

Apellido y nombre: _____ Legajo: _____ Curso con Prof: _____

Cantidad de hojas entregadas: _____ Nota: _____ Evaluó Prof: _____

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

Simulación de Funcionamiento de ascensores

Temas evaluados: Resolución de problemas, estructuras de datos, listas doblemente enlazadas, y lenguaje de programación.

Contexto

Usted es parte de un equipo que desarrolla aplicaciones para *Edificios Inteligentes*, y es el responsable de simular el uso de los ascensores.

Descripción

El movimiento de los ascensores se simula con la ayuda de una estructura de datos. La estructura de datos es una lista doblemente enlazada que va desde el subsuelo más bajo, representado con valores enteros negativos, pasando por la planta baja, representada por el valor cero hasta los pisos superiores representados por valores enteros positivos. Cada nodo, que representa un piso, contiene el numero de piso, la cantidad de personas que suben y que bajan en cada uno y enlaces a los pisos contiguos en ambas direcciones. La lista se recorre con el puntero **elPisoActual**, que apunta a uno de los pisos intermedios pero no a los pisos extremos.

Problema

Desarrollar la función **InsertarPisoArribaDelActualYMedir**, que, dado los punteros **unNuevoPiso** y **unPisoActual**, inserta **unNuevoPiso** arriba de **unPisoActual**, y calcula cuántos pisos separan a **unNuevoPiso** del piso más bajo, del más alto, y de la planta baja.

Asunciones (Precondiciones)

Asuma que, para la función **InsertarPisoArribaDelActualYMedir**, las siguientes condiciones son verdaderas (precondiciones):

- El puntero **unNuevoPiso** apunta a un nodo con sus tres datos correctamente inicializados, y con los enlaces arriba y abajo, ambos, con valor nulo.
- El puntero **unPisoActual** apunta a un piso intermedio y no la planta baja ni los extremos superior ni inferior.

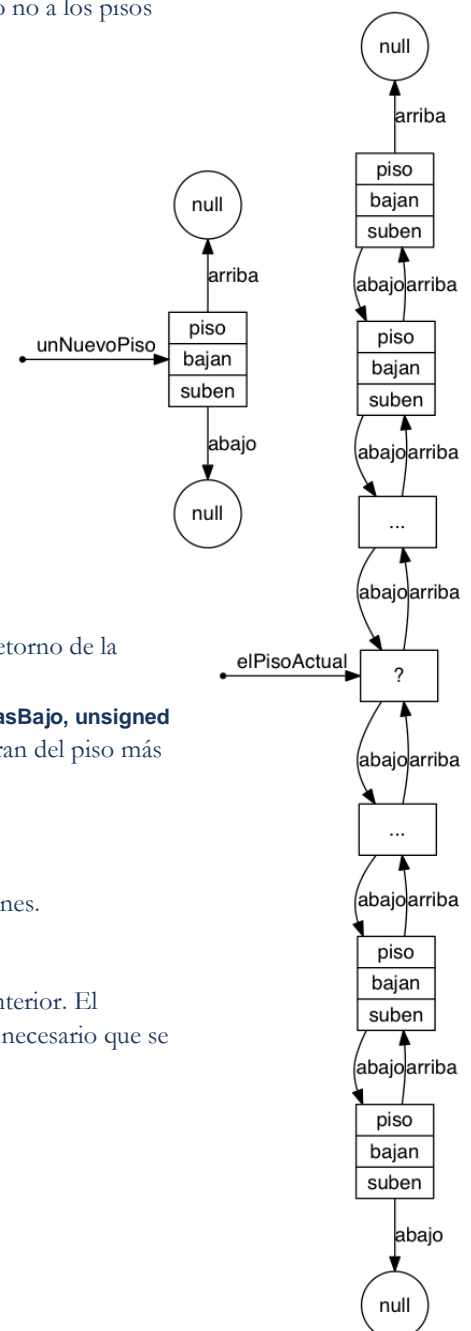
Restricciones

La función **InsertarPisoArribaDelActualYMedir** debe respetar las siguientes restricciones:

- La cantidad de pisos que separan a **unNuevoPiso** del más alto y del más bajo debe ser devuelto por parámetros de salida (*out*).
- La cantidad de pisos que separan a **unNuevoPiso** de la planta baja debe ser el valor de retorno de la función.
- Debe invocar a **void MedirDistanciaAPisosExtremos(const Nodo *unPiso, unsigned &masBajo, unsigned &masAlto)**, que dado **unPiso**, devuelve en **masBajo** y en **masAlto**, cuántos pisos lo separan del piso más bajo y cuántos del más alto, respectivamente.

