

## C) Evaluation Of Postfix Expression

```
#include<iostream>

#include<stack>

Using namespace std;

float operation(int, int, char) ;

float scanNum(char);

int isOperator(char);

int isOperand(char);

float scanNum(char ch)
{
    int value;

    value = ch;

    return float(value-'0'); //return float from character
}

int isOperator(char ch)
{
    if(ch == '+' || ch == '-' || ch == '*' || ch == '/' || ch == '^')
        return 1; //character is an operator
    return -1; //not an operator
}

int isOperand(char ch)
{
    if(ch >= '0' && ch <= '9')
        return 1; //character is an operand
    return -1; //not an operand
}

float operation(int a, int b, char op)
{

```

```

float INT_MIN;; //Perform operation
if(op == '+')
return b+a;
else if(op == '-')
return b-a;
else if(op == '*')
return b*a;
else if(op == '/')
return b/a;
else if(op == '^')
return pow(b,a); //find b^a
else
return INT_MIN; //return negative infinity
}

float postfixEval(string postfix)
{
int a, b;
stack<float> stk;
string::iterator it;
for(it=postfix.begin();
it!=postfix.end(); it++)
{
//read elements and perform postfix evaluation
if(isOperator(*it) != -1)
{
b= stk.top();
stk.pop();
b= stk.top();

```

```

stk.pop();
stk.push(operation(a, b, *it));
}
else if(isOperand(*it) > 0)
{
stk.push(scanNum(*it));
}
}
return stk.top();
}
int main()
{
char postfix[10];
cout<<"Enter postfix Expression:\n";
cin>>postfix;
cout << "The result is: "<<postfixEval(postfix);
}

```

Output:

Enter postfix Expression:

1426++

The result is: 12

Process finished.