Shreyas Sawant
07A-55

AIM: To perform postfix Evaluation using Stack ADT.

THEORY:

Stack:

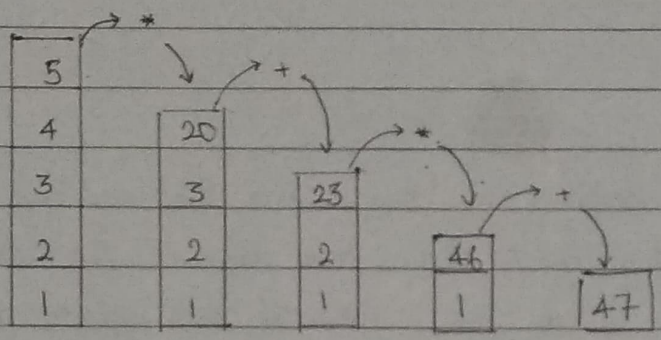
A stack is an ordered collection of items where the addition of new items and the removal of existing items always take place at the same end. This end is commonly referred to as the 'TOP'. The end opposite the top is known as the 'BASE'.

The base of stack is significant since items stored in the stack that are closer to base represent those that have been in the stack of the longest. The most recently added item is the one that is in position to be removed first. This is known as LIFO, last-in-first-out.

Example of Postfix Evaluation:

Expression	Current Character	Action	Stack	Result
12345*+*+)	1	Push it onto the stack	1	
	2	Push it onto the stack	12	
	3	Push it onto the stack	123	
	4	Push it onto the stack	1234	
	5	Push it onto the stack	12345	
	*	Pop off A & B, perform B*A, push on stack	123 20	
	+	Pop off A & B, perform B+A, push on stack	1223	
	*	Pop off A & B, perform B*A, push on stack	146	
	+	Pop of A & B, perform B+A, push on stack	47	
	)			47

Expression = 12345 * + * +



CONCLUSION:

Errors encountered:

1. Incorrect syntax for : #1 define Size = 100

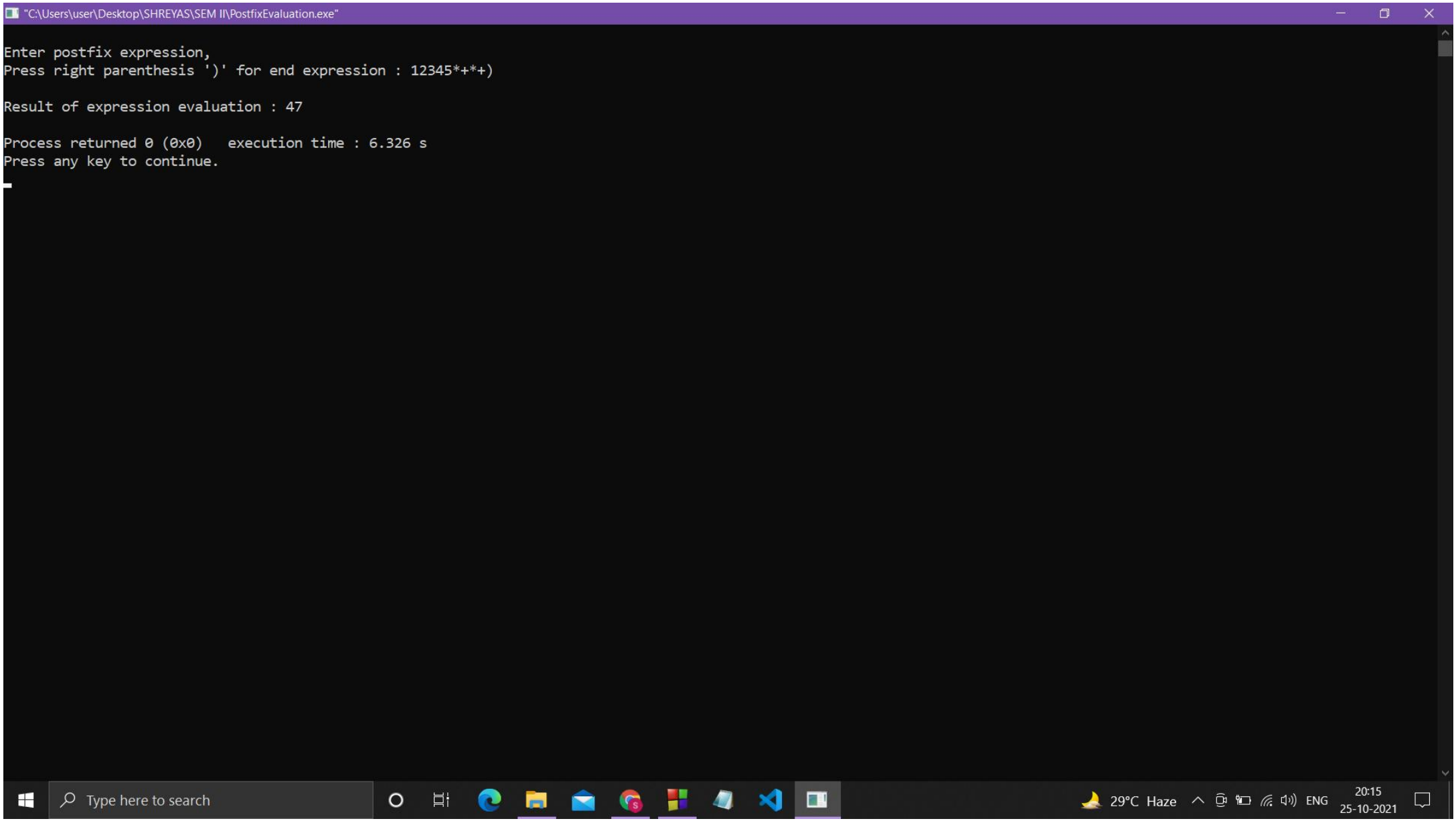
Solution Correct syntax: #1 define Size 100

