**Kathmandu University**

**Department of Computer Science and Engineering**

**Dhulikhel, Kavre**



**A Report on**

**COMP 202: Data Structures and Algorithms**

**Mini Project**

**Submitted by:**

Samip Timalsena(55)

**Submitted to:**

Dr. Rajani Chulyadyo

Department of Computer Science and Engineering

**Submission Date:**

January 19, 2019

**Task:**  To write a program to evaluate postfix expression and to find the time complexity of the program

To calculate postfix expression, we used the stack data structure.

Note: Here we considered only {+, −,∗,/ } operators

**Implementation:**

Algorithm

Input: A postfix expression, stack S

Output: an equivalent postfix expression

Steps:

1: Add ‘d’ to postfix expression.

2: Read postfix expression from left to right until ‘d’ is encountered.

3: If operand is encountered, push it into the stack S.

4: If operator is encountered, pop two elements from stack S,

i: a=pop()

ii: b=pop()

iii: c=a <operator> b

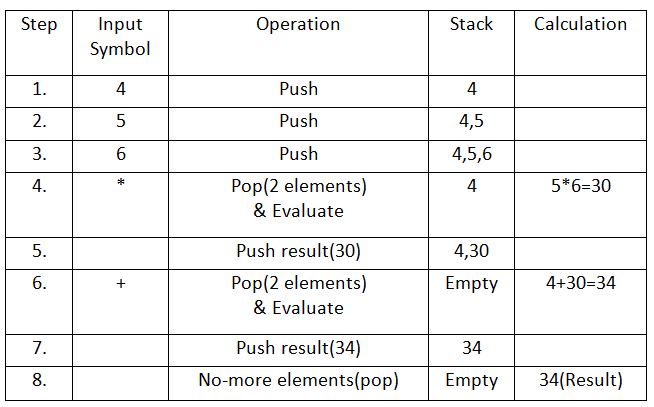
5: Push c to stack S

6: Set result=pop()

7: Print result

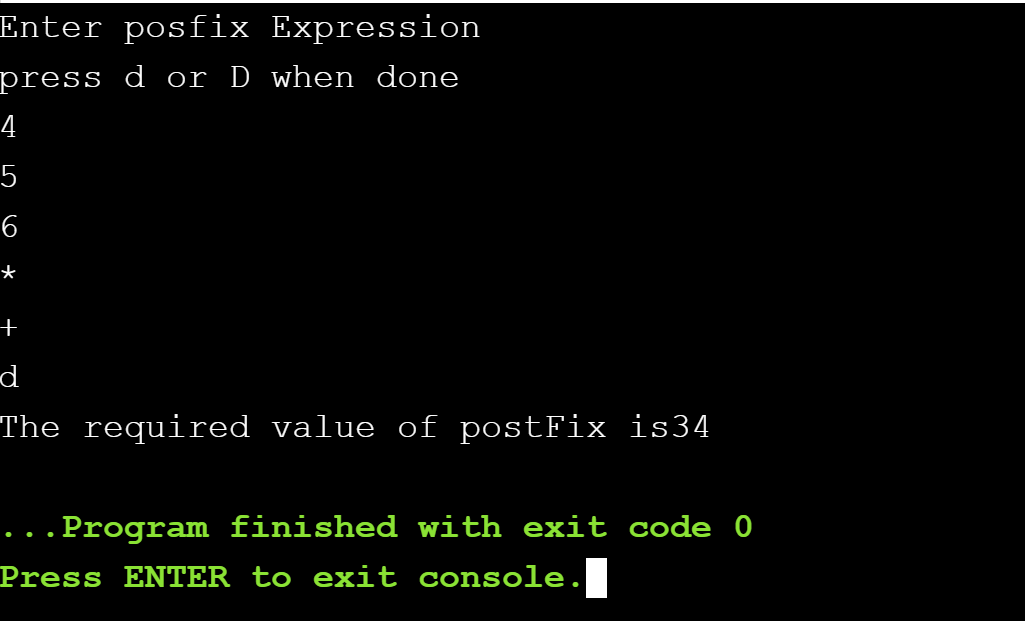
**Example:**

**Expression**: 456\*+



**RESULT**:34

**Output from Program:**



Hence, our program runs correctly.

