

### Pregunta 1

Sin responder aún

Puntúa como 1,00

Se tienen las siguientes sentencias dentro de main:

```
struct materia { string nombre, profesor;} Mat;
```

```
materia * ptr=&Mat;
```

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

Seleccione una:

- ☐ a. Es posible inicializar Mat haciendo:  
(\*ptr).nombre="AEDD";  
(\*ptr).profesor="Daniel Ambort";
- ☒ b. Dado un arreglo (local) de 5 elementos materia, se requiere una función que devuelva true si un profesor pasado como parámetro dicta alguna de las materias del arreglo, y para ello es válido el siguiente prototipo:  
bool dictaMat(string profe);
- ☐ c. Para poder escribir los valores de Mat en un archivo binario se debe hacer:  
ofstream archivo;  
archivo.open("materias.bin");  
archivo.write((char \*)&Mat,sizeof(materia));
- ☐ d. Las sentencias siguientes son válidas para declarar un arreglo de 5 materias y almacenar valores en la posición 0:  
materia vector[5];  
vector[0]=Mat;

[Quitar mi elección](#)

Pregunta **2**

Sin responder aún

Puntúa como 1,00

Indicar Verdadero o Falso y justificar por qué ha elegido esa alternativa.

**No existe ninguna relación entre arreglos y punteros.**



Párrafo



Falso. Podemos relacionar los arreglos con los punteros ya que, el nombre de un arreglo es un puntero hacia el primer elemento del a en cuestión.





Pregunta **3**

Sin responder aún

Puntúa como 1,00

Indicar Verdadero o Falso y justificar por qué ha elegido esa alternativa.

**Las variables dinámicas no tienen un identificador asociado y se acceden exclusivamente a través de punteros.**

	<input type="text" value="Párrafo"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>		<input type="text"/>	
---	--------------------------------------	----------------------	----------------------	----------------------	----------------------	----------------------	---	----------------------	---



Falso. Se considera que una variable es dinámica porque su tamaño y posición varía durante el módulo.



---



Pregunta **4**

Sin responder aún

Puntúa como 1,00

Indicar Verdadero o Falso y justificar por qué ha elegido esa alternativa.

**En un archivo binario, el acceso a los datos se realiza exactamente de la misma manera que en un archivo de texto.**



Párrafo





Verdadero. El acceso a los datos en un archivo binario se realiza de la misma manera que en un archivo de texto, la diferencia que podemos encontrar en estos dos tipos de archivos es el formato en el que se almacena la información.



«	»



Pregunta **5**

Sin responder aún

Puntúa como 1,00

Con respecto a las **listas enlazadas**, indicar cuál de las afirmaciones es incorrecta:

Seleccione una:

- ☐ a. Los nodos de una lista enlazada contienen un espacio para almacenar los datos y un puntero a otra estructura del mismo tipo.
- ☐ b. Para la reserva y liberación del espacio en memoria para cada nodo se emplean las sentencias new y delete respectivamente; dichas operaciones se realizan en tiempo de ejecución.
- ☐ c. El prototipo: void insertarNodo(Nodo \*\*lista, int valor); es válido si se quiere insertar un nuevo nodo con un valor entero como dato dentro de la lista enlazada.
- ☒ d. El final de una lista enlazada está indicado mediante un nodo que carece de puntero al siguiente nodo.

[Quitar mi elección](#)



Pregunta **6**

Respuesta guardada

Puntúa como 1,00

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

Seleccione una:

- ☒ a. La sentencia `char cadena[5]= "UTN-FRSF";` mostrará un error de compilación.
- ☐ b. Las siguientes sentencias: `char cadena1[5]="UTN-", cadena2[7]="FRSF"; cout << strncat(cadena1,cadena2,3);` mostrarán el siguiente resultado por pantalla: UTN-FRS
- ☐ c. Las cadenas de tipo `string` realizan un manejo automático de la memoria principal, es decir que no necesitan de la intervención del programador para modificar su espacio de almacenamiento.
- ☐ d. Son estructuras de datos que pueden ser tanto estáticas (cadenas estilo C) como dinámicas (strings); las cadenas dinámicas se declaran haciendo: `char * ptr = new cadena;`

[Quitar mi elección](#)



Pregunta **7**

Sin responder aún

Puntúa como 1,00

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

Seleccione una:

- ☐ a. El método de inserción es recomendable solo para un número pequeño de elementos.
- ☒ b. 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.
- ☐ c. El método merge sort es el más eficiente de los métodos de ordenamiento vistos, sin embargo tiene como desventaja el requerir de espacio extra para ordenar los valores.
- ☐ d. El número de intercambios realizados en el método de la burbuja es menor que en el de selección.

