Roll NO:27

NAME : Mali Anjali Prakash

DIV: FYMCA-A

Assignment no -4

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

#include <string.h>

#define MAX 10

int stack[MAX];

char infix[MAX];

char postfix[MAX];

int top = -1;

void push(char c)

{

   if( top==MAX-1)

   {

    printf("full");

   }

   top++;

   stack[top]=c;

}

int pop()

{   char c;

   if(top==-1)

   {

    printf("is isempty");

   }

   c = stack[top];

   top--;

   return c;

}

int isEmpty()

{

   if(top== -1)

      {

        return 1;

      }

      else{

        return 0;

      }

}

void print()

{   int i=0;

     while(postfix[i])

     {

        printf("%c",postfix[i++]);

     }

  printf("\n");

}

int precedents(char c)

{

   switch(c)

   {

    case '^':

    return 3;

            case '\*':

            case '/':

            return 2;

            case '+':

            case '-':

            return 1;

            default:

            return 0;

   }

}

void intopost()

{   int j=0;

    char symbol,next;

    for(int i=0;i<strlen(infix);i++)

    {

        symbol=infix[i];

        switch(symbol)

        {

            case '(':

                 push(symbol);

                 break;

            case ')':

                     while((next=pop())!='(')

                     postfix[j++]=next;

                     break;

            case '^':

            case '\*':

            case '/':

            case '+':

            case '-':

                     while(!isEmpty()&&precedents(stack[top])>=precedents(symbol) )

                     postfix[j++]=pop();

                     push(symbol);

                break;

                default:

                postfix[j++]=symbol;

        }

    }

    while(!isEmpty())

    {

        postfix[j++]=pop();

        postfix[j]='\0';

    }

}

int evlpost()

{

     int a,b;

    for(int i=0;i<strlen(postfix);i++)

    {

        if(postfix[i]>='0'&&postfix[i]<='9')

           push(postfix[i]-'0');

           else{

            a=pop();

            b=pop();

            switch(postfix[i])

            {

                case '+':

                push(b+a);

                break;

                case '-':

                push(b-a);

                break;

                case '\*':

                push(b\*a);

                break;

                case '/':

                push(b/a);

                break;

                case '^':

                push(pow(b,a));

                break;

            }

           }

    }

    return pop();

}

void main()

{   int result=0;

    printf("enter any experssion \n ");

scanf("%s",infix);

     intopost();

     result=evlpost();

     print();

     printf("the ans is %d\n",result);

}

