**5. Design and implement an algorithm to evaluate an arithmetic expression which is postfix form, and demonstrate its working and suitable examples**

**ALGORITHM**

Step 1: Add ) to postfix expression.  
Step 2:  Read postfix expression Left to Right until ) encountered  
Step 3:  If operand is encountered, push it onto Stack  
[End If]  
Step 4: If operator is encountered, Pop two elements  
i) A -> Top element  
ii) B-> Next to Top element  
iii) Evaluate B operator A  
push B operator A onto Stack  
Step 5: Set result = pop  
Step 6: END

**PROGRAM**

#include<stdio.h> //standard I/O header file

int stack[20]; //implementation of stack

int top = -1;

void push(int x) //push function

{

stack[++top] = x; //increment of top

}

int pop() //pop function

{

return stack[top--]; //decrement of top

}

int main() //main function

{

char exp[20]; //initializing expression size

char \*e; //initializing pointer to save address

int n1,n2,n3,num;

printf("Enter the expression :\n "); //printing statement

scanf("%s",exp); //assigning the value in exp

e = exp;

while(\*e != '\0') //condition check

{

if(isdigit(\*e)) //condition check

{

num = \*e - 48;

push(num);

}

else

{

n1 = pop();

n2 = pop();

switch(\*e) //switch statement according to option

{

case '+':

{

n3 = n1 + n2;

break;

}

case '-':

{

n3 = n2 - n1;

break;

}

case '\*':

{

n3 = n1 \* n2;

break;

}

case '/':

{

n3 = n2 / n1;

break;

}

}

push(n3);

}

e++;

}

printf("\nThe result of expression %s = %d\n\n",exp,pop()); //printing statement

return 0;

}

**OUTPUT**