[Quitar mi elección](#)



Pregunta **8**

Respuesta guardada

Puntúa como 1,00

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

Seleccione una:

- ☒ a. Pueden ser de cualquier tipo de dato, excepto de tipo puntero.
- ☐ b. Pueden existir constantes de diferente ámbito con igual identificador.
- ☐ c. Las constantes definidas, #define W 10 por ejemplo, ocupan espacio en la memoria principal.
- ☐ d. No pueden participar de expresiones (combinación de operandos y operadores).

[Quitar mi elección](#)



Pregunta 9

Respuesta guardada

Puntúa como 1,00

En relación a **complejidad**, indicar cuál es la afirmación incorrecta:

Seleccione una:

- ☐ a. En una matriz cuadrada, calcular la mayor diferencia en valor absoluto, entre dos elementos pertenecientes a la diagonal principal, tiene complejidad  $O(1)$ .
- ☐ b. Ordenar de manera descendente los nodos de una lista enlazada tiene complejidad  $O(N^2)$ .
- ☐ c. Imprimir los elementos pares de la diagonal secundaria de una matriz cuadrada tiene complejidad  $O(N)$ .
- ☒ d. Mezclar 2 arreglos de  $V1$  y  $V2$  elementos tiene complejidad  $O(V1) + O(V2)$ .

[Quitar mi elección](#)



Pregunta **10**

Respuesta guardada

Puntúa como 1,00

En relación a la **recursión**, cuál de las siguientes afirmaciones es incorrecta?

Seleccione una:

- ☐ a. Su diseño debe garantizar que se llegue al caso base.
- ☒ b. Pueden tener más de un caso base y más de un caso general.
- ☐ c. Suelen requerir mayor cantidad de recursos que su equivalente iterativo.
- ☐ d. Siempre son soluciones menos elegantes que su equivalente iterativo.

[Quitar mi elección](#)





Pregunta **11**

Respuesta guardada

Puntúa como 1,00

Indicar Verdadero o Falso y justificar por qué ha elegido esa alternativa.

**Las colas y las pilas son estructuras de datos que se pueden implementar sobre listas enlazadas o sobre arreglos, con algunas restricciones.**



Párrafo



Verdadero. Las restricciones que podemos nombrar son:

En las pilas se permiten insercciones y borrados solo en la parte superior de la pila. En cambio, en las colas las insercciones se hacen e parte final y las eliminaciones en su parte inicial.



---



Pregunta **12**

Sin responder aún

Puntúa como 1,00

Indicar Verdadero o Falso y justificar por qué ha elegido esa alternativa.

La estructura de control (o combinación de estructuras de control) más óptima para saber si en cada fila de una matriz de 10 x 8 valores enteros hay algún elemento con el valor 7 es un for anidado.



Párrafo



Verdadero. Con la utilización de un for anidado vamos a poder recorrer toda la matriz (fila por fila y columna por columna) y así encor valor 7.



Ruta: p

---

«

»

<

^

Pregunta **13**

Sin responder aún

Puntúa como 1,00

En relación a las **funciones**, indicar cuál de las afirmaciones es incorrecta:

Seleccione una:

- ☒ a. No se requiere la utilización de un prototipo si la definición de una función se realiza antes que la definición de main.
- ☐ b. Los parámetros formales dentro de las funciones/procedimientos permiten acceder a las variables/valores que son utilizados en la invocación de éstos.
- ☐ c. Cuando se realiza el pasaje de un parámetro por valor o copia, el valor del parámetro formal no se modifica en ningún momento.
- ☐ d. En C++ las funciones que no devuelven un resultado se denominan procedimientos y en su invocación ésta es la única sentencia que aparece en la línea de código.

[Quitar mi elección](#)



## EJERCICIO EN MÁQUINA EXAMEN FINAL AEDD 3 DE MARZO DE 2021

### 2) Ejercicio OmegaUp - Regulares:

Ud. debe codificar un programa en C++ para subir como solución al problema “Desplazamientos y algo más” en OmegaUp.com. En el mismo tiene que:

1) **Codificar una función `Funcion1(...)`**, que reciba un vector `V` de elementos enteros (cada elemento del vector en el rango  $0..1000$ ) de tamaño lógico `TL` ( $3 \leq TL \leq 100$ ) y un parámetro `D` que indica la cantidad de “desplazamientos” a ejecutar a derecha ( $0 \leq D < TL$ ). Veamos un ejemplo:

