Name: Sanket Wakankar

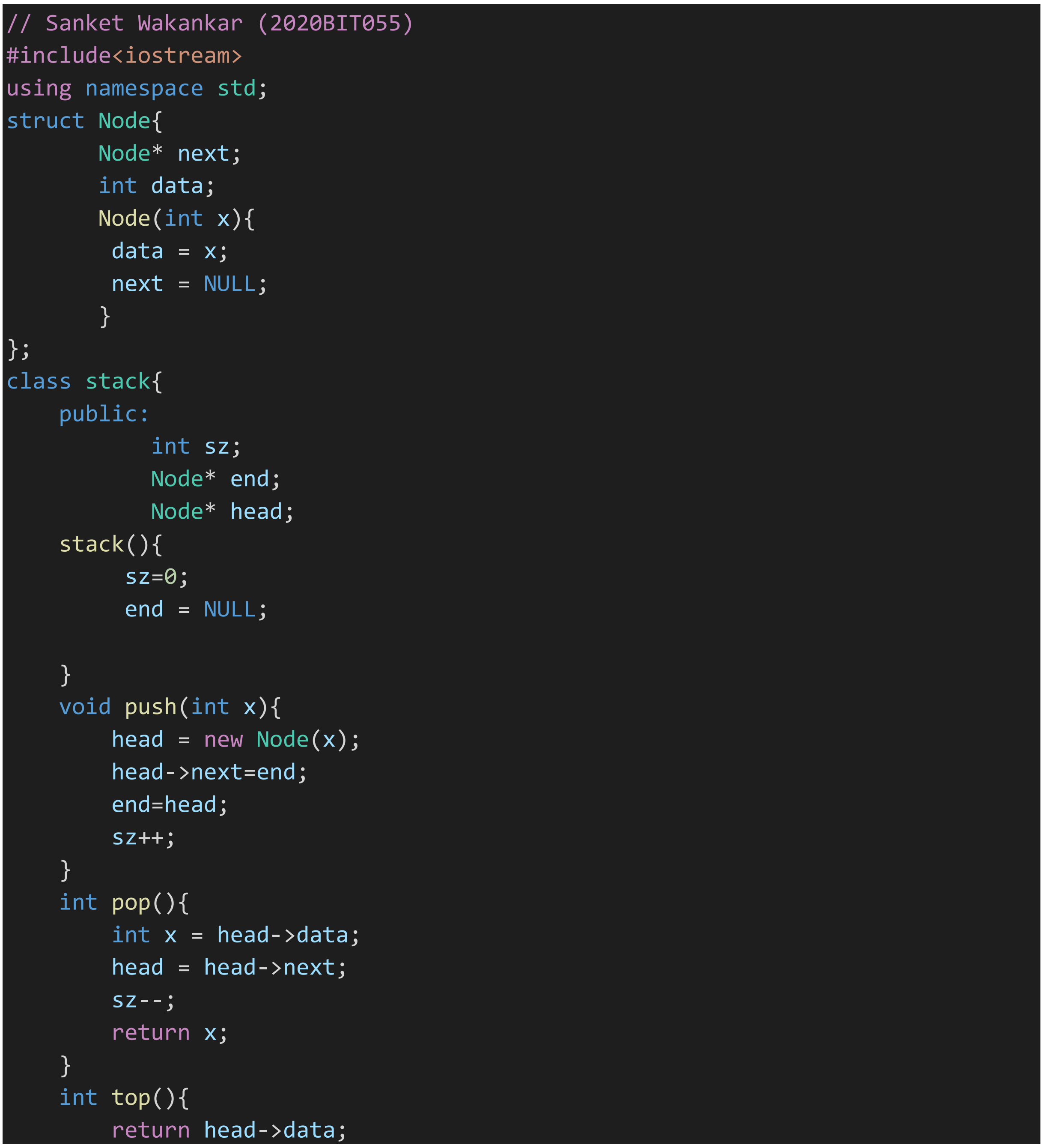
Reg. No.: 2020BIT055

Sub: DAA

