

## Guía de ejercicios SDA #1 y #2

### Listas simplemente enlazadas

**Generalidades:** En cada ejercicio, donde deba realizarse una comparación de elementos de tipo **struct** deberá usarse una sobrecarga del operador correspondiente.

1. Realice los siguientes programas utilizando **typedef** para la información contenida dentro de los nodos, además de funciones **<template>**
  - a. Desarrollar una aplicación que:
    - i. Genere números aleatorios comprendidos entre uno y mil quinientos y que los vaya guardando de menor a mayor en una lista enlazada.
    - ii. Crear una segunda lista enlazada y transferir a esta todos los valores que contengan números menores que 100.
  - b. Hacer un programa que:
    - i. Genere y muestre 100 números aleatorios cualquiera y después los muestre en forma inversa.
2. Realice un programa con las siguientes restricciones:
  - a. Únicamente puede hacer uso de funciones lambda
  - b. Las funciones lambda a utilizar tienen que encontrarse almacenadas en una estructura llamada **storedlambdas**
  - c. El programa deberá tener dos listas *m* y *n*, en la cual se ingresen *m* y *n* números reales respectivamente, *m* y *n* serán elegibles (entrados por consola).
  - d. Deberá existir una tercera lista enlazada en la cual se almacenarán **todos** los productos posibles de un término de la lista *m* por otro de la lista *n*.
3. Escribir un programa que devuelva cierto carácter (#0) si la lista está vacía.
4. Escribir un programa entero que devuelva el número de nodos de una lista enlazada.
5. Desarrollar un programa que devuelva la multiplicación de los números introducidos en los nodos de la lista.
6. Escribir un programa que devuelva los números pares e impares que se encuentran en los nodos de una lista.
7. Desarrollar un programa que muestre por pantalla la posición de un nodo específico (cuyo contenido es pedido por teclado).
8. Se tiene una lista, el campo dato es un registro (estructura, usar **typedef**) con la información de un alumno: nombre, edad, sexo. Escribir una función para transformar la lista de tal forma que si el primer nodo es de un alumno de sexo masculino el siguiente sea de sexo femenino.



9. Escribir un programa para formar una lista que realicen las siguientes tareas:
  - a. Crear una lista enlazadas de números enteros positivos al azar, la inserción se realiza por el último nodo.
  - b. Recorrer la lista para mostrar los elementos por pantalla.
  - c. Mostrar todos los nodos que superen un valor determinado.
10. Desarrollar un programa que intercambie dos elementos de una lista enlazada (según sus valores).
11. Desarrollar un programa que dada una lista enlazada ordene los nodos en forma ascendente.
12. Escribir un programa que permita eliminar el penúltimo nodo de una lista enlazada.
13. Implementar un procedimiento para insertar un dato en orden ascendente en una lista enlazada. Es decir, el nodo que representa el dato debe ser insertado en una posición tal que al recorrer la lista los nodos se recorran de menor a mayor respecto del dato.
  - a. Ahora hágalo en orden descendente.
14. Implemente un procedimiento que inserte un dato de modo similar al insertar básico, al final de la lista. Pero ahora, no se debe permitir insertar datos repetidos, si un dato ya está almacenado entonces la lista no varía.
15. Implemente una función que dada una lista enlazada `l`, devuelva otra lista `l1` con solamente los nodos que almacenan un dato impar.
16. Implemente una función que dada una lista enlazada `l`, devuelva otra lista `l1` con solamente los nodos que almacenan un dato mayor que el promedio.



## Pilas y colas

1. Escribir una función `replace` que tenga como argumentos una pila con tipo de elemento `int` y dos valores `int`: nuevo y viejo de forma que, si el segundo valor aparece en algún lugar de la pila, sea reemplazado por el segundo.
2. Escribir una función que invierta el contenido de una pila.
3. Escribir una función, que, dado un número entero, guarde sus dígitos en una pila.
4. Escribir un programa que le pida enteros al usuario, guarde los números pares en una pila, y los impares en otra pila. Luego, debe determinar si la cantidad de pares es igual a la cantidad de impares.
5. Usando una pila, solicite al usuario una frase, y luego escriba la frase en pantalla con las palabras en orden inverso.
6. Escribir un programa que invierta el contenido de una cola.
7. Asumamos que, en un banco, las personas solo pueden hacer una operación. Los tiempos que cada tipo de operación se tarda son los siguientes:
  - a. Retiro 4 min
  - b. Depósito 2 min
  - c. Consulta 3.5 min
  - d. Actualización 5 min
  - e. Pagos 2 minLlene una cola con varias personas que necesitan hacer operaciones en el banco, por cada persona se debe almacenar su nombre, y que tipo de operación va a realizar (usando el campo dato como un `typedef`), y cuántas operaciones de ese tipo. Su programa, debe devolver el tiempo promedio de atención a clientes del banco, usando de datos esta cola.
8. Unos vehículos blindados intentan pasar por un puente defectuoso. Para ello, forman una cola para atravesarlo y la probabilidad de éxito al momento de cruzar el puente es de 0.9 al inicio. Cada vez que un vehículo entra el puente, éste se deteriora más y la probabilidad de éxito se reduce en 0.06. Para un total de  $n$  vehículos blindados, ¿cuántos logran atravesar el puente? ¿Cuántos se murieron?