

1) IMPLEMENTING STACKS FROM SCRATCH

This is the standard sum of the stacks.

USING ARRAYS:

```
#include<iostream>
using namespace std;

class stack{
    private:
        int stack_size=5;
        int *stack_array=new int[stack_size];
        int stack_top=0;

    public:
        void push(int n){
            if(stack_top==stack_size){
                cout<<"\n The heap is exhausted (stack overflow).";
            }
            else{
                stack_array[stack_top]=n;
                stack_top++;
            }
        }
        void pop(){
            if(stack_top==0){
                cout<<"\n The heap is Empty (stack underflow).";
            }
            else{
                stack_top--;
            }
        }
        void peek(){
            if(stack_top==0){
                cout<<"\n The stack is empty so no element to peek.";
            }
            else{
                cout<<"\n The Topmost element of the stack is
```

```

"<<stack_array[stack_top-1];
    }
}
};

int main(){
    stack s;
    cout<<"\n Operations : ";
    s.peek();
    s.pop();
    s.push(10);
    s.push(23);
    s.push(7);
    s.push(98);
    s.push(67);
    s.push(11);
    s.peek();
    s.pop();
    s.peek();
    return 0;
}

```

USING LINKED LIST:

```

#include<iostream>
using namespace std;

class node{
public:
    int data;
    node *next=NULL;
};

class stack{
private:
    node *head=NULL;
    int stack_size=0;

public:

```

```

void push(int n) {
    node *t=new node;
    if(t==NULL) {
        cout<<"\n The heap is exhausted (stack overflow).";
    }
    else{
        t->data=n;
        if(head==NULL) {
            head=t;
        }
        else{
            t->next=head;
            head=t;
        }
        stack_size++;
    }
}

void pop() {
    if(stack_size==0) {
        cout<<"\n The heap is Empty (stack underflow).";
    }
    else{
        node *t=head;
        head=head->next;
        delete(t);
    }
}

void peek() {
    if(stack_size==0) {
        cout<<"\n The stack is empty so no element to peek.";
    }
    else{
        cout<<"\n The Topmost element of the stack is "<<head->data;
    }
}

};

int main() {
    stack s;

```

```
s.peek();  
s.pop();  
s.push(10);  
s.push(23);  
s.push(7);  
s.push(98);  
s.push(67);  
s.push(11);  
s.peek();  
s.pop();  
s.peek();  
return 0;  
}
```