

```

//Standard Library stack adapter class {can be implemented as vector,deque,list}
#include<iostream>
#include<conio.h>
#include<vector>
#include<list>
#include<deque>
#include<stack>
using namespace std;
template<class T> void pushelement(T & s);      //pushelement() function prototype
template<class T> void popelement(T & s); //popelement() function prototype
int main()
{
    stack <int> dequestack; // stack with default underlying deque

    stack <int,vector<int> > vectorstack;      //stack with underlying int vector

    stack <int,list<int> > liststack; //stack with underlying int list

    //push 10 elements on each of these stacks (i.e
dequestack,vectorstack,liststack)

    cout<<"\n\npushing elements onto dequestack : ";
    pushelement(dequestack);
    cout<<"\n\npushing elements onto vectorstack : ";
    pushelement(vectorstack);
    cout<<"\n\npushing elements onto liststack : ";
    pushelement(liststack);

    cout<<"\n\n-----";

    cout<<"\n\npopping element from dequestack : ";
    popelement(dequestack);
    cout<<"\n\npopping element from vectorstack : ";
    popelement(vectorstack);
    cout<<"\n\npopping element from liststack : ";
    popelement(liststack);

    getch();
    return 0;
}
//Function definition for pushelement()
template<class T>
void pushelement(T & s)
{
    for(int i=0;i<10;i++)
    {
        s.push(i);
        cout<<s.top()<<" ";
    }
}
//Function definition for popelement()
template<class T>
void popelement(T & s)
{
    while(!s.empty())
    {
        cout<<s.top()<<" ";
        s.pop();
    }
}

```

}