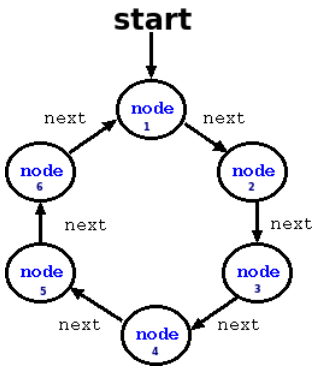
Tercer Exámen Parcial Estructuras de Datos

Prof. Luis Garreta
Pontificia Universidad Javeriana - Cali

1. Lista Circular con Nodos

La lista circular con nodos a diferencia de la que se ha estudiado en el curso, no tiene ni apuntadores a la cabeza (head) ni a la cola (tail) sino un único apuntador llamado start (inicio) que apunta al inicio de la lista, como se muestra en la siguiente figura



La estructura del nodo sigue siendo la misma que el nodo de listas: un campo *data* para los valores y un campo *next* que apunta al siguiente nodo. La implementación de esta lista es:

```
class ListaCircular {
  private:
    Nodo * start;
  public:
    ListaCircular ();
    int longitud ();

};
```

De acuerdo a esta estructura, implemente la función *longitud ()* que retorna el número de elementos de la lista.

2. Tablas Hash

Suponga una tabla hash con solución de colisiones por encadenamiento que está completamente llena, es decir el vector de 100 posiciones está lleno y las listas con capacidad de 50 elementos también están llenas.

1. Cúal sería el costo en número de accesos (numero de pasos) para buscar un valor que NO existe en la tabla hash?

2. Que solución tomaría usted para "agrandar" esta tabla hash para que tenga espacio para nuevos valores y como lo realizaría (describalo en un parrafo corto).

3. Archivos

Dado el siguiente archivo tipo texto con palabras y sus significados, realice una función en C++ que transforme este archivo de 800 palabras en un archivo de índices tipo texto donde el índice viene dado por la posición dentro del archivo (inicia en 0) y la palabra viene dada por la primera palabra que está en el archivo.

"significados.txt"

```
"airplane" "avión, aeronave, aeroplano"
"blue" "azular, añilar, dar azulete a, empavonar"
"car" "carro, auto, automóvil, coche; vagón"
"red" "rojo, colorado, de color rojo, de rojo"
...
...
"very" "muy, asaz, bien"
"yes" "sí, visto bueno, voto afirmativo"
"zombie" "zombie, cadáver resucitado"
```

"indices.txt"

```
0 airplane
1 blue
2 car
3 red
...
...
797 very
798 yes
799 yes
```

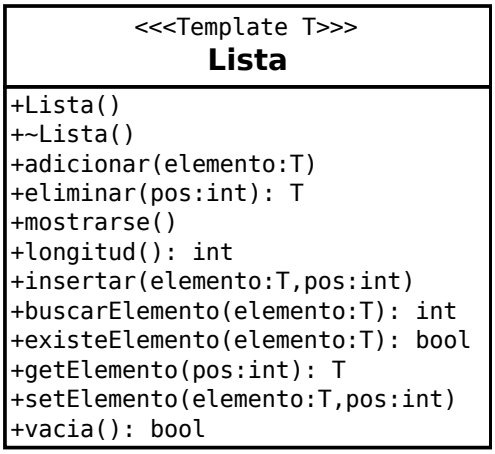
4. Batalla Naval

En el juego batalla naval usted tiene tres listas: una para barcos, otra para submarinos y otra para portaviones, implemente una función que ingresen como parámetros estas tres listas, construya una nueva lista de naves y concatene las naves de cada lista de entrada en una nueva lista de naves que las contiene a todas las demás (barcos, submarinos, y portaaviones), la cual se retorna al final.

El prototipo de la función es el siguiente:

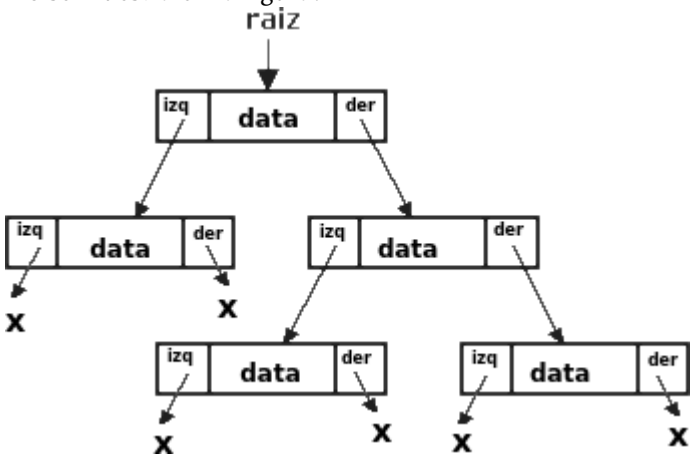
```
Lista <Nave> unirListas (Lista <Barco> lb,
                         Lista <Submarino> ls,
                         Lista <Portaaviones> lp );
```

La implementación de las listas usted no la conoce (puede ser por nodos, o STL, o arreglos) solo conoce la interfaz pública con los siguientes métodos que los puede usar para implementar la función:



5. Definición de Estructuras de Datos

Una estructura de una arbol binario consta de un nodo raiz que apunta hacia dos nodos llamados izquierdo y derecho, como se muestra en la figura:



Cada nodo tiene la misma estructura: un valor y dos apuntadores: uno al izquierdo y otro al derecho.

1. Defina la estructura para este tipo de nodo
2. Defina la clase que represente el árbol binario.
3. Implemente el constructor del árbol binario:
   arbolBinario ();
4. BONUS: implemente la función de adicionar valor al hijo izquierdo:
   adicionarIzquierdo (int valor);