Submitted By : Tanzeela Asghar

Submitted To : Sir Rehan Ahmed

Reg no : 2021-BSE-032

Section :BSE (3A)

LAB 06

Code Task # 01

Use Stack to solve postfix expression as discussed in Lab hours

CODE:

```cpp
#include "stdafx.h"
#include <iostream>
#include<string>
using namespace std;
const int size=20;
class Stack
{
    int mytop;
    int arr[size];
public:
    Stack()
    {
        mytop=-1;
    }
    bool is_empty()
    {
        if (mytop==-1)
            return true;
        else
            return false;
    }
```

```cpp
bool is_full()
{

if(mytop==size-1)
    return true;
else
            return false;
}
void push(int x)
{
    if(!is_full())
    {
        mytop++;
        arr[mytop]=x;
    }
    else
        cout<<"stack is full";
}
int pop()
{
    if(!is_empty())
        return arr[mytop--];
    else
    {
        cout<<"stack is empty"<<endl;
        return -1;
    }
}
int top()
{
    return arr[mytop];
}
```

```cpp
};
int _tmain(int argc, _TCHAR* argv[])
{
    Stack s1;
    string postfix= "273-/215+*+";
    for(int i=0;i<postfix.length();i++)
    {
        if((postfix[i]>='0')&&(postfix[i]<='9'))
            s1.push(postfix[i] - '0');
        else
        {
            int a=s1.top();
            s1.pop();
            int b=s1.top();
            s1.pop();
            switch(postfix[i])
            {
                case '+':
                s1.push(b+a);
                    break;
                case '-':
                s1.push(b-a);
                    break;
                case '*':
                s1.push(b*a);
                    break;
                case '/':
                s1.push(b/a);
                    break;
            }
        }
    }
```
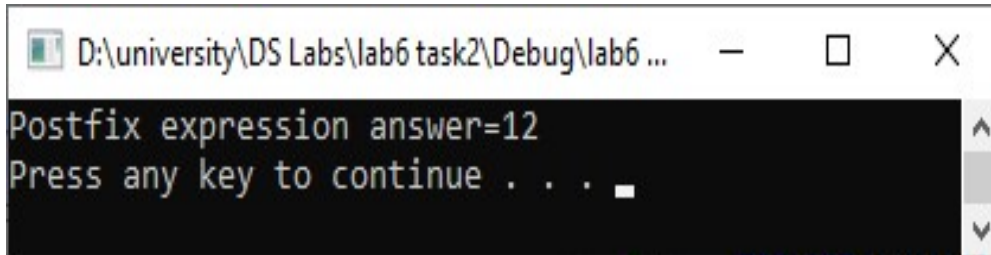
```
    cout<<"Postfix expression answer="<<s1.top()<<endl;
        system ("pause");
        return 0;
}
```

OUTPUT


```
Postfix expression answer=12
Press any key to continue . . . _
```

Code Task # 02

Use Stack AND Queue combination to find weather a word is a PALINDROME or not as discussed in Lab hours

CODE:

```
#include "stdafx.h"
#include<iostrea
m>
#include<string>
using namespace
std; const int
size=20; class
Stack
{
    int
mytop;    int
arr[size];
public:
    Stack()
    {
        mytop=-1;
```

```cpp
        }
    bool is_empty()
        {
                if(mytop==-1)
                        return
true;                   else
                        return false;
        }
    bool is_full()
        {
            if(mytop==size-1)
                        return true;
                else
                        return false;
        }
    void push(int x)
        {
            mytop++;
            arr[mytop]=x;
        }
     int pop()
        {
            return arr[mytop--];
        }
     int top()
        {
            return arr[mytop];
        }
}; class
Queue
{
```

```cpp
        int
front,rear;
int arr[size];
public:
    Queue()
    {
            rear=-1;
    front=-1;
    }
    bool is_empty()
    {
        if((front==-1)&&(rear==-1))
                return true;
            else
                return false;
    }
    void addqueue(int x)
    {
        if(is_empty())
        {
                front++;
            rear++;
                arr[rear]=x;
        }
        else if(!is_empty())
        {
                rear++;
                arr[rear]=x;
        }
            else
                cout<<"Queue is full\n";
    }
```

```cpp
    int dequeue()
    {
        if(!is_empty())
        {
            return arr[front++];
        }
    }
};

int _tmain(int argc, _TCHAR* argv[])
{
    string s1;
    Stack s;
    Queue q;
    bool check=true;
    cout<<"Enter word you want to check: ";
    cin>>s1;
    for(int i=0;i<s1.length();i++)
    {
        s.push(s1[i]);
        q.addqueue(s1[i]);
    }
    for(int i=0;i<s1.length();i++)
    {
        if(s.pop()==q.dequeue())
                check=true;
        else
        {
            check=false;
            break;
        }
    }
```
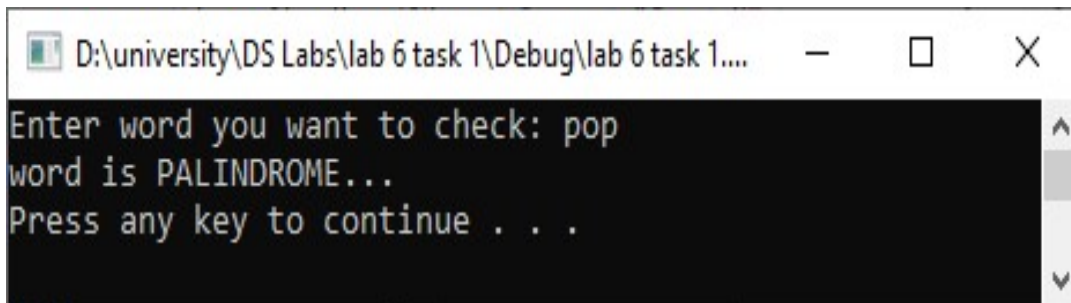
```cpp
        if(check==true)
                cout<<"word is
PALINDROME..."<<endl;  else
                cout<<"word is not a PALINDROME..."<<endl;
        system("pause");
        return 0;
}
```

OUTPUT: