

# Week 3 Lab A – Stack& Queue

Avni Arora\_20103153\_B6\_week#3

1)

Output:

x=3

y=9

7

13

4

7

2)

**With STL**

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n;
```

```
    cout<<"Enter the Number: ";
```

```
    cin>>n;
```

```
    stack<int> s;
```

```
    int i = 2;
```

```
    while (n != 1)
```

```
    {
```

```
        if (n % i == 0)
```

```
        {
```

```
            s.push(i);
```

```
            while (n%i == 0)
```

```
            {
```

```
                n = n / i;
```

```

    }
}
i++;
}
cout<<"Prime Factors of given Number are: ";
while (!s.empty())
{

    printf("%d ", s.top());
    s.pop();
}
return 0;
}

```

Output:

```

Enter the Number: 30
Prime Factors of given Number are: 5 3 2
Process returned 0 (0x0)   execution time : 21.967 s
Press any key to continue.

```

## Without STL

```

#include <bits/stdc++.h>
using namespace std;
struct Node
{
    int data;
    struct Node* next;
};
class Stack
{
private:

```

```
Node *top;

public:
    Stack()
    {
        top=NULL;
    }
    void push(int data)
    {
        Node* temp;
        temp = new Node();
        if (!temp)
        {
            cout << "\error";
            return;
        }
        temp->data = data;
        temp->next = top;
        top = temp;
    }
    int isEmpty()
    {
        return top == NULL;
    }
    int tp()
    {
        if (!isEmpty())
            return top->data;
        else
            exit(0);
    }
```

```
void pop()
{
    Node* temp;
    if (top == NULL)
    {
        cout << "\nStack is empty now" << endl;
        return;
    }
    else
    {
        temp = top;
        top = top->next;
        temp->next= NULL;
        delete(temp);
    }
}
```

```
void display()
{
    Node* temp;
    if (top == NULL)
    {
        cout << "\nStack is empty";
        return;
    }
    else
    {
        temp = top;
        while (temp != NULL)
        {
            cout << temp->data << " ";
```

```

        temp = temp->next;
    }
}
};

int main()
{
    Stack s;
    cout<<"Enter The Number: ";
    int n;
    cin>>n;
    int i=2;
    while (n != 1)
    {
        if (n % i == 0)
        {
            s.push(i);
            while (n%i == 0)
            {
                n = n / i;
            }
        }
        i++;
    }
    cout<<"Prime Factors of given Number are: ";

    s.display();

    return 0;
}

```

Output:

```
Enter The Number: 30
Prime Factors of given Number are: 5 3 2
Process returned 0 (0x0)   execution time : 2.781 s
Press any key to continue.
```

3)

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    cout<<"Enter no of elements: ";
```

```
    int n;
```

```
    cin>>n;
```

```
    stack<int> s,s1,s2,s3;
```

```
    cout<<"Enter elements for the Stack: ";
```

```
    for(int i=0;i<n;i++)
```

```
    {
```

```
        int a;
```

```
        cin>>a;
```

```
        s.push(a);
```

```
    }
```

```
    for(int i=0;i<n/2;i++)
```

```
    {
```

```
        int a=s.top();
```

```
        s1.push(a);
```

```
        s.pop();
```

```

    }
    while(!s.empty())
    {
        int a=s.top();
        s2.push(a);
        s.pop();
    }
    while(!s1.empty())
    {
        int a=s1.top();
        s3.push(a);
        s1.pop();
    }
    while(!s2.empty())
    {
        int a=s2.top();
        s3.push(a);
        s2.pop();
    }

    while(!s3.empty())
    {
        cout<<s3.top()<<" ";
        s3.pop();
    }

    return 0;
}

```

Output:

```
Enter no of elements: 10
Enter elements for the Stack: 1 2 3 4 5 6 7 8 9 10
5 4 3 2 1 10 9 8 7 6
Process returned 0 (0x0)   execution time : 8.986 s
Press any key to continue.
```

4)

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    cout<<"Enter number: ";
```

```
    int n;
```

```
    cin>>n;
```

```
    int base;
```

```
    cout<<"enter base: (between 2 to 9): ";
```

```
    cin>>base;
```

```
    stack<int> s;
```

```
    while(n!=0)
```

```
    {
```

```
        s.push(n%base);
```

```
        n=n/base;
```

```
    }
```

```
    cout<<"Converted Number: ";
```

```
    while(!s.empty())
```

```
    {
```

```
        cout<<s.top();
```

```
        s.pop();
```

```
    }
```

```
    return 0;
```



}

Output:

```
Enter number: 6
enter base: (between 2 to 9): 2
Converted Number: 110
Process returned 0 (0x0)   execution time : 4.013 s
Press any key to continue.
```

5)

a)

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
string postfixtoprefix(string s1)
```

```
{
```

```
    stack<string> s;
```

```
    for(int i=0; i< s1.length(); i++)
```

```
    {
```

```
        char c=s1[i];
```

```
        if((c>='A'&&c<='Z')||(c>='a'&&c<='z'))
```

```
        {
```

```
            string a(1, s1[i]);
```

```
            s.push(a);
```

```
        }
```

```
    else
```

```
    {
```

```
        string op1 = s.top();
```

```
        s.pop();
```

```
        string op2 = s.top();
```

```
        s.pop();
```

```
        string s2 = s1[i] + op2 + op1;
```

```
        s.push(s2);
```

```

    }

}

return s.top();
}

int main()
{
    cout<<"Enter postfix expression: ";
    string s;
    cin>>s;
    cout<<"Prefix Expression: "<<postfixtoprefix(s);
    return 0;
}

```

Output:

```

Enter postfix expression: AB+C*DE-FG+*-
Prefix Expression: -*+ABC*-DE+FG
Process returned 0 (0x0)   execution time : 28.133 s
Press any key to continue.

```

b)

```

#include <bits/stdc++.h>

using namespace std;

string prefixtopostfix(string s1)
{
    stack<string> s;
    for(int i=s1.length()-1; i>=0; i--)
    {

```

```

char c=s1[i];
if((c>='A'&&c<='Z')||(c>='a'&&c<='z'))
{
    string a(1, s1[i]);
    s.push(a);
}
else
{
    string op1 = s.top();
    s.pop();
    string op2 = s.top();
    s.pop();
    string s2 = op1+op2 +s1[i];
    s.push(s2);

}

}

return s.top();
}

```

```

int main()
{
    cout<<"Enter prefix expression: ";
    string s;
    cin>>s;
    cout<<"Postfix Expression: "<<prefixtopostfix(s);
    return 0;
}

```

Output:

```
Enter prefix expression: -*+ABC*-DE+FG
Postfix Expression: AB+C*DE-FG+*-
Process returned 0 (0x0)   execution time : 19.874 s
Press any key to continue.
```

c)

```
#include <bits/stdc++.h>

using namespace std;

int precision(char c) {
    if(c == '^')
        return 3;
    else if(c == '/' || c=='*')
        return 2;
    else if(c == '+' || c == '-')
        return 1;
    else
        return -1;
}

string infixToPostfix(string s)
{
    stack<char> st;
    string s1;
    for(int i = 0; i < s.length(); i++)
    {
        char c = s[i];
        if( (c >= '0' && c <= '9') || (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') )
            s1 += c;
        else if(c == '(')
```

```

        st.push('(');
    else if(c == ')')
    {
        while(st.top() != '(')
        {
            s1 += st.top();
            st.pop();
        }
        st.pop();
    }
    else {
        while(!st.empty() && precision(s[i]) <= precision(st.top())) {
            s1 += st.top();
            st.pop();
        }
        st.push(c);
    }
}

while(!st.empty()) {
    s1 += st.top();
    st.pop();
}

return s1;
}

int value ( string s)
{
    string s1=infixToPostfix(s);
    stack<int> st;
    for (int i = 0;i<s1.length(); ++i)

```

```

{
    if (s1[i]>='0'&& s1[i]<='9')
    {
        st.push(s1[i]-'0');
    }
    else
    {
        int val1 = st.top();
        st.pop();
        int val2 = st.top();
        switch (s1[i])
        {
            case '+': st.push(val2 + val1); break;
            case '-': st.push(val2 - val1); break;
            case '*': st.push(val2 * val1); break;
            case '/': st.push(val2/val1); break;
        }
    }
}
return st.top();
}

int main()
{
    cout<<"Enter prefix expression: ";
    string s;
    cin>>s;
    cout<<"Postfix Expression: "<<infixToPostfix(s);
    string s1="(4+9*6)-((8-6)/2*4)*9/3";
    cout<<"\nValue of the expression: (4 + 9 * 6) - ((8 - 6) / 2 * 4) * 9 / 3 : "<<value(s1);
    return 0;
}

```

```
}
```

Output:

```
Enter prefix expression: A+B*C/D-F+A*E
Postfix Expression: ABC*D/+F-AE*+
Value of the expression: (4 + 9 * 6) - ((8 ÷ 6) / 2 * 4) * 9 / 3 : 24
Process returned 0 (0x0)   execution time : 70.565 s
Press any key to continue.
```

6)

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
bool balance(string s1)
```

```
{
```

```
    stack<char> s;
```

```
    char x;
```

```
    for (int i = 0; i < s1.length(); i++)
```

```
    {
```

```
        if (s1[i] == '(' || s1[i] == '['
            || s1[i] == '{' || s1[i] == '/')
```

```
        {
```

```
            s.push(s1[i]);
```

```
            if(s1[i]== '/')
```

```
                i++;
```

```
            continue;
```

```
        }
```

```
        if (s.empty())
```

```
            return false;
```

```
        switch (s1[i])
```

```
        {
```

```
            case ')':
```

```
x = s.top();  
s.pop();  
if (x == '{' || x == '['||x=='/')  
    return false;  
break;
```

```
case '}':  
    x = s.top();  
    s.pop();  
    if (x == '(' || x == '['||x=='/')  
        return false;  
    break;
```

```
case ']':  
    x = s.top();  
    s.pop();  
    if (x == '(' || x == '{'||x=='/')  
        return false;  
    break;
```

```
case '*':  
    x = s.top();  
    i++;  
    s.pop();  
    if (x == '(' || x == '{'||x=='[')  
        return false;  
    break;
```

```
}
```

```
}
```

```
return (s.empty());
```

```
}
```



```

int main()
{
    cout<<"Enter expression: ";
    string s;
    cin>>s;
    if(balance(s))
    {
        cout<<"Yes the expression is balanced";
    }
    else
    {
        cout<<"No the expression is not balanced";
    }

    return 0;
}

```

Output:

```

Enter expression: ((/**/){})
Yes the expression is balanced
Process returned 0 (0x0)   execution time : 24.703 s
Press any key to continue.

```

7)

```

#include <bits/stdc++.h>
using namespace std;
void display(queue<char>q)
{
    while (!q.empty()) {
        cout<< q.front();
        q.pop();
    }
}

```

```

    }
    cout << '\n';
}
queue<char> compress (queue<char>q)
{
    queue<char>q1;
    while(!q.empty())
    {
        int count=0;
        char c=q.front();
        q.pop();
        if(c!=' ')
        {
            q1.push(c);
            while(!q.empty())
            {
                if(q.front()==c)
                {
                    count++;
                    q.pop();
                }
                else
                {
                    break;
                }
            }
            if(count>0)
            {
                q1.push((char)(count+1+48));
            }
        }
    }
}

```

```

    }
}
return q1;
}
int main()
{
    queue<char> q;
    cout<<"Enter text and end it with '.' :";
    while(1)
    {
        char c;
        cin>>c;
        if(c!='.')
            q.push(c);
        else
            break;
    }
    cout << "original text is : ";
    display(q);
    q=compress(q);
    cout<<"compressed text is : ";
    display(q);
    return 0;
}

```

Output:

```

Enter text and end it with '.' :asd ddfghjdff kj.
compressed text is : asd3fghjdf2kj

Process returned 0 (0x0)   execution time : 14.382 s
Press any key to continue.

```

8)

```
#include <bits/stdc++.h>
using namespace std;
void display(queue<int>q)
{
    while (!q.empty()) {
        cout<< q.front();
        q.pop();
    }
    cout << '\n';
}
queue<int> moveNthFront (queue<int>q,int pos)
{
    queue<int>q1,q2;
    while(!q.empty()&&--pos)
    {
        int c=q.front();
        q.pop();
        q1.push(c);
    }
    if(!q.empty())
    {
        q2.push(q.front());
        q.pop();
        while(!q1.empty())
        {
            q2.push(q1.front());
            q1.pop();
        }
    }
}
```

```

    }
    while(!q.empty())
    {
        q2.push(q.front());
        q.pop();
    }
    return q2;
}
else
    return q1;

}

int main()
{
    queue<int> q;
    cout<<"Enter queue (press 0 to end)";
    while(1)
    {
        int c;
        cin>>c;
        if(c!=0)
            q.push(c);
        else
            break;
    }
    cout << "original queue is : ";
    display(q);
    cout<<"Enter the position no to move to front : ";
    int n;
    cin>>n;

```

```

q=moveNthFront(q,n);
cout<<"after moving n th element queue is : ";
display(q);
return 0;
}

```

Output:

```

Enter queue (press 0 to end): 5 11 34 67 43 55 0
original queue is : 5 11 34 67 43 55
Enter the position no to move to front : 3
after moving n th element queue is : 34 5 11 67 43 55

Process returned 0 (0x0)   execution time : 15.009 s
Press any key to continue.

```

9)

```

#include<bits/stdc++.h>
using namespace std;
bool palindrome (stack<char> s, queue <char>q)
{
    while(!s.empty())
    {
        if(s.top()==q.front())
        {
            s.pop();
            q.pop();
        }
        else
            break;
    }
    if(s.empty())
    {
        return true;
    }
}

```

```

    }

    else
        return false;
}

int main()
{
    cout<<"enter the string:";
    string s;
    cin>>s;
    stack<char> st;
    queue<char> q;
    for(int i=0;i<s.length();i++)
    {
        st.push(s[i]);
        q.push(s[i]);

    }
    if(palindrome(st,q))
        cout<<"YES it is a palindrome";
    else
        cout<<"NO it is not a palindrome";
    return 0;
}

```

Output:

```

enter the string:noon
YES it is a palindrome
Process returned 0 (0x0)   execution time : 26.360 s
Press any key to continue.

```