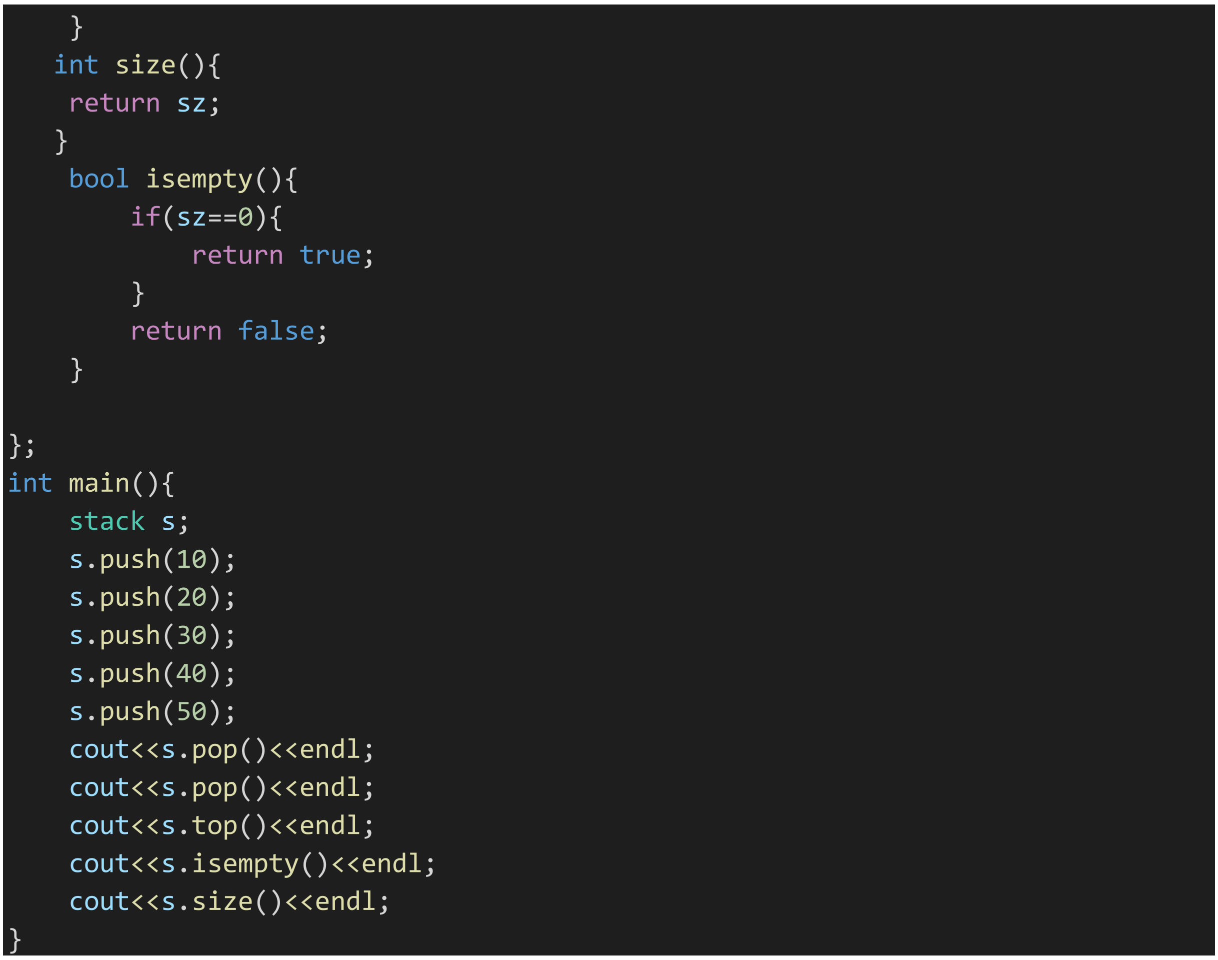
PRACTICAL NO. 2

Write C/C++ code to implement:

1.Stack Using Linked List.

