


Pregunta 1

Incorrecta

Puntúa 0,00
sobre 1,00
 Marcar
pregunta


En relación a los **arreglos** en C++, indicar cuál de las afirmaciones es correcta:

Seleccione una:

- ☐ a. Dado que el nombre de un arreglo es un puntero a su primer elemento, es posible asignar un arreglo a un puntero, haciendo: `int arr[5], *punt; punt=arr;`
- ☒ b. La eliminación de 1 elemento de un arreglo consiste en correr 1 posición hacia la izquierda todos los valores que se encuentren a la derecha del valor que se quiere borrar. ✖
- ☐ c. El prototipo correcto de una función Mayor que permite obtener el mayor elemento de un arreglo de 5 enteros es: `void Mayor(int & arr[], int tl, int & may);`
- ☐ d. El acceso al último elemento de un arreglo requiere de más tiempo que si se accede al primer elemento (especialmente en vectores grandes, por ejemplo de 50000 elementos).

Pregunta 2

Correcta

Puntúa 1,00
sobre 1,00
 Marcar
pregunta


En relación a los **archivos** en C++, indicar cuál de las afirmaciones es incorrecta:

Seleccione una:

- ☐ a. Un archivo de texto sólo puede accederse de manera secuencial, mientras que un archivo binario puede accederse de 2 maneras: secuencial y directa.
- ☒ b. La diferencia entre un archivo de texto y un archivo binario está dada por la forma de acceder a los datos. ✔
- ☐ c. Todo programa que utilice un archivo, ya sea binario o de texto, debe indicar si el archivo será de lectura (declarando una variable de tipo `ifstream`), de escritura (declarando una variable de tipo `ofstream`), o de lectura/escritura (declarando una variable de tipo `fstream`).
- ☐ d. Para volver a leer la primera posición de un archivo, se debe cerrarlo y hacer un `open` si el archivo es de texto, o hacer `seekg(0)` si el archivo es binario.

Pregunta 3

Correcta

Puntúa 1,00
sobre 1,00
 Marcar
pregunta


En relación a las **matrices** en C++, indicar cuál de las afirmaciones es correcta:

Seleccione una:

- ☐ a. El algoritmo recomendado para determinar cuántas filas de suma cero tiene una matriz cuadrada es un `while` anidado.
- ☐ b. Determinar el elemento máximo por debajo de la diagonal principal tiene complejidad \sqrt{N} .
- ☐ c. Determinar cuántos elementos de la diagonal secundaria, son mayores que el promedio de los elementos de la misma diagonal, se puede resolver con una única pasada sobre dicha diagonal (por ejemplo con un ciclo `for`).
- ☒ d. El algoritmo más óptimo para determinar si todos los elementos de la diagonal principal de una matriz cuadrada son no nulos es un `while`. ✔


Pregunta 4

Correcta

Puntúa 1,00
sobre 1,00
 Marcar
pregunta


En relación a los **structs** en C++, indicar cuál de las afirmaciones es incorrecta:

Seleccione una:

- ☐ a. Es posible asignar una estructura E1 a otra estructura E2 del mismo tipo de 2 maneras: campo por campo o mediante la asignación de la estructura completa ($E2=E1$).
- ☐ b. El uso del operador flecha ($-->$) para acceder a sus campos solo es posible si el acceso se realiza mediante un puntero.
- ☒ c. `struct alumno{ char nombre[10]; int edad; };` reserva un espacio de 14 bytes en memoria. 
- ☐ d. Se utilizan en combinación con los punteros para definir los nodos de las listas enlazadas.


Pregunta 5

Correcta

Puntúa 1,00
sobre 1,00
 Marcar
pregunta


En relación a los **tipos de datos abstractos (TDA)** en C++, indicar cuál de las afirmaciones es correcta:

Seleccione una:

- ☐ a. Si se modifica la implementación del TDA (sin cambiar la interfaz), los programas que trabajen con él se verán afectados y deberán modificarse también.
- ☐ b. El acceso a los campos de un TDA se realiza indicando el nombre del TDA, el operador punto (.) y el nombre de los campos.
- ☐ c. El programador que utilice el TDA debe conocer no solo la representación y la interfaz sino también la implementación de las operaciones que se pueden realizar con él.
- ☒ d. Los TDA permiten ampliar el lenguaje dado que son tipos de datos definidos por el programador en base a tipos de datos provistos por C++, junto con las operaciones que pueden ejecutarse sobre ellos. 