**Time Complexity of the Program**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Program** | **Worst case** | **Best case** |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | int main(){  int i=0,size=1;  char postFix[POSTFIX\_SIZE];  cout<<"Enter posfix Expression"<<endl;  cout<<"press d or D when done"<<endl;  for(i=0;i<POSTFIX\_SIZE;i++){  cin>>postFix[i];  size++;  if(postFix[i]=='d'||postFix[i]=='D'){  break;  }}  postFixEval(postFix,size);  }  void postFixEval(char postFix[],int size){  int i,A,B;  int val=0;  char ch;  stack s;  for(int i=0;i<size-1;i++){  ch=postFix[i];  if(isdigit(ch)){  s.push(ch-'0');  }  if(ch == '+' || ch == '-' || ch == '\*' || ch == '/' ){  A=s.pop();  B=s.pop();  switch(ch){  case '+':  val=A+B;  break;  case '-':  val=B-A;  break;  case '\*':  val=B\*A;  break;  case '/':  val=B/A;  break; }  s.push(val);}}  cout<<"The required value of postFix is"<<s.pop();} | 0  2  0  1  1  n  n  n  n  1  0  0  0  0  0  1  0  0  n-1  n-2  n-2  0  0  n-2  n-2  n-2  n-2  n-2  n-2  1  0  0  0  0  0  0  0  0  0  n-2  1 | 0  2  0  1  1  n  n  n  n  1  0  0  0  0  0  1  0  0  n-1  n-2  n-2  n-2  0  n-2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 |

From table,

**Best case:** 9n-2

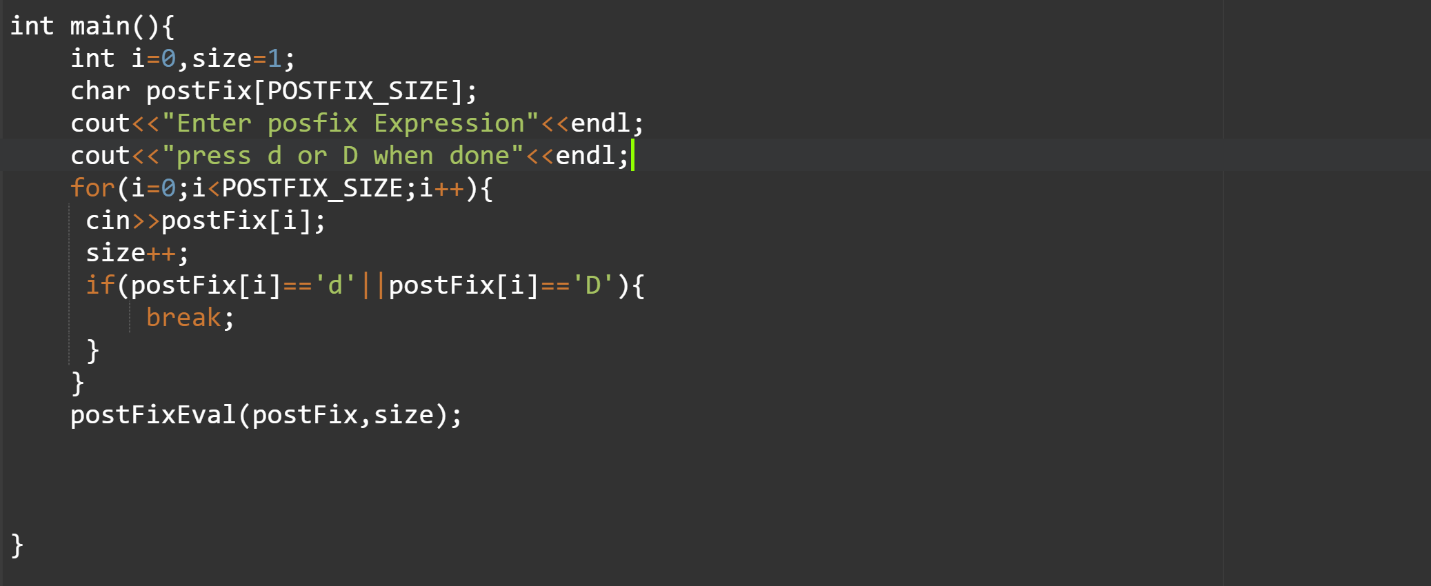
**Worst case:** 14n-11

**Average case:** 11.5n-6.5

Hence, Time complexity of program is O(n).

**Screenshots:**

Main Block:



Sample output (postfix expression:123+\*321-+\*):