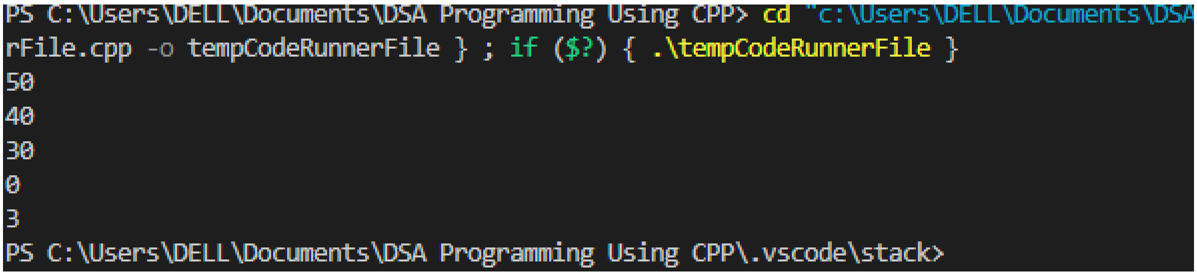


```
// Sanket Wakankar (2020BIT055)
#include<iostream>
using namespace std;
struct Node{
       Node* next;
       int data;
       Node(int x){
        data = x;
        next = NULL;
       }
};
class stack{
    public:
           int sz;
           Node* end;
           Node* head;
    stack(){
         sz=0;
         end = NULL;

    }
    void push(int x){
        head = new Node(x);
        head->next=end;
        end=head;
        sz++;
    }
    int pop(){
        int x = head->data;
        head = head->next;
        sz--;
        return x;
    }
    int top(){
        return head->data;
```



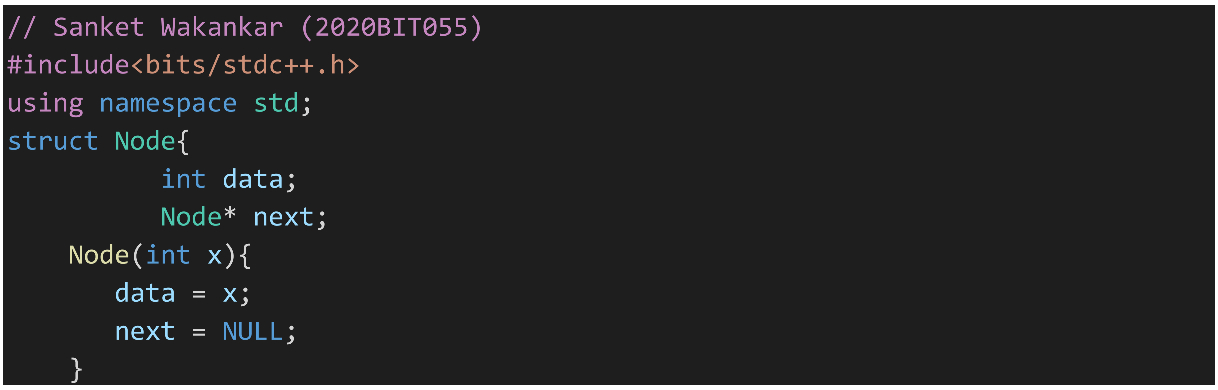
```
    }
   int size(){
    return sz;
   }
    bool isempty(){
        if(sz==0){
            return true;
        }
        return false;
    }

};
int main(){
    stack s;
    s.push(10);
    s.push(20);
    s.push(30);
    s.push(40);
    s.push(50);
    cout<<s.pop()<<endl;
    cout<<s.pop()<<endl;
    cout<<s.top()<<endl;
    cout<<s.isempty()<<endl;
    cout<<s.size()<<endl;
}
```

Output:

