**Experiment No. 2**

**Title :** Evaluation of postfix expression.

**Problem Statement :** Evaluation of postfix expression using stack data structure

**Algorithm :**

**S1 :** Start

**S2 :** Declare an array of some size, a variable X is declared to keep the track of index of stack, an array of type ‘char’ to input postfix expression.

**S3 :** Call the functions in a switch statement according to the choice value

**S4 :** Declare push(), pop(), and convert()( to convert the character in expression to integer) and evaluate function

**S5 :** In a loop take every character of the expression

**S6 :** Check if the character is a digit with isdigit() if yes then push into stack.

**S7 :** If the character is an operator pop twice(operation is performed b/n newly entered 2 character) from stack and perform operation.

**S8 :** The value in stack at the end of the expression is answer of postfix expression

**S9 :** Stop

**Code :**

#include<stdio.h>

int stack[20];

int X = -1;

char postfix[100];

int i=0,j,a,b;

int ascii(char ch)

{

int i;

i = ch;

return((i-48));

}

void push(int k)

{

X = X + 1;

stack[X] = k;

}

int pop()

{

int s;

s = stack[X];

X = X -1;

return(s);

}

void evaluate()

{

for(i=0;postfix[i]!='\0';i++)

{

if(isdigit(postfix[i]))

{

j = ascii(postfix[i]);

push(j);

}

else

{

switch(postfix[i])

{

case '+' :

a = pop();

b = pop();

push((b+a));

break;

case '-' :

a = pop();

b = pop();

push((b-a));

break;

case '\*' :

a = pop();

b = pop();

push((b\*a));

break;

case '/' :

a = pop();

b = pop();

push((b/a));

break;

}

}

}

}

int main()

{

printf("Enter the postfix expression : ");

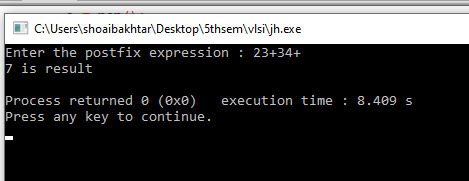
scanf("%s",postfix);

evaluate();

printf("%d is result\n",stack[X]);

}

**Output :**



**Analysis :**

* Operations will be performed between single digit number.