**Experiment No. 2**

**Title :** Evaluation of postfix expression.

**Problem Statement :** Write a C code to evaluate a postfix expression

**Algorithm :**

**Step 1:** Start.

**Step 2:** Declare a stack of type integer and a top variable ,define corresponding functions for it i.e. push() and pop().

**Step 3:** Declare a character array to store postfix expression and a function to get numeric value from ASCII value of a digit.

**Step 4:** Define a evaluate() function in which traverse through the character array of postfix expression and check if it is digit if yes then push to stack else pop two operators and perform the operation ,recursively call this evaluate() till null character .

**Step 5:** Input postfix expression and call evaluate function .

**Step 6:** Display the result.

**Step 7:** Stop

**Program:**

#include<stdio.h>

int stack[20]; //Declare stack int array

int top = -1;// stack top variable

char postfix[100]; //to store postfix expression

int i=0,j,a,b;

int ascii(char ch) //function to get numeric value of char type numbers

{

int i;

i = ch;

return((i-48));

}

void push(int val) //Push function

{

top = top + 1;//increment top

stack[top] = val;//push value

}

int pop() //Pop function

{

int s;

s = stack[top];//store value to pop

top = top -1;//decrement top

return(s);//return popped value

}

void evaluate() //Evaluate function

{

for(i=0;postfix[i]!='\0';i++)//traverse through the entered expression

{

if(isdigit(postfix[i]))//check if the char in expression is a digit

{

j = ascii(postfix[i]);//get the numeric value of type int

push(j);//push the integer value to stack

}

else//if its not digit the ts a operator

{

switch(postfix[i]) // switch to the operation based on the operator

{

//for each operation pop two operands and perform operation

//push the result to stack

case '+' ://addition

a = pop();

b = pop();

push((b+a));

break;

case '-' ://subtraction

a = pop();

b = pop();

push((b-a));

break;

case '\*' ://multiplication

a = pop();

b = pop();

push((b\*a));

break;

case '/' ://division

a = pop();

b = pop();

push((b/a));

break;

}

}

}

}

int main()

{

printf("\*\*\*\*\*P O S T F I X E V A A L U A T I O N \*\*\*\*\n");

printf("Enter the postfix expression : ");

scanf("%s",postfix);//input of postfix expression

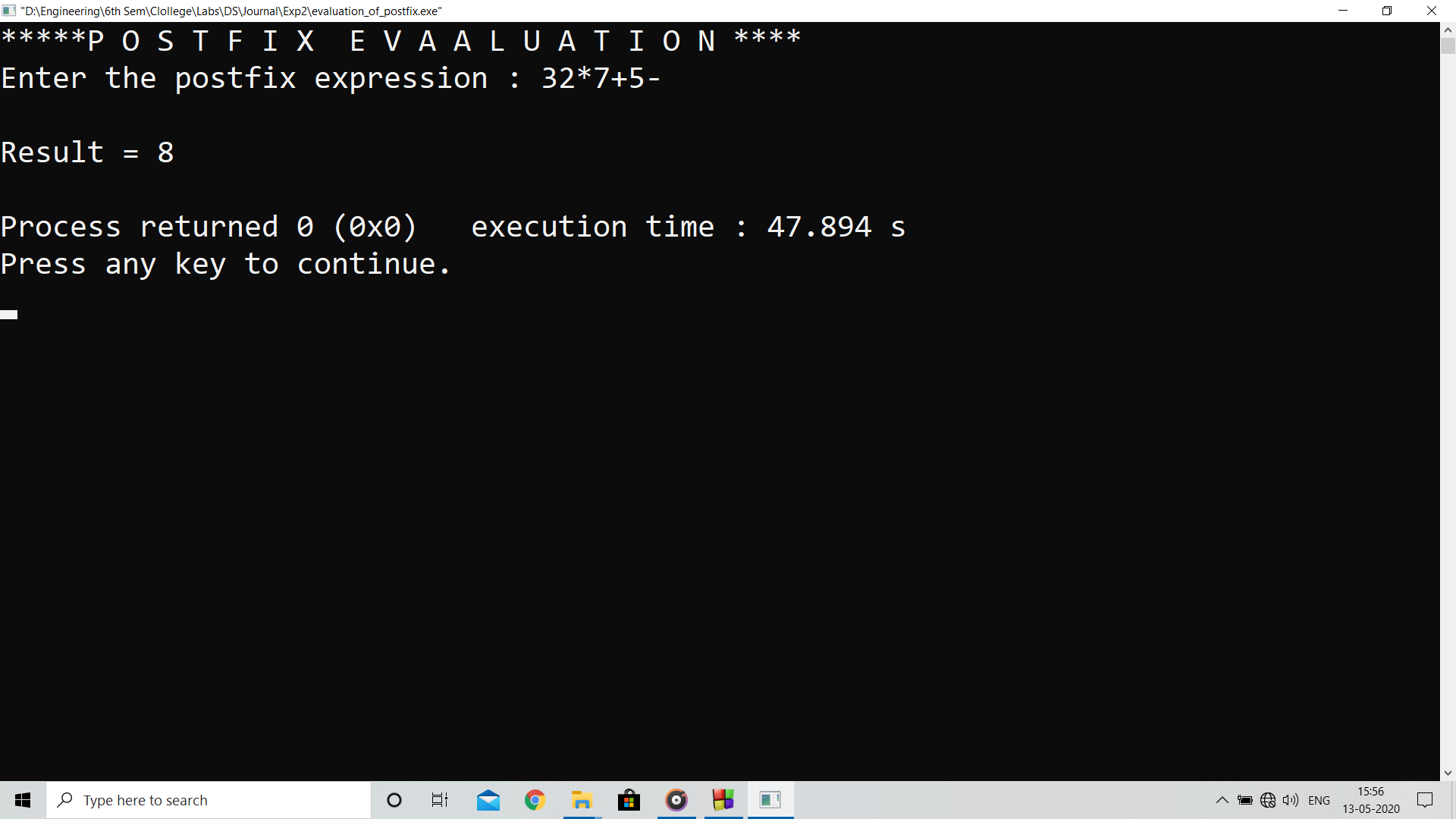
printf("\n");

evaluate();//evaluate the entered expression

printf("Result = %d\t\n",stack[top]);//display result

}

**Output:**

****

**Analysis:**

The program evaluates postfix expressions entered without parenthesis correctly

**Limitations:**

* Fails to evaluate if parenthesis or power operator are involved
* Only evaluates single digit expressions