ASSIGNMENT 2

```
#include <bits/stdc++.h>
using namespace std;

class Node {
public :
int data;
Node *next;
};

class Stack {
public :
Node *top;
Stack() {
top = NULL;
}

void push(int n) {
Node *New = new Node;
New->data = n;
New->next = top;
top = New;
cout<<"Pushed "<<n<<" into the stack."<<endl;
}

void pop() {
if(isEmpty()) {
cout<<"Stack is empty. Cannot pop."<<endl;
return;
}
Node *temp = top;
cout<<"Popped "<<top->data<<" from the stack."<<endl;
top = top->next;
delete temp;
}

void peek() {
if(isEmpty()) {
cout<<"Stack is empty."<<endl;
return;
}
cout<<"Top element is "<<top->data<<"."<<endl;
}

bool isEmpty() {
if(top==NULL) return true;
return false;
```

```
}
};

class Queue {
public :
Node *front;
Node *rear;

Queue() {
front = NULL;
rear = NULL;
}

void enqueue(int n) {
Node *New = new Node;
New->data = n;
New->next = NULL;
if(isEmpty()) {
front = rear = New;
}
else {
rear->next = New;
rear = New;
}
cout<<"Enqueued "<<n<<" to the queue."<<endl;
}

void dequeue() {
if(isEmpty()) {
cout<<"Queue is empty. Cannot dequeue."<<endl;
return;
}
if(front==rear) {
Node *temp = front;
front = rear = NULL;
delete temp;
return;
}
Node *temp = front;
front = front->next;
delete temp;
}

void peek() {
if(isEmpty()) {
cout<<"Queue is empty."<<endl;
return;
}
cout<<"Front element is "<<front->data<<"."<<endl;
}
```

```
bool isEmpty() {
return front == NULL;
}
};

int main() {
Stack s;
s.push(1);
s.push(2);
s.push(3);
s.push(4);
s.push(5);
s.push(6);

s.peek();
s.pop();
s.peek();
}
```

OUTPUT :

```
pict@a3-309-06-OptiPlex-5070:~/23170$ g++ assignment2.cpp
pict@a3-309-06-OptiPlex-5070:~/23170$ ./a.out
Pushed 1 into the stack.
Pushed 2 into the stack.
Pushed 3 into the stack.
Pushed 4 into the stack.
Pushed 5 into the stack.
Pushed 6 into the stack.
Top element is 6.
Popped 6 from the stack.
Top element is 5.
pict@a3-309-06-OptiPlex-5070:~/23170$
```