**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 along with the size, a top variable to keep a track of the index of stack, a choice variable and 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 ascii()( 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 so the push in stack

**S7 :** If the character is an operator pop twice 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 top = -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)

{

top = top + 1;

stack[top] = k;

}

int pop()

{

int s;

s = stack[top];

top = top -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 : ");

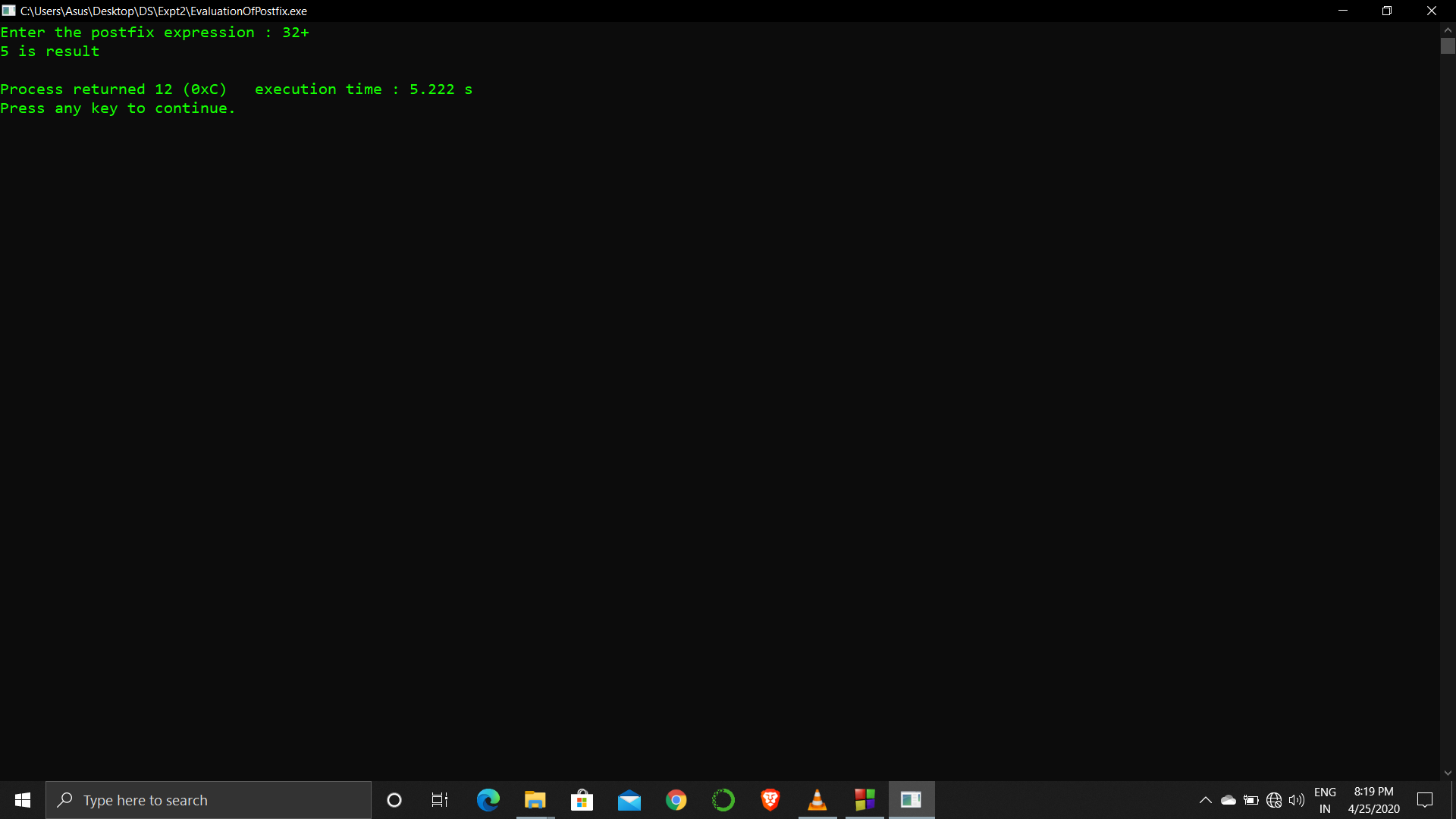
scanf("%s",postfix);

