**Pimpri Chinchwad College of Engineering**

**Department of MCA**

**Data Structure Lab**

**Assignment No. 02 : Assignment based on operations on Stack, Application of Stack**

**Date of Submission : 26th February 2021**

1. Write a menu driven program to perform push and pop operations on stack

**Solution :**

#include <stdio.h>

#include <stdlib.h>

#define SIZE 10

int arr[SIZE];

int c,i,n,top=-1;

int main()

{

while(1){

printf("\n\n ===== MENU =====");

printf("\n 1.Push");

printf("\n 2.Pop");

printf("\n 3.Exit");

printf("\n Enter your choice : ");

scanf("%d",&c);

switch(c){

case 1 : Push();

break;

case 2 : Pop();

break;

case 3 : exit(0);

break;

default : printf("\n Invalid Choice");

break;

}

}

return 0;

}

void Push(){

printf("\n Enter the Element : ");

scanf("%d",&n);

arr[++top] = n;

printf("\n Element Pushed!");

display();

}

void Pop(){

if(top==-1)

printf("\n Stack is Empty!");

else{

--top;

printf("\n Element Pop!");

display();

}

}

void display(){

printf("\n\n Stack Elements are : ");

for(i=0;i<=top;i++){

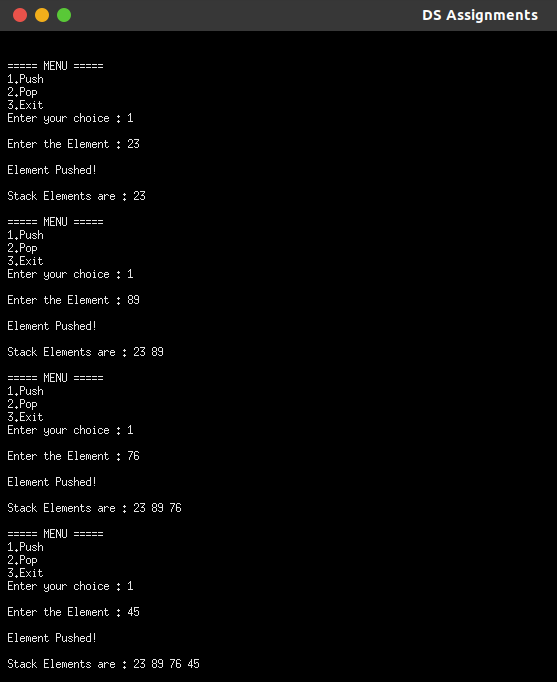
printf("%d ",arr[i]);

}

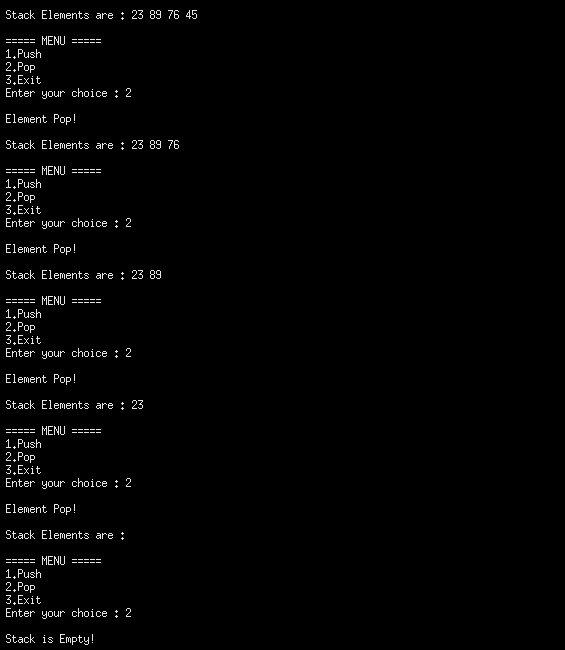
}

**Output :**

**1 . Push the Elements in the Stack :**

****

**2 . Pop the Element from Stack :**

****

1. Write a program to convert infix expression to postfix expression

**Solution :**

#include<stdio.h>

#define SIZE 100

int isOperand(char);

void push(char);

char pop();

int prec(char);

char stack[SIZE];

int top=-1;

int main(){

char infix[SIZE];

int i;

printf("\n Enter an Arithmetic Experession : ");

scanf("%s",infix);

printf("\n Postfix Expression : ");

for(i=0;infix[i]!='\0';i++){

if(isOperand(infix[i])==1)

printf("%c",infix[i]);

else

if(infix[i]=='(')

push(infix[i]);

else

if(infix[i]==')'){

while(stack[top]!='('){

printf("%c",pop());

}

pop();

}

else{

while(prec(infix[i])<=prec(stack[top]))

printf("%c",pop());

push(infix[i]);

}

}

while(top!=-1)

printf("%c",pop());

return 0;

}

int isOperand(char ch){

if((ch>='a' && ch<='z') || (ch>='A' && ch<='Z') || (ch>='0' && ch<=9))

return 1;

else

return 0;

}

void push(char ch){

stack[++top]=ch;

}

char pop(){

return stack[top--];

}

int prec(char ch){

switch(ch){

case '^' : return 3;

case '\*' :

case '/' : return 2;

case '+' :

case '-' : return 1;

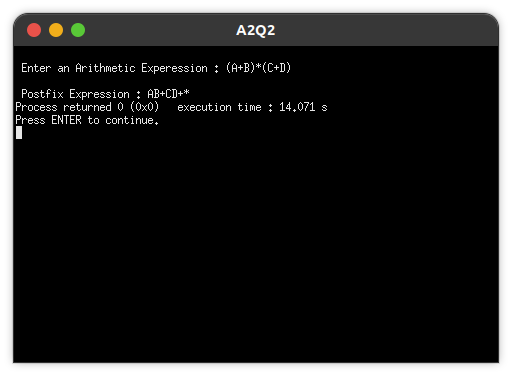
default : return 0;

}

}

**Output :**

****

****