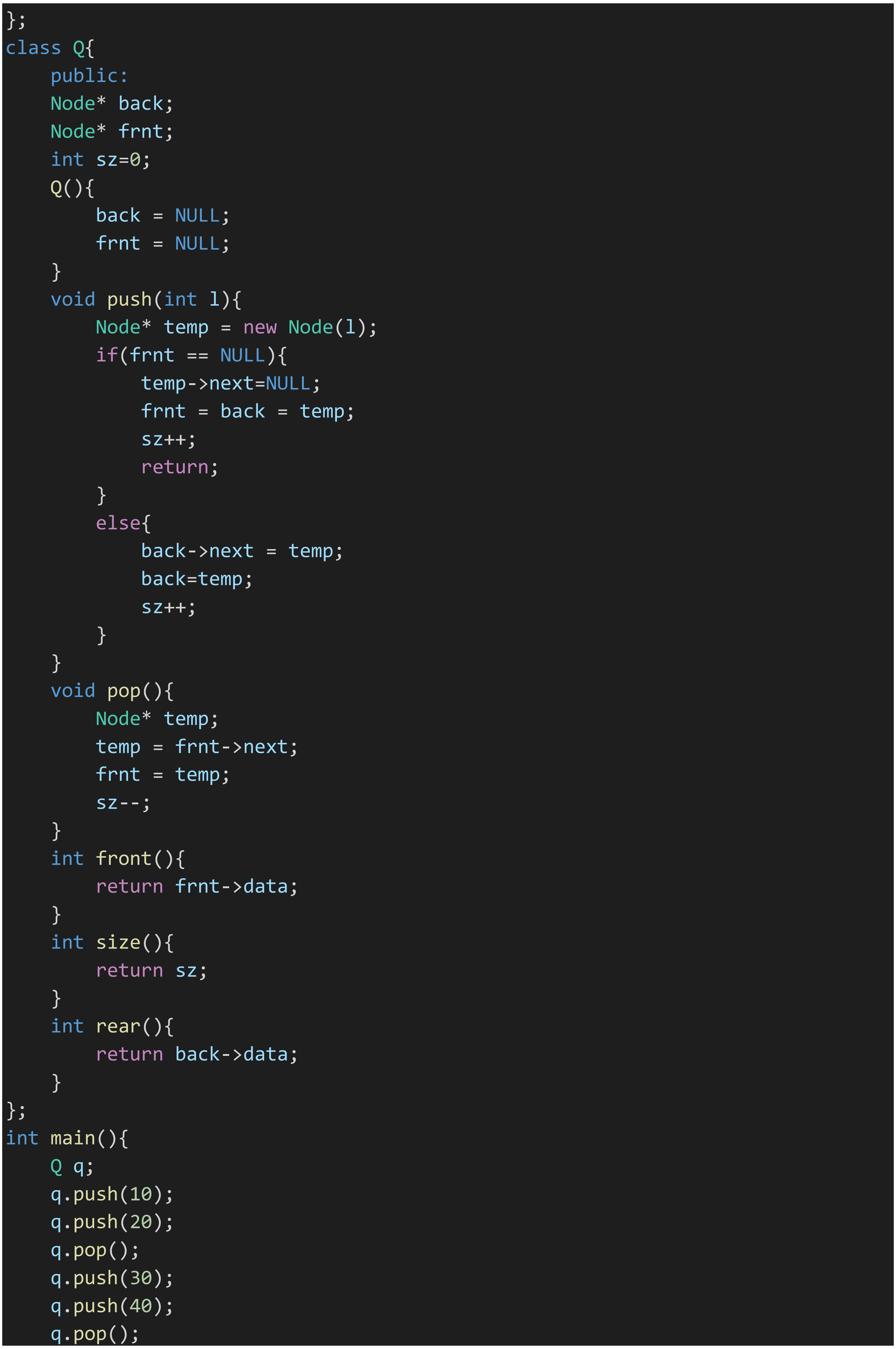


```
PS C:\Users\DELL\Documents\DSA Programming Using CPP> cd "c:\Users\DELL\Documents\DSA
rFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
50
40
30
0
3
PS C:\Users\DELL\Documents\DSA Programming Using CPP\.vscode\stack>
```