2. Using assignment operator instead of if statement : if(top = size - 1) .

Solution: Using '==' instead of '=' i.e. if (top == size - 1),

```c
//SHREYAS SAWANT D7A 55
//Postfix Expression using Stack ADT

#include <stdio.h>
#include <ctype.h>
#include <math.h>
#define Size 100

int stack[Size];
int top = -1;

void push(int item)
{
    if (top == Size - 1) {
        printf("Stack over flow");
        return;
    }
    else {
        top++;
        stack[top] = item;
    }
}

int pop()
{
    int item;
    if (top < 0) {
        printf("Stack under flow");
    }
    else {
        item = stack[top];
        top--;
        return item;
    }
}

void EvalPostfix(char postfix[])
{

    int i;
    char ch;
    int val;
    int A, B;


    for (i = 0; postfix[i] != ')'; i++) {
        ch = postfix[i];
        if (isdigit(ch)) {
            push(ch - '0');
        }
        else if (ch == '+' || ch == '-' || ch == '*' || ch == '/') {
            A = pop();
            B = pop();
            switch (ch)
            {
            case '*':
                val = B * A;
                break;

            case '/':
                val = B / A;
                break;

            case '+':
                val = B + A;
                break;

            case '-':
                val = B - A;
                break;
            }
            push(val);
        }
    }
    printf("\nResult of expression evaluation : %d \n", pop());
}
int main()
{

    int i;


    char postfix[Size];
    printf(" \nEnter postfix expression,\nPress right parenthesis ')' for end expression : ");
```

```
85
86
87
88          for (i = 0; i <= Size-1; i++) {
89              scanf("%c", &postfix[i]);
90
91              if (postfix[i] == ')')
92              {
93                  break;
94              }
95          }
96          EvalPostfix(postfix);
97
98          return 0;
99      }
100
```


"C:\Users\user\Desktop\SHREYAS\SEM II\PostfixEvaluation.exe"
Enter postfix expression,
Press right parenthesis ')' for end expression : 12345*+*+)
Result of expression evaluation : 47
Process returned 0 (0x0) execution time : 6.326 s
Press any key to continue.
Type here to search
29°C Haze
ENG
20:15
25-10-2021