

Parcial

Parcial 02/11

Consideraciones:

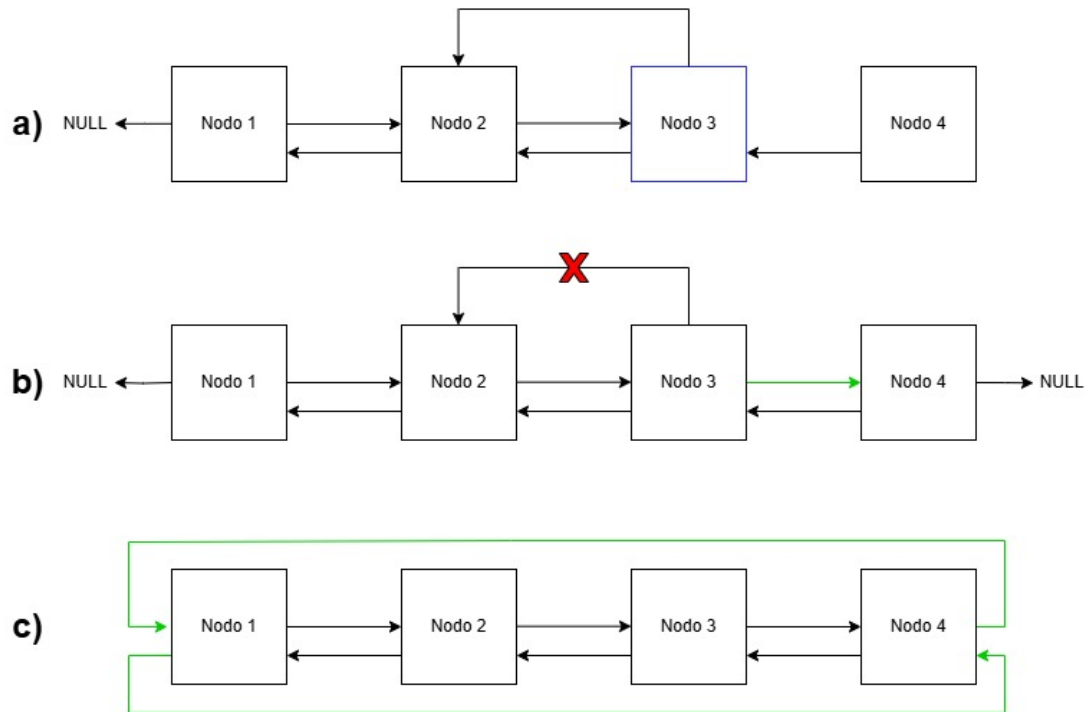
- Los numeros de todas las listas son positivos.
- Ninguna lista contiene números duplicados.
- Ninguna lista contiene más de 1000 elementos.
- No necesita usar **NADA** del archivo Catedra_1.c

Ejercicio 1 (3 puntos):

1. Lea el archivo `archivo.bin` que contiene m números enteros, luego:
 - a. Imprima la lista (0.5 ptos)
 - b. Calcule el promedio. (0.5 ptos)
 - c. Modifique la lista tal que todos los numeros, cuyo proximo valor es impar sean multiplicados por 5. Imprima la lista modificada. (1 pto)
 - d. Encuentre el primer numero que sea mayor AL DOBLE DEL PROMEDIO calculado en el punto B y cuyo tercer bit menos significativo sea un 0. Imprima este numero. 1 (pto)

Ejercicio 2 (3 puntos):

2. Ejecute la funcion `CATEDRA_CrearListaDoble` , que contiene m nodos y un ciclo. Luego:
 - a. Busque el ciclo e imprima tanto el valor del nodo que lo contiene (el nodo azul de la figura) como los valores de sus nodos adyacentes (deberian tener el mismo valor). (1 pto)
 - b. Elimine el ciclo. (1 pto)
 - c. Convierta la lista en una circular (conectando el principio y el final). Imprima la lista una única vez. (1pto)



Ejercicio 3 (4 puntos):

3. Utilizando el vector provisto, `lista_corta` :

- a. Cree una lista simplemente enlazada a partir de los números de este vector. Respetando el orden, de manera tal que `lista_corta[0]` sea el primer nodo de la lista. Imprima la lista enlazada. (1 pto)

`lista_corta[0] → lista_corta[1] → lista_corta[2] → ...`

b. Elija un grupo de consignas:

i. Grupo 1

1. Escriba una función para una lista enlazada, que encuentre el nodo que cumpla las siguientes condiciones:
 - a. Su valor sea menor que X.
 - b. De entre todos los nodos que cumplen la condición anterior, seleccionar el nodo con el valor más grande.

Una vez identificado el nodo, la función debe eliminarlo de la lista, liberar la memoria asociada a ese nodo y devolver el valor que contenía.

Se puede dividir el código en dos sub-funciones.

A continuación se deja un ejemplo de prototipo que pueden usar. (1.5 ptos)

```
int extraerMax(Node *primerNodo, int last_max);
```

2. Ordene la lista de menor a mayor. Imprima la lista ordenada. (Recomendamos no utilizar Bubble Sort) (1.5 ptos)

ii. Grupo 2

1. Cree la funcion intercambiar nodos.
Esta debe intercambiar las pociones de dos nodos en una lista. (1.5 ptos)

antes: 1 → 3 → x → 6 → 8 → y → 2

despues: 1 → 3 → y → 6 → 8 → x → 2

2. Utilice la función creada en el punto anterior para ordenar la lista enlazada. Puede utilizar los algorírtmos de ordenamiento vistos en clase. (1.5 ptos)