

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.

Simulación de Funcionamiento de ascensores

Temas evaluados: Resolución de problemas, estructuras de datos, listas doblemente enlazadas, flujo binario 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 número de piso, la cantidad de personas que suben y que bajan en cada uno y enlaces a los pisos contiguos en ambas direcciones, implementado con punteros empotrados en los nodos. La lista se recorre con el puntero externo **elPisoActual**, que apunta a uno de los pisos intermedios pero no a los pisos extremos.

Problema

Desarrollar la función **EliminarLosPisosPrimosArribaDe**, que, dado el puntero externo **unPisoActual**, elimina todos los pisos por encima de **unPisoActual** cuyo número de piso sea primo, y, por cada piso que elimine, escriba en el flujo (*archivo abierto*) **unFlujoDeEscritura** el número de piso y la diferencia entre los que suben y bajan; la función debe retornar la cantidad de pisos eliminados.

Asunciones (Precondiciones)

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

- El puntero **unPisoActual** apunta a un nodo válido.
- El flujo **unFlujoDeEscritura** permite la escritura.
- Se desarrollará una función que determina si un número es primo o no.

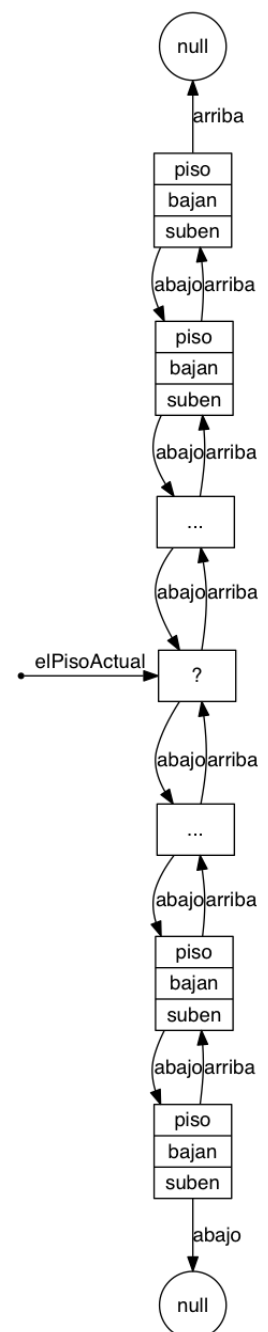
Restricciones

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

- Debe invocar a **void EliminarPiso(Nodo& *unPiso)**, que dado **unPiso** no nulo lo elimina de la estructura.
- Debe invocar a la función que determina si un número es primo o no.

Se pide que

1. Codifique todas las **declaraciones** necesarias para la variable **unPisoActual** del diagrama, y la estructura del flujo binario. Respete los identificadores (*nombres*).
2. Codifique el prototipo de **EliminarLosPisosPrimosArribaDe**. Justifique sus decisiones.
3. Codifique o diagrame una función que determina si un número es primo.
4. Codifique o diagrame **EliminarPiso**.
5. Codifique o diagrame **EliminarLosPisosPrimosArribaDe**.



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.

```
#include <iostream>
```

```
using namespace std;
```

```
#include <stdio.h>
```

1. Codifique las declaraciones ...

```
struct InfoPiso {           //la información del nodo
    signed piso;
    unsigned bajan;
    unsigned suben;
};
```

```
struct Nodo {               // el nodo con la información y los 2 punteros empotrados
    InfoPiso infoPiso;
    Nodo* abajo;
    Nodo* arriba;
};
```

```
struct TipoPisoElimiado { //la estructura del flujo binario
    signed nro;
    signed diffSubeBaja;
};
```

**2. Codifique el prototipo de EliminarLosPisosPrimosArribaDe
unsigned EliminarLosPisosPrimosArribaDe(Nodo* , FILE*)**

3. Codifique o diagrame una función que determina si un número es primo.

```
bool EsPrimo (int numero) {

    for (int i = 2; i < numero ; i++) {
        if(numero % i == 0) return false;
    }
    return true;
}
```

4. Codifique o diagrame EliminarPiso.

```
void EliminarPiso(Nodo* & unPiso) {
    if (unPiso->arriba != NULL) {
        unPiso->arriba->abajo = unPiso->abajo;
    }
    if (unPiso->abajo != NULL) {
        unPiso->abajo->arriba = unPiso->arriba;
    }
    delete(unPiso);
}
```

unPiso → arriba

unPiso

unPiso → abajo

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.
5. Codifique a diagraame

unsigned EliminarLosPisosPrimosArribaDe(Nodo* unPisoActual, FILE* unFlujoDeEscritura) {

```

    unsigned cantPisosEliminados = 0;
    Nodo* unPisoAbajo = unPisoActual;
    unPisoActual = unPisoActual->arriba; // para "saltar" el actual
    while( unPisoActual != NULL ) {
        unPisoAbajo = unPisoActual;           // se lo retiene para procesarlo si corresponde
        unPisoActual = unPisoActual->arriba;   // Subir al piso próximo
        if(EsPrimo(abs(unPisoAbajo->piso))) {    // en caso de procesar al piso de abajo
            GuardarPisoEliminado(unFlujoDeEscritura, unPisoAbajo);
            EliminarPiso(unPisoAbajo);
            ++cantPisosEliminados;
        }
    }

    return cantPisosEliminados;
}
```

void GuardarPisoEliminado(FILE* unFlujoDeEscritura, Nodo *unPiso) {

```

    TipoPisoElimiado tpe;
    tpe.nro = unPiso->InfoPiso.piso;
    tpe.diffSubeBaja = unPiso->infoPiso.suben - unPiso->infoPiso.bajan; //Puede dar un nro negativo

    fwrite(&tpe,sizeof(tpe),1,unFlujoDeEscritura);

}
```