2. Queue using Linked list.

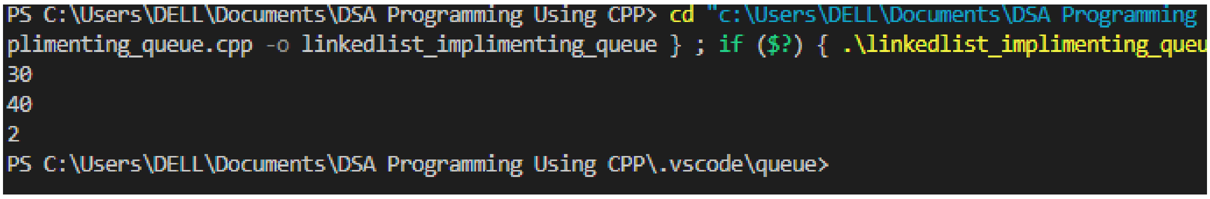


```
// Sanket Wakankar (2020BIT055)
#include<bits/stdc++.h>
using namespace std;
struct Node{
          int data;
          Node* next;
    Node(int x){
       data = x;
       next = NULL;
    }
```



```
};
class Q{
    public:
    Node* back;
    Node* frnt;
    int sz=0;
    Q(){
        back = NULL;
        frnt = NULL;
    }
    void push(int l){
        Node* temp = new Node(l);
        if(frnt == NULL){
            temp->next=NULL;
            frnt = back = temp;
            sz++;
            return;
        }
        else{
            back->next = temp;
            back=temp;
            sz++;
        }
    }
    void pop(){
        Node* temp;
        temp = frnt->next;
        frnt = temp;
        sz--;
    }
    int front(){
        return frnt->data;
    }
    int size(){
        return sz;
    }
    int rear(){
        return back->data;
    }
};
int main(){
    Q q;
    q.push(10);
    q.push(20);
    q.pop();
    q.push(30);
    q.push(40);
    q.pop();
    cout<<q.front()<< endl;
    cout<<q.rear()<<endl;
    cout<<q.size()<<endl;
}
```

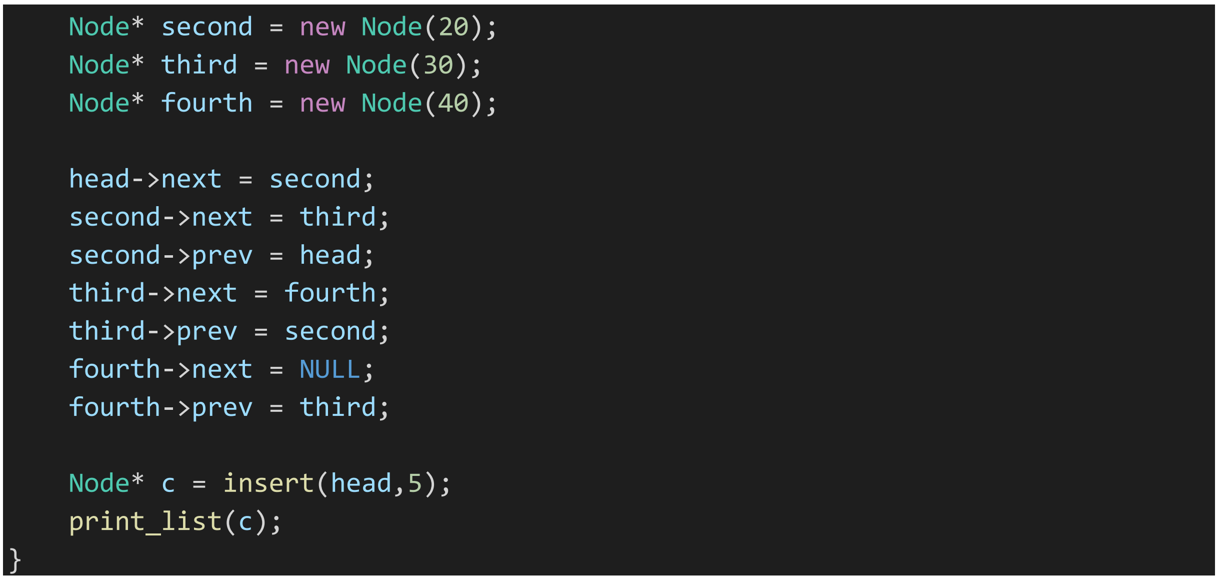
Output:



```
PS C:\Users\DELL\Documents\DSA Programming Using CPP> cd "c:\Users\DELL\Documents\DSA Programming
plimenting_queue.cpp -o linkedlist_implimenting_queue } ; if ($?) { .\linkedlist_implimenting_queu
30
40
2
PS C:\Users\DELL\Documents\DSA Programming Using CPP\.vscode\queue>
```

