

Pregunta 1

Sin responder  
aún

Puntúa como  
2,00

🚩 Marcar  
pregunta

Con respecto al proceso de **depuración de un programa**, indicar cuál de las opciones es CORRECTA:

Seleccione una:

- ☒ a. Los errores fatales no permiten llevar a cabo la compilación, mientras que las advertencias o warnings sólo indican condiciones sospechosas aunque legítimas que no impiden la compilación.
- ☐ b. El compilador puede detectar errores de cualquier tipo, ya sean lógicos, de sintaxis y/o de ejecución.
- ☐ c. Los errores que violan las reglas gramaticales de un lenguaje de programación no impiden que se realice la compilación.
- ☐ d. Se denomina así al proceso de organizar el código fuente de un programa.

[Quitar mi elección](#)

Pregunta 2

Respuesta  
guardada

Puntúa como  
2,00

🚩 Marcar  
pregunta

Con respecto a los **tipos de datos** en C++, indicar cuál de las opciones es INCORRECTA:

Seleccione una:

- ☒ a. En los prototipos de funciones no es necesario indicar el tipo de dato de los parámetros, pero sí sus identificadores.
- ☐ b. Es posible convertir un tipo char a un tipo float.
- ☐ c. Es posible dividir un valor entero por un valor booleano.
- ☐ d. Las constantes definidas -mediante DEFINE- no poseen un tipo de dato.

[Quitar mi elección](#)

Pregunta **3**

Respuesta  
guardada

Puntúa como  
2,00

🚩 Marcar  
pregunta

En relación a las **constantes**, indicar cuál de las siguientes afirmaciones es INCORRECTA:

Seleccione una:

- ☐ a. Las constantes definidas se indican mediante instrucciones al preprocesador, de la forma: #define CONSTANTE valor
- ☒ b. Pueden ser locales o globales, al igual que las variables.
- ☐ c. Para las constantes declaradas el compilador reserva espacio en memoria, en base a su tipo de dato.
- ☐ d. Las constantes definidas pueden pasarse por referencia, a una función, como parámetro.

[Quitar mi elección](#)

Pregunta **4**

Respuesta  
guardada

Puntúa como  
2,00

🚩 Marcar  
pregunta

En relación a las **estructuras de selección**, indicar cuál de las siguientes sentencias es CORRECTA:

Seleccione una:

- ☐ a. En una sentencia IF anidada, para indicarle al compilador la asociación de los IF con los ELSE, no se requiere el uso de llaves sino que bastará con que el IF y su ELSE correspondiente se escriban utilizando la misma indentación.
- ☐ b. La estructura SWITCH, que permite seleccionar una de entre múltiples alternativas, producirá un comportamiento en cascada en aquellos casos en los que esté ausente la sentencia break.
- ☒ c. Una selección incompleta es una sentencia IF que no posee un ELSE y que indicará el camino de ejecución a seguir cuando su condición sea falsa.
- ☐ d. El anidamiento de sentencias IF es menos eficiente que escribir una secuencia equivalente de IFs (una sentencia IF debajo de otra, y de otra, y de otra.... todas con el mismo nivel de indentación).

[Quitar mi elección](#)

Pregunta 5

Respuesta  
guardada

Puntuación como  
2,00

🚩 Marcar  
pregunta

Con respecto a las **variables**, indicar cuál de las opciones es CORRECTA:

Seleccione una:

- ☐ a. No es necesario reservar espacio en memoria para las variables globales puesto que éstas pueden accederse desde cualquier lugar del programa.
- ☐ b. La asignación de un valor inicial debe realizarse al momento de declarar una variable.
- ☐ c. En caso de tener declaraciones de diferentes variables con el mismo nombre (identificador) en diferentes niveles de anidamiento, la variable externa ocultará a la interna.
- ☒ d. La declaración de variables locales reserva un espacio de memoria, que contiene un valor "basura" hasta tanto no se le asigne un valor inicial.

[Quitar mi elección](#)

Pregunta 6

Respuesta  
guardada

Puntúa como  
2,00

🚩 Marcar  
pregunta

En relación a las **funciones** en C++, indicar cuál de las siguientes sentencias es INCORRECTA:

Seleccione una:

- ☒ a. Un prototipo y la cabecera de una función deben coincidir en el orden y tipo de dato de sus parámetros, aunque los identificadores de éstos solo se requieren en la cabecera de la función.
- ☐ b. Las llamadas a las funciones que retornan un valor no son instrucciones independientes, es decir, no deben ser la única operación en la línea o sentencia.
- ☐ c. Cuando se pasa un vector a una función, el pasaje se realiza por referencia y en la cabecera de la función se indica:  
tipo-retorno nombre-función (tipo-dato & nombre-arreglo, .....)
- ☐ d. La función que invoca a otra función se queda en espera mientras se ejecuta la función invocada, y ésta, al finalizar, devuelve el control a la función llamadora.

