Tarea 2
Algoritmos y Estructuras de Datos I

Indicaciones:

- La tarea se realizará por equipos de 1 a 3 personas, la calificación será más estricta dependiendo del número de integrantes en el equipo.

- Como entregables se entregará un archivo por ejercicio para los ejercicios 3 - 6 , éstos archivos deberán contener el código fuente de la implementación. Además se deberá entregar un documento PDF en el cual se describa brevemente la solución dada a cada unos de los problemas, y un ejemplo de la ejecución. Dentro del PDF no se debe de poner el código de la implementación ya que este estará en los archivos adjuntos.

- Los archivos se empaquetarán en un archivo zip y se enviarán por correo. El asunto del correo será [Tarea 02] (ojo poner los corchetes). En el contenido del correo se deberán escribir los nombres de los integrantes del equipo.

- Por ningún motivo se recibirán tareas de forma extemporánea a la fecha límite. Evite esperar hasta el último momento para enviar la tarea y así evitar contratiempos.

Ejercicios:

1.- (1 punto)
Describa brevemente las estructuras de datos Pila, Cola, Lista Ligada, y Lista doblemente ligada. Mencione similitudes o diferencias entre estas si es que existen, usos, funciones principales y tiempos de ejecución de estas.

2.- (1 puntos)
Explique como implementar dos Stacks usando un solo arreglo $A[1 \dots N]$ de tal manera que ambos stacks puedan seguir ingresando elementos hasta llegar a su limite sin interferirse. El tamaño en conjunto de ambos stacks debe de ser N y ambos pueden ser o no de tamaños distintos. En la implementación las operaciones de `push` y `pop` deben de ejecutarse en tiempo de $O(1)$.

3.- (2 puntos)
Sean A y B dos stacks los cuales únicamente implementan las funciones `push` y `pop`. Haciendo uso unicamente de A y B implemente una estructura cola con sus dos funciones elementales `enqueue` y `dequeue`. Analice el tiempo de ejecución de ambas operaciones.
Nota: Para la implementación de este problema se puede hacer uso de la implementación `stack` del repositorio de código del curso. La implementación de la función se deberá de probar con una función `main` donde se aprecie el funcionamiento de las funciones implementadas.

4.- (2 puntos)
Extiende la funcionalidad de la implementación de la librería `double_linked_list` contenida en el repositorio de código de la clase implementando las siguientes funciones:

4.1 Elimina el último elemento de la lista y regresa el dato contenido en el nodo
`int remove_last(Linked_List *list);`

4.2 Elimina el elemento en la posición index de la lista y regresa el dato contenido en el nodo. Si no existe el nodo regresa NULL
`int remove(Linked_List *list, int index);`

4.3 Regresa el dato contenido en la cabeza de la lista
`int at_head(Linked_List *list);`

4.4 Regresa el dato contenido en la cola de la lista
`int at_tail(Linked_List *list);`

4.5 Regresa el dato contenido en la posición index de la lista, Si no existe el nodo regresa NULL
`int at(Linked_List *list, int index);`

5.- (2 puntos)
Implementar un directorio de estudiantes con los siguientes datos para los estudiantes: Nombre, No. de cuenta y materia isncrita dados como la clave de materia. El directorio deberá contar con las siguientes funciones:

Agregar un registro.

Eliminar un registro.

Buscar un registro por numero de cuenta

Nota: Para guardar los datos de cada estudiante se deberá de crear una estructura. Para guardar cada registro se deberá de implementar un lista ligada con las funciones necesarias. Para probar el programa se deberá ejecutar un método main donde se despliegue un menú con las opciones antes mencionadas.

6.- (2 puntos)
Implementar un programa que lea el contenido de un directorio y al presionar la tecla "F" se vaya mostrando el nombre de los diferentes archivos contenidos en el directorio de forma ascendente, y si se presiona la tecla "B" se muestre el nombre de los archivos contenidos de manera descendente. Nota: revisar la documentación de la clase `std::list` en C++ la cual implementa una lista ligada. Si lo desea puede hacer uso de esta clase para realizar la implementación del ejercicio.