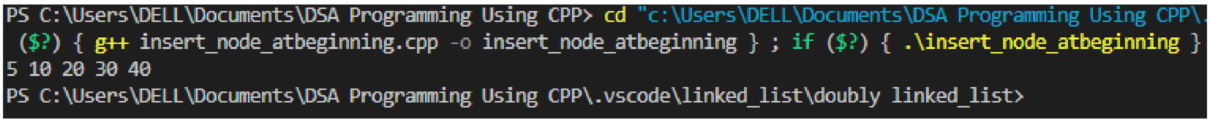
3. Doubly Linked List:

```
// Sanket Wakankar (2020BIT055)
#include<iostream>
using namespace std;
class Node{
    public: int data;
            Node* next;
            Node* prev;
    Node(int x){
        data = x;
        next = NULL;
        prev = NULL;
    }
};
Node* insert(Node* head , int element){
    Node* temp = new Node(element);
    temp->next = head;
    if(head == NULL){
        return temp;
    }
    else{
        head->prev = temp;
        return temp;
    }
}
void print_list(Node* head){
    while(head!=NULL){
        cout<<head->data<<" ";
        head = head->next;
    }
}
int main(){
    Node* head = new Node(10);
```



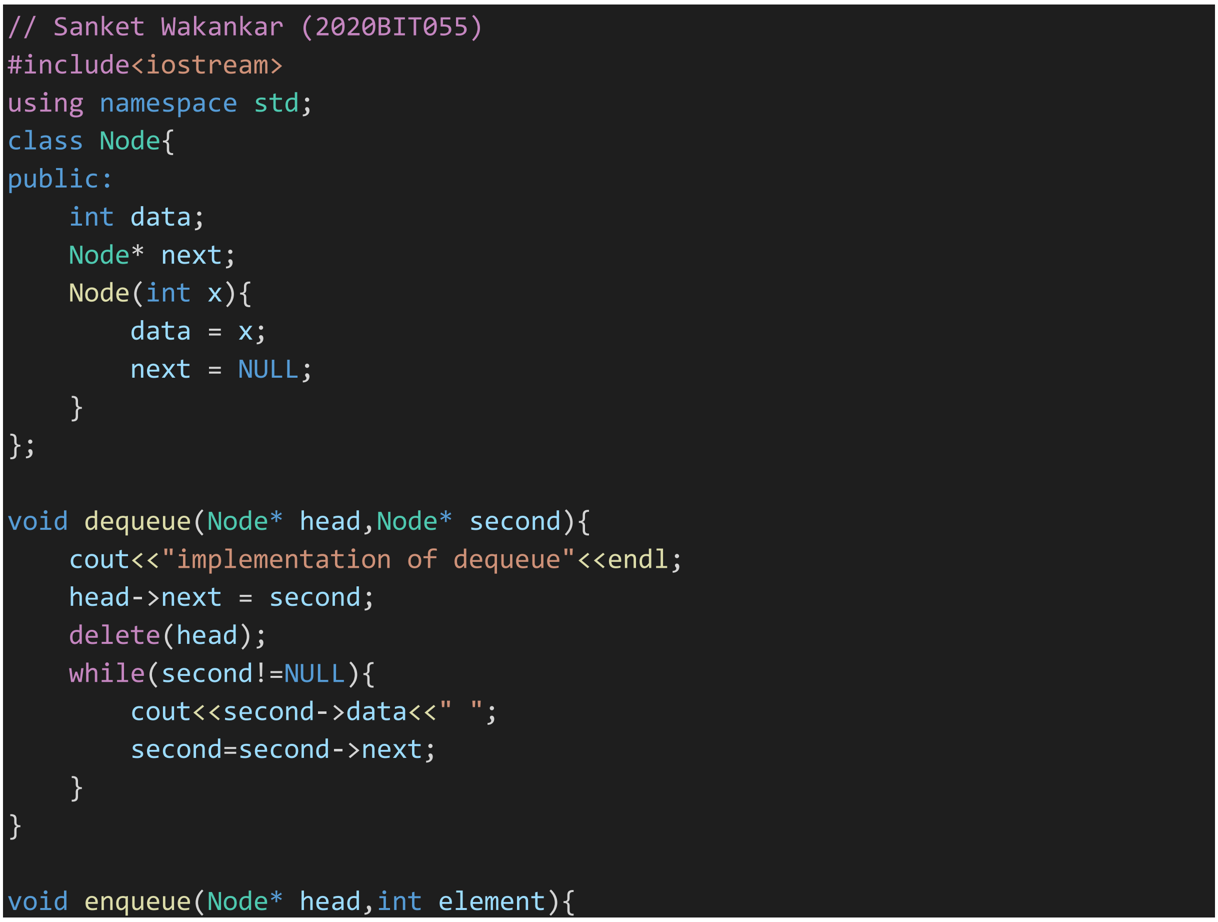
```
    Node* second = new Node(20);
    Node* third = new Node(30);
    Node* fourth = new Node(40);

    head->next = second;
    second->next = third;
    second->prev = head;
    third->next = fourth;
    third->prev = second;
    fourth->next = NULL;
    fourth->prev = third;

    Node* c = insert(head,5);
    print_list(c);
}
```

