|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Agnel Charities’***  **Fr. C. RODRIGUES INSTITUTE OF TECHNOLOGY DEPARTMENT: COMPUTER ENGINEERING**  **LABORATORY CONTINUOUS ASSESSMENT FORMAT**  **Second Half of 2020** | | | | | |
| **Course Name: Data Structures Lab/CSL301** | | | | | |
| **Name of the Teacher: Prof. Kavita Shelke** | | | | | |
| **Name of the Student: Shreya Sushil Sapale** | | | | | |
| **Roll No: 1019154** | | | **Semester: III** | | |
| **Batch: 2** | | | **Practical No: 4** | | |
| **Date of Practical: 5/08/2020** | | | **Date of Report Submission: 17/08/2020** | | |
| **Title:**    **Implement program for Postfix expression evaluation using Stack data structure.** | | | | | |
| **Course Outcome:**    **Develop a program to implement stack data structure and its application.** | | | | | |
| **ASSESSMENT** | | | | | |
| **Sr. No.** | **Parameter for**  **Assessment** | **Marks** | **Rubrics** | | |
| **1.** | **Practical Performance / Active Participation (03Marks)** |  | Above Average  (03) | Average  (02) | Below Average  (01) |
| **2.** | **Report Presentation**  **(02 Marks)** |  | Above Average  (02) | Average  (01) | Below Average  (00) |
| **3.** | **Understanding**  **(03 Marks)** |  | Above Average  (03) | Average  (02) | Below Average  (01) |
| **4.** | **Regularity in Submission**  **(02 Marks)** |  | Timely  (02) | Late  (01)  (≤ 2 Weeks from the date of Practical) | Very Late  (00)  (> 2 Weeks from the date of Practical) |
| **Total Marks (10):** | | | | | |
| **Teacher’s Signature: Date:** | | | | | |

A close up of text on a white background

Description automatically generated

A close up of text on a whiteboard

Description automatically generated

Program **:**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

#include<math.h>

#define max 100

int st[max],top = -1;

void push(int );

int pop ();

void main ()

{

    char exp[100];

    int i=0;

    int op1,op2,val,num;

    printf("\nEnter the postfix expression :\n ");

    scanf("%s",exp);

    do

    {

        if (exp[i]!='+'&& exp[i]!='-'&& exp[i]!='\*'&&exp[i]!='/'&&exp[i]!='^')

        {

           num = exp[i] - '0';

           //printf("\n%d is the int value",num);

           push(num);

        }

        else

        {

            op2=pop();

            op1=pop();

            switch(exp[i])

            {

                case '+':

                {

                    val = op1 + op2 ;

                    //printf("\n %d is the value",val);

                    break;

                }

                 case '-':

                {

                    val = op1 - op2 ;

                    //printf("\n %d is the value",val);

                    break;

                }

                 case '\*':

                {

                    val = op1 \* op2 ;

                    //printf("\n %d is the value",val);

                    break;

                }

                 case '/':

                {

                    val = op1 / op2 ;

                    //printf("\n %d is the value",val);

                    break;

                }

                 case '^':

                {

                    val = pow(op1,op2) ;

                    //printf("\n %d is the value",val);

                    break;

                }

            }

            //out of switch and got the val

            push(val);

        }

        i++;

    } while (i<strlen(exp));

    printf("The solution of the given expression is %d.\n",val);

}

void push ( int c)

{

    if (top == max-1)

    {

        printf("/nStack is Full");

    }

    else

    {

        //printf("\n%d is the pushed int ",c);

        top++;

        //printf("\n%d is the value of top",top);

        st[top]= c;

    }

}

int pop ()

{

    char val ;

    if(top == -1)

    {

        printf("\nStack is empty");

    }

    else

    {

        val=st[top];

        top--;

    }

    return val;

}

Output **:**

**A screenshot of a computer screen

Description automatically generated**