niveles es:</p> <pre> 1 </pre>	<p>Ing</p> <p>El Tie El</p> <p>1</p>

			http://lev2.efn.uncor.edu/mod/quiz/review.php?a...
	Input	Expected	Got
✓	8	<p>Ingrese la altura del triangulo:</p> <p>El numero maximo que aparecera en el triangulo es: 35</p> <p>Tiene 2 cifras.</p> <p>El triangulo de Pascal de 8 niveles es:</p> <div><div>1</div><div>11</div><div>121</div><div>1331</div><div>14641</div><div>15101051</div><div>1615201561</div><div>172135352171</div></div>	<p>Ing</p> <p>El</p> <p>Tie</p> <p>El</p>

Todas las pruebas superadas. ✓

Question author's solution:

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

int fact(int n);
int coef_binomial(int n, int m);
int contar_cifras(int n);

/*
    Calcula el factorial de un numero entero n.
    Notar que no se realiza ningun control de overflow.
    13! no es representable en 32 bits.
*/
int fact(int n)
{
    int res = 1;
    for(int i=2; i<=abs(n); i++)
        res *= i;

    return res;
}

/*
    Calcula el coeficiente binomial de n elementos tomados en conjuntos de k elementos.
*/
int coef_binomial(int n, int k)
{
    return fact(n) / (fact(k) * fact(n-k));
}

/*
    Cuenta la cantidad de cifras decimales que tiene el numero n.
    Ejemplo:
        contar_cifras(100) = 3
*/
int contar_cifras(int n)
{
    int cifras = 1;

    while(n/10 > 0)
    {
        cifras += 1;
        n/=10;
    }

    return cifras;
}

//Programa principal
int main()
```

```

{
    http://lev2.efn.uncor.edu/mod/quiz/review.php?a...
    //h almacenara la cantidad de niveles del triangulo.
    int h;
    cout<<"Ingrese la altura del triangulo: ";
    cin>>h;
    cout<<endl;

    /*
    Calculo el maximo valor que apareciera en el triangulo
    para definir la cantidad de lugares que ocupara cada num
    ero
    de esta manera lograre la alineacion de los valores.
    */
    int max_valor_binomial = coef_binomial(h-1, (h-1)/2 + (h-
    1)%2);

    /*
    La cantidad de lugares que ocupara un numero en e
    l triangulo sera
    La cantidad de posiciones que ocupe el numero mas
    grande.
    */
    int cifras = contar_cifras(max_valor_binomial);

    cout<<"El numero maximo que apareciera en el triangulo es:
    "<< max_valor_binomial <<endl;
    cout<<"Tiene "<< cifras <<" cifras."<<endl;
    cout<<"El triangulo de Pascal de "<< h <<" niveles es: "
    <<endl<<endl;

    /*
    Imprimo triangulo de Pascal. Cada numero ocupa (p
    osiblemente con espacios en blanco)
    tantas posiciones como es el valor de cifras.
    */
    for(int nivel=0; nivel<h; nivel++)
    {
        // imprimo el nivel "nivel"
        // comienzo con cifras*(h-1-nivel) espacios en bl
        anco

        // para alinear este nivel con los sucesivos
        for(int i=0; i<cifras*(h-nivel); i++)
            cout<<" ";

        //imprimo los valores de los coeficientes binomia
        les del nivel "nivel"
        for(int j=0; j<nivel+1; j++)
        {
            cout<< setw(cifras)<<right << coef_binomi
            al(nivel, j);

            // Dejo cifras lugares en blanco para log
            rar la alineacion requerida
            for(int i=0; i<cifras; i++)

```

```
        }  
        cout<<endl;  
    }  
    cout<<endl;  
  
    return 0;  
}
```

Correcta

Puntos para este envío: 1,00/1,00.

◀ Recursos Adicionales Clase 8

[Ir a...](#)

Guía de Ejercicios 9 ▶