[Quitar mi elección](#)



Pregunta **7**

Respuesta  
guardada

Puntúa como  
2,00

🚩 Marcar  
pregunta

Con respecto a los **arreglos**, indicar cuál de las siguientes declaraciones es **INCORRECTA**, teniendo la directiva **#define TAM 30** :

Seleccione una:

- ☐ a. `int arreglo[TAM]={0};`
- ☐ b. `int n, arreglo[30];`
- ☐ c. `int n, arreglo[TAM];`
- ☒ d. `int n=TAM, arreglo[n];`

[Quitar mi elección](#)

Pregunta 8

Respuesta  
guardada

Puntuación como  
4,00

🚩 Marcar  
pregunta

Dada la siguiente función:

```
void funcion(int x){  
    int b=4;  
    if (x > 10 and x%2!=0) {  
        if(x%3==0)  
            cout << b; }  
    else  
        if ((!x) and (pow(b,3)/0))  
            cout << x--;  
    else cout << x*b++;  
}
```

Escribir cuál es la salida para cada una de las siguientes invocaciones:

- 1) funcion(8);
- 2) funcion(12);
- 3) funcion(0);
- 4) funcion(157);



- 1) La salida es: 32
- 2) La salida es: 48
- 3) La salida es: 0
- 4) La salida es: 4



Pregunta 9

Respuesta  
guardada

Puntúa como  
2,00

🚩 Marcar  
pregunta

En relación a las **estructuras repetitivas**, indicar cuál de las siguientes sentencias es CORRECTA:

Seleccione una:

- ☒ a. Las estructuras inexactas (WHILE y DO..WHILE) se utilizarán cuando la cantidad de iteraciones a realizar pueda variar entre cada ejecución de un programa, mientras que una estructura exacta (FOR) se utilizará cuando la cantidad de iteraciones se mantenga constante.
- ☐ b. El uso de cualquier estructura repetitiva implica una bifurcación en el flujo de ejecución del programa.
- ☐ c. Según lo estudiado en clases, las 3 estructuras repetitivas vistas deben contener en su cuerpo una sentencia donde la variable de control actualice su valor de manera que en algún momento la condición de iteración se vuelva falsa y se salga del bucle.
- ☐ d. El funcionamiento correcto de la estructura FOR es el siguiente:
  - 1. se inicializa la variable de control (vc)
  - 2. se ejecuta el cuerpo del FOR
  - 3. se incrementa/decrementa la vc
  - 4. se chequea la condición del FOR: si el resultado es verdadero se ejecuta nuevamente el cuerpo del FOR, si es falso se sale del FOR.

[Quitar mi elección](#)

Pregunta 10

Respuesta  
guardada

Puntuación como  
4,00

🚩 Marcar  
pregunta

Indicar cuál es la salida CORRECTA para el siguiente código:

```
int n=5, vec[]={63,4,17,29,38};

for(int i=0; i<n; i+=2)
    vec[i%2]=vec[i]/2;

for(int i=0; i<n; i++)
    cout << vec[i] << "  ";
```

Seleccione una:

- ☒ a. 19 4 17 29 38
- ☐ b. 0 1 2 29 38
- ☐ c. 31 4 17 29 19
- ☐ d. 0 2 4 6 8

[Quitar mi elección](#)

Pregunta **11**

Respuesta  
guardada

Puntúa como  
2,00

🚩 Marcar  
pregunta

En relación a la **recursión**, indicar cuál de las siguientes afirmaciones es INCORRECTA:

Seleccione una:

- ☐ a. Una función es recursiva si, como parte de su definición, existe una llamada a sí misma.
- ☒ b. Las funciones recursivas, a diferencia de las funciones iterativas, no pueden tener void como tipo de retorno.
- ☐ c. El caso general de toda función recursiva incluye una llamada a sí misma con un tamaño de datos menor.
- ☐ d. La recursión es una técnica que emplea mucha memoria debido a las sucesivas llamadas a la misma función.

[Quitar mi elección](#)

Pregunta 12

Respuesta  
guardada

Puntuación como  
2,00

🚩 Marcar  
pregunta

Indicar cuál es la salida CORRECTA para el siguiente código:

```
int contador=0;
while (contador<5){
    switch(contador%3){
        case 1:
            cout << "1";
            break;
        case 2:
            cout << "2";
        case 3:
            cout << "3";
            break;
        default:
            cout << "Fin";
    }
    contador++;
}
```

Seleccione una:

- ☐ a. 1123231
- ☐ b. Fin1231Fin
- ☒ c. Fin123Fin1
- ☐ d. Fin12FinFin

[Quitar mi elección](#)

Pregunta **13**

Respuesta  
guardada

Puntúa como  
2,00

🚩 Marcar  
pregunta

Con respecto al concepto de **Algoritmo**, indicar cuál de las opciones es INCORRECTA:

Seleccione una:

- ☐ a. La secuencia de pasos de un algoritmo está orientada a conseguir un objetivo, es indicada con precisión y sigue un orden.
- ☒ b. El tiempo que le lleva a un algoritmo resolver un problema no está acotado dado que no es importante.
- ☐ c. La codificación de un algoritmo utilizando un determinado lenguaje de programación, genera un programa de computadora.
- ☐ d. Los algoritmos son independientes de los lenguajes de programación.

[Quitar mi elección](#)