Así, si `V = [ 1, 2, 3, 4 ]`, `TL=4`, la llamada a `Funcion1(V, TL, 1)` debe retornar en `V` los valores del vector una vez que todos se desplazaron `D` posiciones (1 posición en este caso) a la derecha (de manera circular en el caso del último valor). Así el contenido de `V` luego de la llamada a la función es: `[ 4, 1, 2, 3 ]`

2) Codificar una función `main()` que debe hacer lo siguiente:

- Leer por teclado un valor entero `F` ( $3 \leq F \leq 100$ ) que indica la cantidad de filas de una matriz cuadrada `M` de valores enteros (valores en  $0..1000$ ). Luego se ingresan por teclado los valores de la matriz, fila por fila y de izquierda a derecha.
- Visualizar la diferencia en valor absoluto entre el menor valor impar y el máximo valor par de la matriz (se asegura que en cada matriz ingresada existe como mínimo un valor par y un valor impar).
- Realizar una llamada a **`Funcion1( )`** pasándole como contenido cada una de las filas de la matriz, su tamaño lógico y como parámetro `D`, el valor del primer elemento de dicha fila (si este valor es mayor que `TL`, utilizar dicho valor  $\% TL$ ).
- Mostrar el contenido final de la matriz.

### Ejemplos de Entrada y Salida

4 1 2 3 8 4 1 2 0 3 2 1 3 5 4 3 2	7 8 1 2 3 4 1 2 0 2 1 3 3 2 5 4 3
3 2 2 6 4 4 7 2 0 6	1 2 6 2 7 4 4 0 6 2



## **AEDD - EXAMEN FINAL - 3/3/2021**

### ***Ejercicio de Práctica en Papel***

La vacunación es una prioridad que busca contribuir con el bienestar socio-sanitario de las personas y la comunidad en general y en el marco de la planificación de la Campaña de Vacunación contra Covid-19, es fundamental la divulgación de conocimientos claros y precisos.

Los interesados en vacunarse, deben completar un registro de inscripción. Una inscripción consta de: *DNI* (entero largo), *número de trámite* (11 caracteres alfanuméricos), *sexo* (char), *fecha de nacimiento* (tipo fecha), *patologías declaradas* (tipo patologías) y *grupo poblacional* (valor entero, donde 0 indica *60 años o más*, 1 indica *personal de la salud*, 2 indica *docentes y auxiliares* y 3 indica *edad entre 18 y 59 años*). Las patologías a ser declaradas corresponden a un dato booleano por cada una de las siguientes enfermedades: obesidad, diabetes, enfermedad cardiovascular. El valor *true* indica la presencia de la enfermedad, mientras que el valor *false* indica la ausencia.

En base a estas definiciones, se le solicita:

**a)** Defina todas las estructuras de datos necesarias para representar una lista de inscripciones. Considere que como máximo existirán 500000 inscripciones.

**b)** Defina e implemente la función *agregarInscripcion* la cual recibe una lista de inscripciones existentes y, por medio de un ingreso por teclado, agrega una nueva inscripción a la lista. Como resultado de la ejecución de esta función, la lista inicial debe quedar actualizada.

**c)** Defina e implemente la función *borrarInscripto* que recibe una lista de inscripciones junto con un número de trámite, y elimina del listado la inscripción cuyo número de trámite coincide con el recibido como argumento. Como resultado de la ejecución de esta función, la lista inicial debe ser actualizada.

**d)** Defina e implemente la función *porcentajeSexo* que recibe una lista de inscripciones e informa los porcentajes de inscriptos para cada sexo. Además, devuelve dos nuevas listas *inscriptosFemenino* e *inscriptosMasculino* que contienen las inscripciones asociadas a cada sexo.

**e)** Defina e implemente la función *existeDuplicado* que recibe una lista de inscripciones y muestra por pantalla los DNI de todas las personas que han registrado más de una inscripción.

**f)** Defina e implemente la función *verInscriptos* que recibe una lista de inscripciones y muestra por pantalla todas las inscripciones registradas.

**g)** Considerando que existe un arreglo *V* de 100 enteros que contiene los DNI de 100 personas que ya han recibido la vacuna, defina e implemente la función *porcentajesVacunadosPorGrupoPoblacional* que recibe una lista de inscripciones junto con

el arreglo V, y muestra por pantalla los porcentajes de personas vacunadas según el grupo poblacional al que pertenecen.

**Importante:** Para la resolución del problema el alumno puede codificar todas las funciones que considere necesarias. El puntaje final obtenido tendrá en cuenta la eficiencia de la estrategia de resolución elegida.