

Algoritmos y Estructuras de Datos

Examen Final 19/4/2023

Ejercicio de Práctica en Papel

Nombre y Apellido:

Año de Cursado:.....

El objetivo de este ejercicio es evaluar la aplicación integral de los contenidos de la materia. Para esto, evaluaremos la forma en la que interpretas código C++, la solución algorítmica desarrollada en base a la información definida en la consigna, y la aplicación de buenas prácticas de programación. Para la resolución, no tenes que alterar las estructuras de datos ni de programa ya brindadas. En los casos donde no se especifica una estructura en particular, tenes la libertad de diseñar las estructuras según tu criterio. En los casos en los que se solicita completar las estructuras propuestas, debes tener en cuenta lo descripto en la consigna. El puntaje final obtenido corresponderá al global de la solución entregada. No se califica por incisos.

La resolución puede entregarse en esta hoja con lapicera/lápiz o, alternativamente, en un archivo con extensión .cpp. Todas las entregas deben tener nombre y apellido (en caso de archivo .cpp, colocá un comentario al principio indicando también año de cursado). Tener en cuenta la legibilidad de la solución entregada para entregas en papel, ya que de no comprenderse lo escrito no se podrá corregir. Deben entregarse todas las hojas que contengan código asociado a la solución. Recuerda que, si haces uso de funciones auxiliares, debes incluirlas y/o codificarlas según corresponda.

Tiempo de Resolución: 90 minutos.

Puntaje Requerido: 24/40 puntos.

Consigna:

En una clínica veterinaria se atiende a los pacientes según el orden de llegada y el tipo de atención requerida. Para trasladar la atención a un sistema de software, se ha definido una lista de turnos en base a las siguientes estructuras de datos:

```
#define MAX 150

struct Turno {

    int numero; //orden de llegada.

    char tipo; //tipo de turno, siendo R=regular, U=urgencia, D=derivación.

};

struct NodoTurno {

    Turno t; //COMPLETAR .....

    NodoTurno* sig; //COMPLETAR .....

};
```

Tomando como base estas definiciones, se solicita:

a) Definí e implementá la función `crearVacia` que crea y retorna una lista enlazada de `NodoTurno` sin nodos asociados (es decir, vacía). La función, no debe recibir ningún argumento. Además, completa las líneas indicadas en la definición de `NodoTurno` con `//COMPLETAR` haciendo uso de un comentario que indique la utilidad de cada campo.

Algoritmos y Estructuras de Datos

Examen Final 19/4/2023

b) Definí e implementá una función que cargue en una lista enlazada de `NodoTurno` toda la información disponible en el archivo binario `info.bin`. Se sabe que dicha información se encuentra estructurada según la definición de `Turno`.

c) En `main()`: Haciendo uso de las funciones definidas en los ítems a) y b), diseña el código requerido para obtener el subconjunto de, a lo sumo, los 15 primeros elementos de tipo `Turno` almacenados en el archivo `info.bin` que corresponden a urgencias. Estos elementos deben almacenarse en una estructura de datos estática, junto a toda la información que consideres necesaria. La definición de esta estructura debe presentarse junto con la solución.

d) Definí e implementá una función `guardar()` que reciba como parámetro el subconjunto resultante del proceso indicado en el ítem c) y guarde su contenido en un archivo binario llamado `result.bin`. La función devuelve `true` si se ha podido almacenar todo el contenido con éxito (`false` en caso contrario).

e) Implementá la función recursiva `toString()` que tomando como parámetro el subconjunto resultante del proceso indicado en el ítem c), devuelva un `string` que contiene el orden de llegada de los turnos concatenados en sentido inverso a como aparecen en el subconjunto original.

Importante: Para la resolución del problema podés codificar todas las funciones que consideres necesarias. Los campos de las estructuras de datos deben respetar lo enunciado en la consigna. En los casos donde no se indica un prototipo explícito en la consigna, los parámetros formales de funciones (cantidad y tipo) deben definirse según los objetivos propuestos. El puntaje final obtenido tendrá en cuenta la eficiencia de la estrategia de resolución elegida.