Se pide

1. Codificar todas las **declaraciones** necesarias para la estructura del diagrama.
2. Codificar el prototipo de **InsertarPisoArribaDelActualYMedir**. Justificar sus decisiones.
3. Codificar o diagramar **InsertarPisoArribaDelActualYMedir**
4. Codificar o diagramar **MedirDistanciaAPisosExtremos**
5. Codificar o diagramar un fragmento de la función **main** que la invoque función anterior. El fragmento debe incluir las declaraciones de las variables que se utilice, pero no es necesario que se las inicialice.



Apellido y nombre: _____ Legajo: _____ Curso con Prof: _____

Cantidad de hojas entregadas: _____ Nota: _____ Evaluó Prof: _____

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

SOLUCION 3 de diciembre de 2014

```
#include <iostream>
using namespace std;
```

1 Codificar todas las declaraciones

```
struct InfoPiso {
    signed piso;
    unsigned bajan;
    unsigned suben;
};

struct Nodo {
    InfoPiso info;    //la información es el piso y cuantos bajan y suben
    Nodo* abajo;     //puntero al piso anterior
    Nodo* arriba;    //puntero al piso siguiente
};
```

2 Codificar el prototipo de ...

unsigned **insertarPisoArribaDelActualYMedir**(Nodo*, Nodo*, unsigned&, unsigned&)
 Valor de retorno es la distancia a la planta baja, un entero positivo
 Nodo* unNuevoPiso y Nodo* unPisoActual los punteros a los pisos dato
 unsigned& para los enteros positivos de distancias a los extremos

3 Codificar la función...

```
unsigned insertarPisoArribaDelActualYMedir(
    Nodo* unNuevoPiso, Nodo* unPisoActual, unsigned& masBajo, unsigned& masAlto) {

    insertarPisoArribaDelActual(unNuevoPiso,unPisoActual);
    //Uso unNuevoPiso para medir distancias
    medirDistanciaAPisosExtremos(unNuevoPiso,masBajo,masAlto);

    //Cuantos pisos separa unNuevoPiso de la planta baja
    unsigned distPB = 0;
    Nodo* ptr = unNuevoPiso;
    while (ptr != NULL && ptr->infoPiso.piso != 0) {
        distPB++;
        ptr = (unNuevoPiso->infoPiso.piso > 0) ? ptr->abajo : ptr->
arriba;
    }

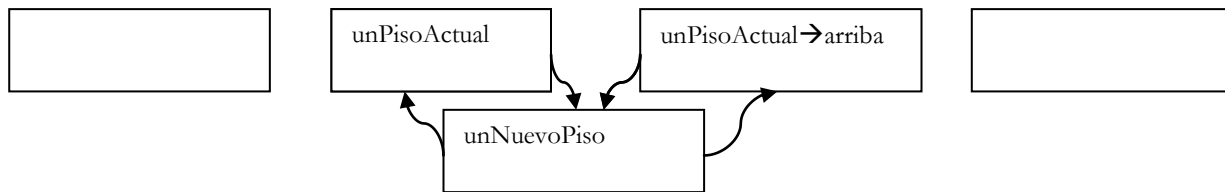
    return distPB;
}
```

Apellido y nombre: _____ Legajo: _____ Cursó con Prof: _____

Cantidad de hojas entregadas: _____ Nota: _____ Evaluó Prof: _____

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

```
void insertarPisoArribaDelActual (Nodo* unNuevoPiso, Nodo* unPisoActual) {
    unNuevoPiso->abajo = unPisoActual;
    unNuevoPiso->arriba = unPisoActual->arriba;
    unPisoActual->arriba->abajo = unNuevoPiso;
    unPisoActual->arriba = unNuevoPiso;
}
```



4 Codificar la función...

```
void medirDistanciaAPisosExtremos (const Nodo* unPiso, unsigned& masBajo,
unsigned& masAlto) {
    masBajo = 0;
    masAlto = 0;

    Nodo* ptrAbajo = unPiso->abajo;
    while (ptrAbajo != NULL) {
        masBajo++;
        ptrAbajo = ptrAbajo->abajo;
    }

    Nodo* ptrArriba = unPiso->arriba;
    while (ptrArriba != NULL) {
        masAlto++;
        ptrArriba = ptrArriba->arriba;
    }
}
```