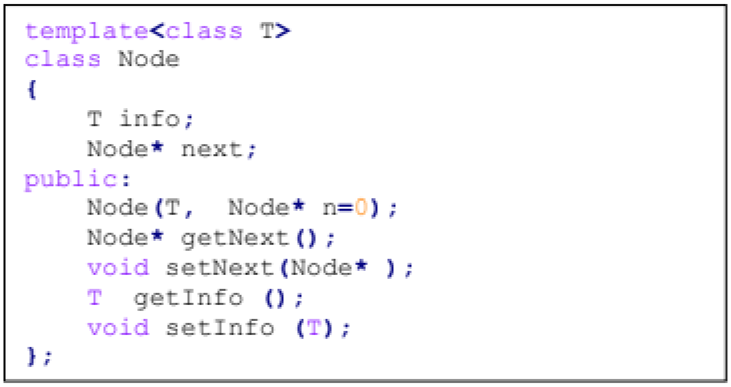
Data Structures - Summer 2023
Assignment #2

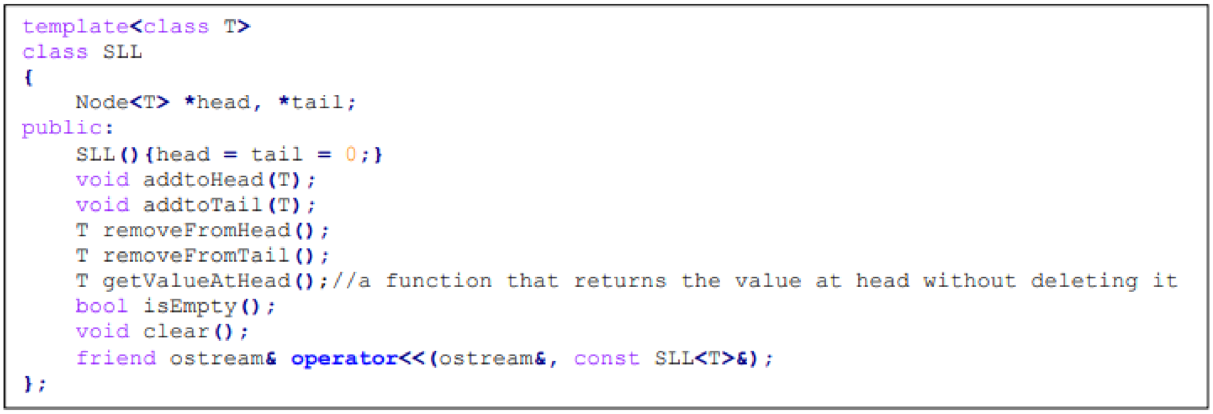
Linked List Problems.

1. Implement a class template Node that has the following declaration: ***1 Point***



```
template<class T>
class Node
{
    T info;
    Node* next;
public:
    Node(T,  Node* n=0);
    Node* getNext();
    void setNext(Node* );
    T  getInfo ();
    void setInfo (T);
};
```

2. Create a class template SLL, which is a single linked list with the following declaration: ***1 Point***



```
template<class T>
class SLL
{
    Node<T> *head, *tail;
public:
    SLL(){head = tail = 0;}
    void addtoHead(T);
    void addtoTail(T);
    T removeFromHead();
    T removeFromTail();
    T getValueAtHead();//a function that returns the value at head without deleting it
    bool isEmpty();
    void clear();
    friend ostream& operator<<(ostream&, const SLL<T>&);
};
```

3. Pairwise Swap Nodes of a given Linked List. ***1 Point***
   Given a singly linked list, write a function to swap elements pairwise.

Example:

Input: *1->2->3->4->5->6->NULL*
Output: *2->1->4->3->6->5->NULL*

Input: *1->2->3->4->5->NULL*
Output: *2->1->4->3->5->NULL*

Input: *1->NULL*
Output: *1->NULL*

4. Remove duplicate element from sorted Linked List. *1 Point*

Given a singly linked list consisting of **N** nodes. The task is to remove duplicates (nodes with duplicate values) from the given list (if exists).

***Example*:**

Input: *2->2->4->5*

Output: *2 4 5*

Stack Problems.

1. Implement a class template Stack: *1 Point*

Implement a stack with **push()**, **pop()**, **top()**, and **empty()** operations

2. Reverse a Stack using Recursion *1 Point*

Implement a function to reverse a stack using recursion, without using any loop.

***Example*:**

Input: *elements present in stack from top to bottom 1 2 3 4*
Output: *4 3 2 1*

3. Delete middle element of a stack *1 Point*

The task is to delete the **middle** element of stack without using any additional data structure.

***Example*:**

Input: *Stack[] = [1, 2, 3, 4, 5]*
Output: *Stack[] = [1, 2, 4, 5]*

Queue Problems.

1. Generate Binary Numbers from 1 to N. ***1 Point***
   Write a function that generates and prints all binary integers from 1 to N

   Example:

   Input: *3*

   output: *1 10 11*

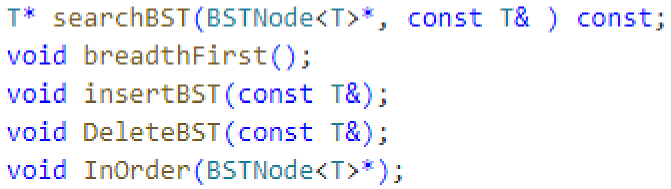
2. Sorting a Queue. ***1 Point***
   You will have a queue with random integer elements as an input, you have to sort it with any sort algorithm.

Binary Search Tree (BST) Problems.

1. Implement a class template node that has the following declaration:

```
template <class type>
class node
{
    node<type>* leftChild;
    node<type>* rightChild;
    type data;
};
```

2. Implement the following functions:



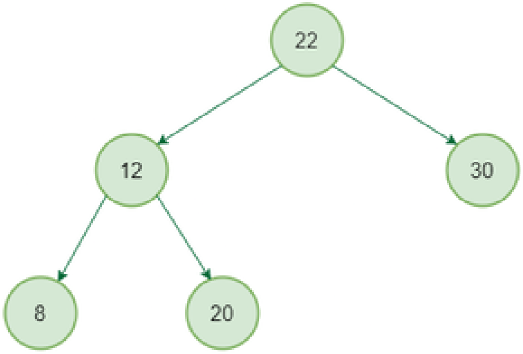
```
T* searchBST(BSTNode<T>*, const T& ) const;
void breadthFirst();
void insertBST(const T&);
void DeleteBST(const T&);
void InOrder(BSTNode<T>*);
```

3. Find the node with minimum value

Write a function to find the node with minimum value in a Binary Search Tree.

Example

Input: **Output: 8**



Submission Rules

1- Assignment is submitted in teams of 3 from any group.

2- You will upload a zipped folder that contains your code (Don’t include any .exe files in your submission).

3- Assignment submission is on Google Classroom (No submission through mail).

4- Follow this convention for naming your folder: ID1_ID2_ID3_A#_G# (i.e 20200111_20200222_20200333_A2_G5_G6)

5- Deadline of the Assignment: 21 August, 2023, at 11:59 p.m

Failure to follow any of the above rules will result in your submission being discarded and your team being considered to have not submitted