Ds lab[28-12-23]

Lab2

Infix to postfix

Code:-

#include<stdio.h>

#include<ctype.h>

#define SIZE 50

char stack[SIZE];

int top=-1;

push(char elem)

{

stack[++top]=elem;

}

char pop()

{

retrun(stack[top--]);

}

int pr(char symbol)

{

if(symbol == '^')

{

return(3);

}

else if(symbol== '\*' ||symbol=='/')

{

return(2);

}

else if(symbol =='+' || symbol =='-')

{

return(1);

}

else

{

return(0);

}

}

void main()

{

char infix[50],postfix[50],ch,elem;

int i=0,k=0;

printf("enter infix expression");

scanf("%s",infix);

push('#');

while((ch=infix[i++])!='\0')

{

if(ch=='(') push(ch);

else

if(isalnum(ch)) postfix[k++]=ch;

else

if(ch==')')

{

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

postfix[k++]=pop();

elem=pop();

}

else

{

while(pr(stack[top])>=pr(ch))

postfix[k++]=pop();

push(ch);

}

}

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

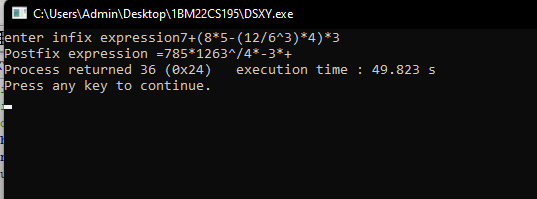
postfix[k++]=pop();

postfix[k]='\0';

printf("Postfix expression =%s",postfix);

}

Output:-



Prgm:- postfix evaluation

Code:-

#include<stdio.h>

int stack[20];

int top=-1;

void push(int x)

{

stack[++top]=x;

}

int pop()

{

return stack[top--];

}

int main()

{

char exp[20];

char \*e;

int n1,n2,n3,num;

printf("Enter experssion ::");

scanf("%s",exp);

e=exp;

while(\*e !='\0')

{

if(isdigit(\*e))

{

num=\*e -48;

push(num);

}

else

{

n1=pop();

n2=pop();

switch(\*e)

{

case '+':

{

n3=n1+n2;

break;

}

case '-':

{

n3=n2-n1;

break;

}

case '\*':

{

n3=n2\*n1;

break;

}

case '/':

{

n3=n2/n1;

break;

}

}

push(n3);

}

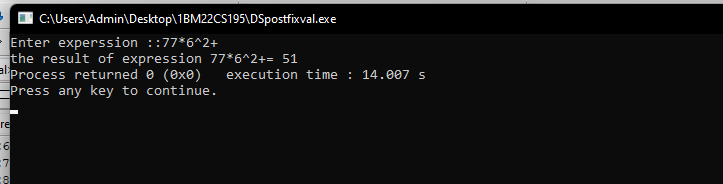
e++;

}

printf("the result of expression %s= %d",exp,pop());

}

Output:-



Lab:-3a

Code:-

#include<stdio.h>

#define MAX 50

int queue\_array[MAX];

int rear=-1;

int front=-1;

display()

{

int i;

if(front==-1)

printf("queue is empty\n");

else

{

printf("queue is :\n");

for(i=front;i<=rear;i++)

printf("%d",queue\_array[i]);

printf("\n");

}

}

main()

{

int choice;

while(1)

{

printf("1.insert\n");

printf("2.delete\n");

printf("3.display\n");

printf("4.exit\n");

printf("enter your choice:");

scanf("%d",&choice);

switch(choice)

{

case 1:

insert();

break;

case 2:

delete();

break;

case 3:

display();

break;

case 4:

exit(1);

break;

default:

printf("invalid choice\n");

}

}

}

insert()

{

int add\_item;

if(rear==MAX-1)

printf("queue overflow\n");

else

{

if(front==-1)

front=0;

printf("insert the element in the queue:");

scanf("%d",&add\_item);

rear+=1;

queue\_array[rear]=add\_item;

}

}

delete()

{

if(front==-1 || front>rear)

{

printf("queue underflow\n");

return;

}

else

{

printf("deleted element is : %d\n",queue\_array[front]);

front+=1;

}

}

OUTPUT:-

