**Code**

#include<iostream>

using namespace std;

const int size=10;

template <class t>

class StackOperations

{

t a[size];

int top;

public:

void push(t);

t pop();

int isempty();

int isfull();

t topmost();

void clear();

void display();

StackOperations()

{

top=-1;

}

};

template <class t>

void StackOperations<t>::push(t p)

{

top++;

a[top]=p;

}

template <class t>

t StackOperations<t>::pop()

{

t d;

d=a[top];

top--;

return d;

}

template <class t>

void StackOperations<t>::display()

{

if(top==-1)

cout<<"Stack is empty "<<endl;

else

{

cout<<"Contents of the stack starting from the top are ";

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

cout<<a[i]<<" ";

cout<<endl;

}

}

template <class t>

int StackOperations<t>::isempty()

{

if(top==-1)

{

return(1);

}

else

{

return(0);

}

}

template <class t>

int StackOperations<t>::isfull()

{

if(top==size-1)

{

return(1);

}

else

{

return(0);

}

}

template <class t>

t StackOperations<t>::topmost()

{

t d;

d=a[top];

return d;

}

template <class t>

void StackOperations<t>::clear()

{

top=-1;

}

int main()

{

StackOperations<int> s1;

int choice,e,f,p,d,a;

char ch='y';

do

{

cout<<"Main Menu "<<endl;

cout<<"1.Push onto the stack "<<endl;

cout<<"2.Pop from the stack "<<endl;

cout<<"3.Check if stack is empty "<<endl;

cout<<"4.Check if stack is full "<<endl;

cout<<"5.Return the topmost element of the element "<<endl;

cout<<"6.Clear the stack "<<endl;

cout<<"Enter your choice "<<endl;

cin>>choice;

switch(choice)

{

case 1: f=s1.isfull();

if(f==1)

cout<<"Stack is full so insertion not possible ";

else

{

cout<<"Enter the element to push ";

cin>>p;

s1.push(p);

s1.display();

}

break;

case 2: e=s1.isempty();

if(e==1)

cout<<endl<<"Deletion is not possible ";

else

{

d=s1.pop();

s1.display();

}

break;

case 3: e=s1.isempty();

if(e==1)

cout<<endl<<"Stack is empty ";

else

cout<<"Stack is not empty "<<endl;

break;

case 4: f=s1.isfull();

if(f==1)

cout<<endl<<"Stack is full ";

else

cout<<"Stack is not full "<<endl;

break;

case 5: a=s1.topmost();

s1.display();

break;

case 6 : s1.clear();

s1.display();

break;

default: cout<<"Error in input. ";

}

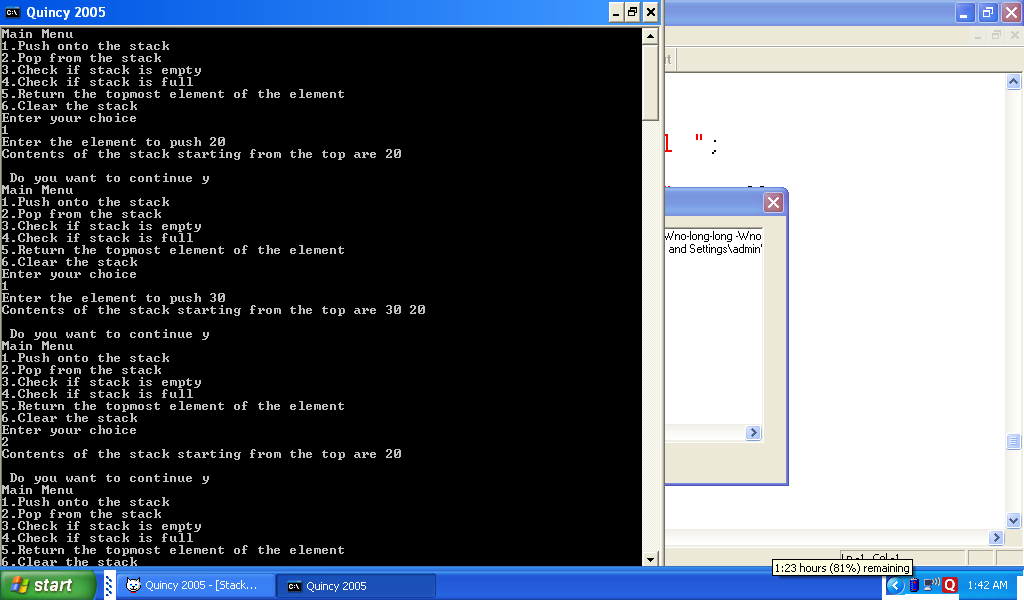
cout<<"\n Do you want to continue ";

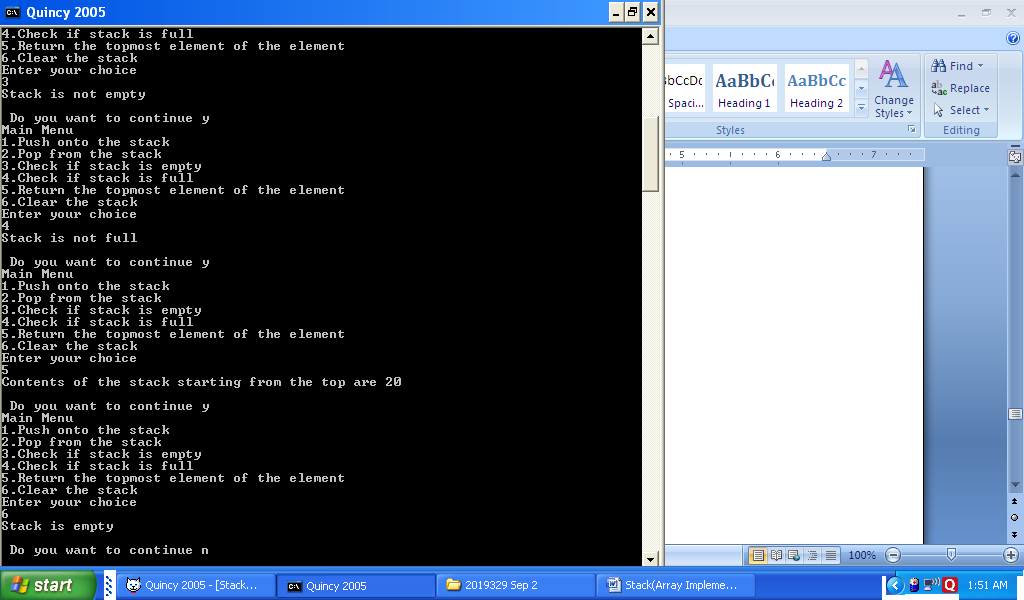
cin>>ch;

}while(ch=='y');

return 0;

}



****