Output:

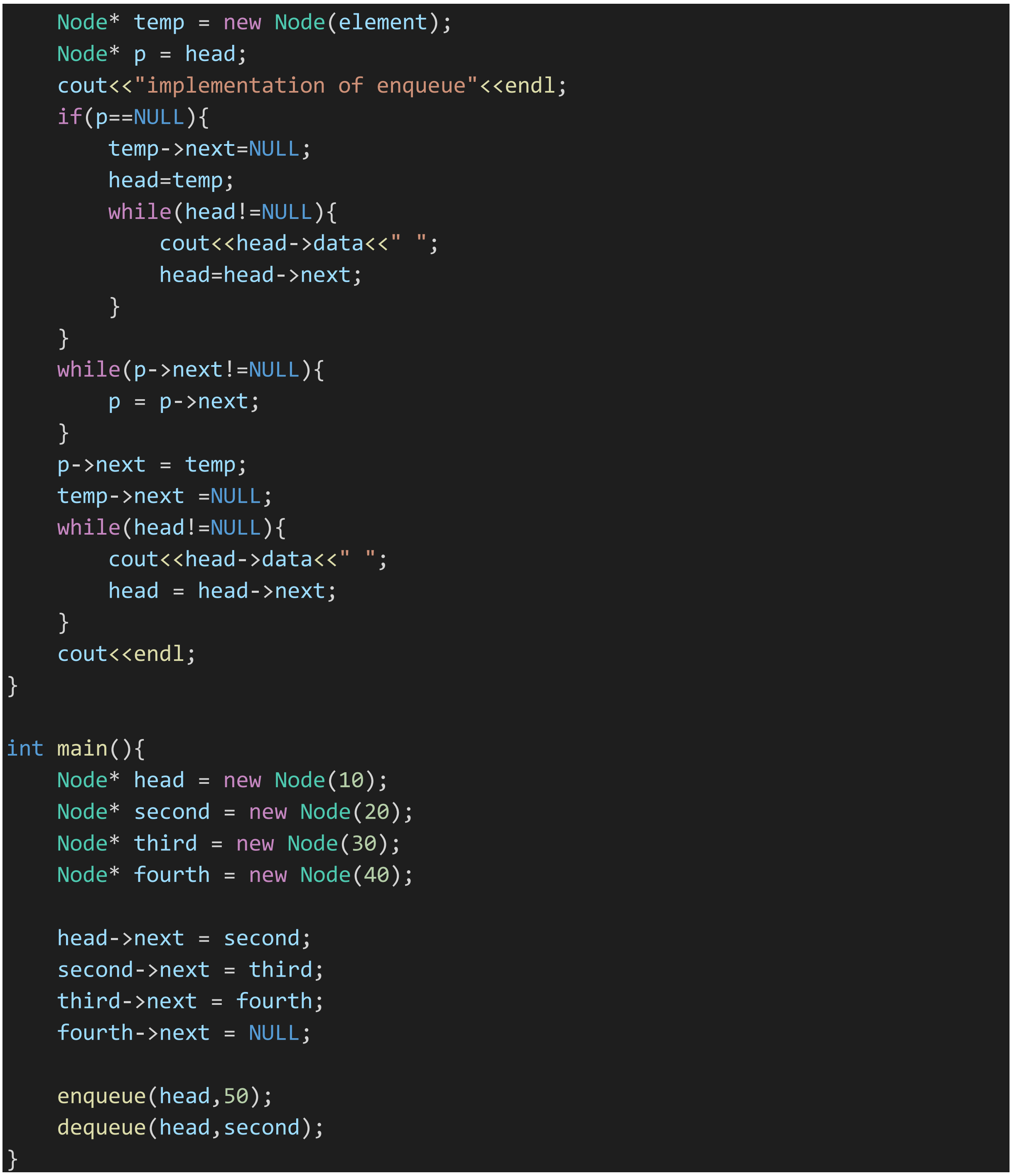


```
PS C:\Users\DELL\Documents\DSA Programming Using CPP> cd "c:\Users\DELL\Documents\DSA Programming Using CPP\.
 ($?) { g++ insert_node_atbeginning.cpp -o insert_node_atbeginning } ; if ($?) { .\insert_node_atbeginning }
5 10 20 30 40
PS C:\Users\DELL\Documents\DSA Programming Using CPP\.vscode\linked_list\doubly linked_list>
```