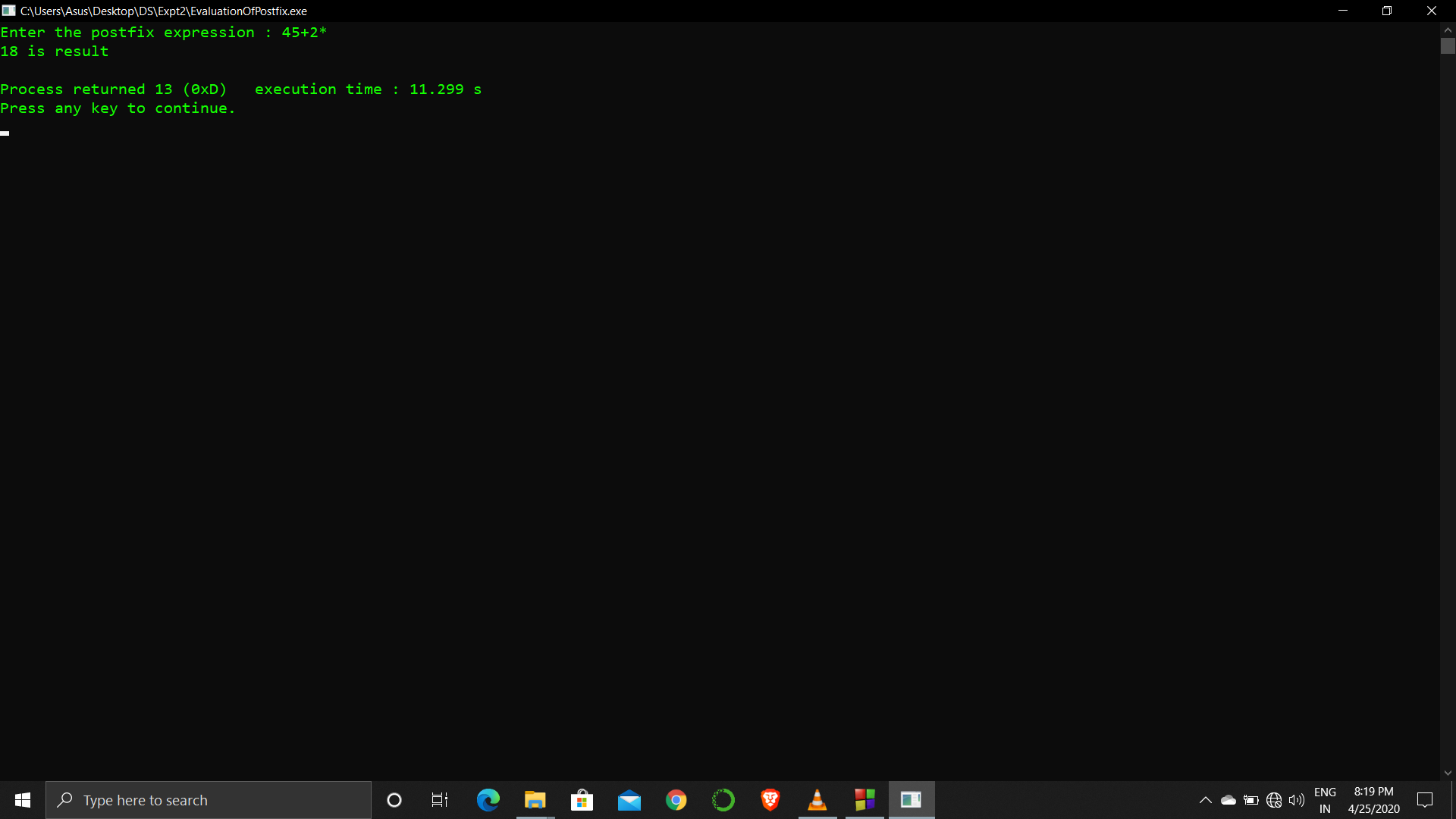
evaluate();

