**/\*Write a c++ program to implement Stack using class Templates**

**note:**

**1. Push - Insert the element onto the top of the Stack**

**if top<=size, read data and then insert element and print "10 has been inserted"**

**if top>size, then print "Stack overflow"**

**2. Pop - Delete the element from top of the Stack and print msg "10 has been deleted"**

**& if Stack is empty then print "Stack underflow"**

**3. getTop - Displays the top most element of the Stack and print msg "Top most element is 50"**

**& if Stack is empty print "Stack underflow"**

**4. Display in push order - displays all the elements of the Stack in the order of insertion**

**& if Stack is empty then print "Stack underflow"**

**5. Display in pop order - displays all the elements of the Stack in LIFO order**

**& if Stack is empty then print "Stack underflow"**

**6. Exit**

sample input & output

Enter size of array

5

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

1

Enter data 10

10 has been inserted

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

1

Enter data 20

20 has been inserted

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

1

Enter data 30

30 has been inserted

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

1

Enter data 40

40 has been inserted

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

1

Enter data 50

50 has been inserted

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

1

Enter data 60

Stack overflow

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

4

10

20

30

40

50

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

5

50

40

30

20

10

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

3

Top most element is 50

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

2

50 has been deleted

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

2

40 has been deleted

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

2

30 has been deleted

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

2

20 has been deleted

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

2

10 has been deleted

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

2

Stack underflow

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

3

Stack underflow

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

4

Stack underflow

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

5

Stack underflow

1. Push 2. Pop 3. Get top most element 4. Display in push order 5. Display in pop order 6. Exit

Enter choice

6

Exiting

\*/

#include<iostream>

using namespace std;

template <class T>

class Stack\_Array{

T \*s;

int top,max;

public:

Stack\_Array(int);

void push(T x);

void pop();

int getTop();

void display\_pushorder();

void display\_poporder();

};

template <class T>

Stack\_Array<T>::Stack\_Array(int size)

{

top=-1;

max=size;

s=new T[max];

}

template <class T>

void Stack\_Array<T>::push(T x)

{

if(top==max-1)

{

cout<<"Stack overflow"<<endl;

return;

}

top++;

s[top]=x;

cout<<x<<" has been inserted"<<endl;

}

template <class T>

void Stack\_Array<T>::pop()

{

T temp;

if(top==-1)

{

cout<<"Stack underflow"<<endl;

return;

}

temp=s[top];

top--;

cout<<temp<<" has been deleted"<<endl;

}

template <class T>

int Stack\_Array<T>::getTop()

{

if(top==-1)

return top;

else

return s[top];

}

template <class T>

void Stack\_Array<T>::display\_pushorder()

{

int i;

if(top==-1)

{

cout<<"Stack underflow"<<endl;

return;

}

for(i=0;i<=top;i++)

cout<<s[i]<<endl;

}

template <class T>

void Stack\_Array<T>::display\_poporder()

{

int i;

if(top==-1)

{

cout<<"Stack underflow"<<endl;

return;

}

for(i=top;i>=0;i--)

cout<<s[i]<<endl;

}

int main()

{

int ch,size,num;

cout<<"Enter size of array"<<endl;

cin>>size;

Stack\_Array<int> s(size);

do

{

cout<<"\n1. Push ";

cout<<"2. Pop ";

cout<<"3. Get top most element ";

cout<<"4. Display in push order ";

cout<<"5. Display in pop order ";

cout<<"6. Exit"<<endl;

cout<<"Enter choice"<<endl;

cin>>ch;

switch(ch)

{

case 1: cout<<"Enter data ";

cin>>num;

s.push(num);

break;

case 2:s.pop();

break;

case 3:num=s.getTop();

if(num==-1)

cout<<"Stack underflow"<<endl;

else

cout<<"Top most element is "<<num<<endl;

break;

case 4:s.display\_pushorder();

break;

case 5:s.display\_poporder();

break;

case 6:cout<<"Exiting"<<endl;

break;

default:

cout<<"Wrong choice"<<endl;

}

}while(ch!=6);

return 0;

}