4 & 5. Enqueue & Dequeue:

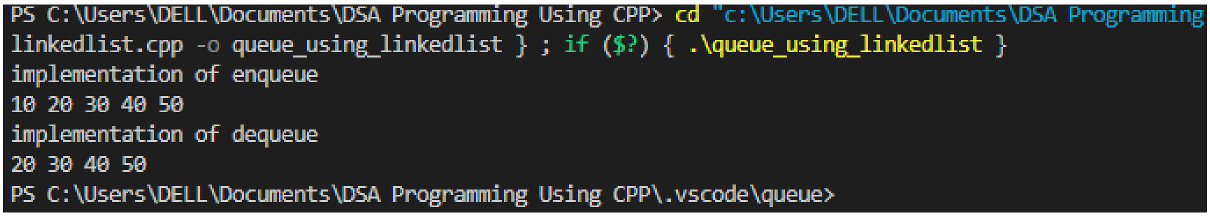


```
// Sanket Wakankar (2020BIT055)
#include<iostream>
using namespace std;
class Node{
public:
    int data;
    Node* next;
    Node(int x){
        data = x;
        next = NULL;
    }
};

void dequeue(Node* head,Node* second){
    cout<<"implementation of dequeue"<<endl;
    head->next = second;
    delete(head);
    while(second!=NULL){
        cout<<second->data<<" ";
        second=second->next;
    }
}

void enqueue(Node* head,int element){
```



```cpp
    Node* temp = new Node(element);
    Node* p = head;
    cout<<"implementation of enqueue"<<endl;
    if(p==NULL){
        temp->next=NULL;
        head=temp;
        while(head!=NULL){
            cout<<head->data<<" ";
            head=head->next;
        }
    }
    while(p->next!=NULL){
        p = p->next;
    }
    p->next = temp;
    temp->next =NULL;
    while(head!=NULL){
        cout<<head->data<<" ";
        head = head->next;
    }
    cout<<endl;
}

int main(){
    Node* head = new Node(10);
    Node* second = new Node(20);
    Node* third = new Node(30);
    Node* fourth = new Node(40);

    head->next = second;
    second->next = third;
    third->next = fourth;
    fourth->next = NULL;

    enqueue(head,50);
    dequeue(head,second);
}
```

Output:



```
PS C:\Users\DELL\Documents\DSA Programming Using CPP> cd "c:\Users\DELL\Documents\DSA Programming
linkedlist.cpp -o queue_using_linkedlist } ; if ($?) { .\queue_using_linkedlist }
implementation of enqueue
10 20 30 40 50
implementation of dequeue
20 30 40 50
PS C:\Users\DELL\Documents\DSA Programming Using CPP\.vscode\queue>
```