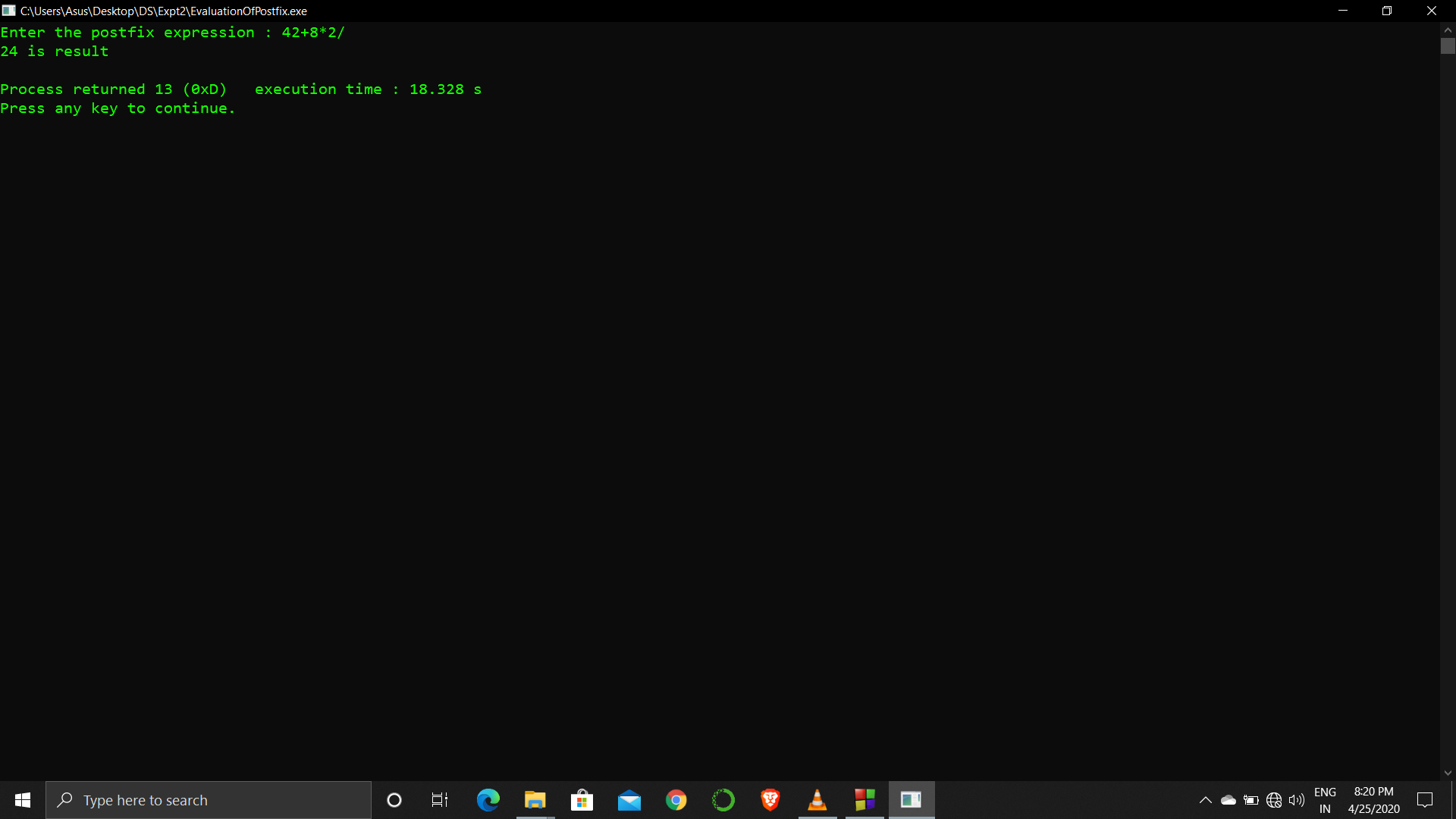
printf("%d is result\n",stack[top]);

}

**Output :**

****

****

****

**Analysis :**

The program takes postfix expression from user and if any operation is not mentioned to be operated then it is unable to process.

The numbers processed in the expression have to be single digit only.