Pregunta 6

Incorrecta

Puntúa 0,00
sobre 1,00
 Marcar
pregunta


En relación a los **punteros** en C++, indicar cuál de las afirmaciones es incorrecta:

Seleccione una:

- ☐ a. Un incremento de 1 en una variable puntero (por ejemplo, `Ptr++`) es relativo al tipo de dato apuntado, lo que implica mover a la derecha tantos bytes como lo indica dicho tipo de dato.
- ☒ b. Los nombres de los vectores son punteros constantes, por lo tanto no pueden modificar su contenido, es decir, no pueden apuntar a otras direcciones de memoria que no sean las que recibieron cuando fueron declarados. 
- ☐ c. Es posible definir variables punteros a datos de cualquier tipo, inclusive a otro puntero.
- ☐ d. Desreferenciar un puntero significa obtener la dirección de memoria del puntero.

Pregunta 7

Finalizado

Puntúa como
1,00
 Marcar
pregunta

Sea una lista enlazada simple L, cuyos nodos contienen un campo info de tipo char. Si el contenido de la lista es:

L -> 'm' -> 'n' -> 'o' -> 'p' -> 'q' -> 'r' -> 's' -> 't' -> 'u' -> 'v' -> 'w' -> 'x' -> 'y' -> 'z' -> 'a' -> NULL
y sea el siguiente código:

```
struct NodoLista {
    char info;
    struct NodoLista * sig; };
typedef NodoLista * NodoListaPtr;

void WTF( NodoListaPtr L, int T){
    NodoListaPtr Aux= L;   int i=0,c;
    while ( Aux != NULL ){
        cout << Aux->info << " ";
        i = (i+1) % T;
        c = 0;

        while (c <= i and Aux!=NULL) {
            Aux = Aux->sig; c++;
        }
    }
}
```

realice un seguimiento del código de la función WTF y muestre la salida del mismo para la siguiente llamada:

WTF(L, 3);


Puede usar el espacio para escribir, y completar allí el valor que toman las variables que considere seguir como prueba

La salida por pantalla de la siguiente llamada WTF(L, 3); es:

m
o
r
s
•


Pregunta 8

Correcta

Puntúa 1,00
sobre 1,00 Marcar
pregunta


En relación a los **métodos de ordenamiento**, indicar cuál de las afirmaciones es correcta:

Seleccione una:

- ☐ a. El método de la burbuja, obtiene el elemento más pequeño en cada pasada y lo compara contra cada uno de los elementos restantes.
- ☐ b. De los métodos de ordenamiento simples vistos, el ordenamiento por inserción es el de menor complejidad.
- ☒ c. El método merge sort divide una estructura desordenada en dos estructuras más pequeñas, ordena cada una de ellas de manera recursiva aplicando el ordenamiento por intercalación, y finalmente mezcla estas dos estructuras obteniendo una sola ordenada. 
- ☐ d. En el ordenamiento por selección los elementos se mueven varias veces hasta que todos hayan quedado ordenados, mientras que en el ordenamiento por inserción cada elemento se mueve como máximo 1 vez.


Pregunta 9

Correcta

Puntúa 1,00
sobre 1,00 Marcar
pregunta

En relación a las **cadenas**, indicar cuál de las afirmaciones es correcta:

Seleccione una:

- ☐ a. En C++ es posible manejar las cadenas de caracteres con 2 alternativas: cadenas al estilo C, que terminan en un carácter nulo y cadenas tipo string; ambas requieren que la asignación del valor inicial se realice en la línea de la declaración.
- ☐ b. Es posible asignar una cadena estilo C a otra, haciendo: cadena1 = cadena2, siempre que cadena2 tenga menor longitud que cadena1;
- ☒ c. Las cadenas tipo string hacen crecer el espacio de almacenamiento, por encima de la memoria asignada inicialmente, para acomodarse al tamaño de los datos almacenados. 
- ☐ d. Ni los strings ni las cadenas estilo C pueden declararse como constantes dentro de una función.