

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<math.h>
#include<ctype.h>
#define MAX 50

struct postfix
{
    int stack[MAX];
    int top,nn;
    char *s;
};

void initpostfix(struct postfix *);
void setexpr(struct postfix *,char *);
void push(struct postfix *,int);
int pop(struct postfix *);
void calculate(struct postfix *);
void show(struct postfix);

void main()
{
    struct postfix q;
    char expr[MAX];
    clrscr();
    initpostfix(&q);
    printf("\n _____");
    printf("\n EVALUATION OF POSTFIX EXPRESSION");
    printf("\n -----");
    printf("\n DON'T GIVE SPACE INBETWEEN NUMBERS OR OPERATORS");
    printf("\n Enter two single digit numbers and a operator\n");
    printf("\n Enter an expression:");
    gets(expr);
    setexpr(&q,expr);
    calculate(&q);
    show(q);
    getch();
}

void initpostfix(struct postfix *p)
{

```

```
p->top=-1;
}
```

```
void setexpr(struct postfix *p,char *str)
{
    p->s=str;
}
```

```
void push(struct postfix *p,int item)
{
    if(p->top==MAX-1)
        printf("\n Stack is full");
    else
    {
        p->top++;
        p->stack[p->top]=item;
    }
}
```

```
int pop(struct postfix *p )
{
    int data;
    if(p->top== -1)
    {
        printf("Stack is empty");
        return NULL;
    }
    data=p->stack[p->top];
    p->top--;
    return(data);
}
```

```
void calculate(struct postfix *p)
{
    int n1,n2,n3;
    while(*(p->s))
    {
        if(isdigit(*(p->s)))
        {
            p->nn=*(p->s)-'0';
            push(p,p->nn);
        }
        else
```

```
{
n1=pop(p);
n2=pop(p);
switch(*(p->s))
{
case '+':
    n3=n2+n1;
    break;
case '-':
    n3=n2-n1;
    break;
case '*':
    n3=n2*n1;
    break;
case '/':
    n3=n2/n1;
    break;
case '%':
    n3=n2%n1;
    break;
default:
    printf("\n Unknown operator");
    getch();
    exit(0);
}
push(p,n3);
}
p->s++;
}
```

```
void show(struct postfix p)
{
p.nn=pop(&p);
printf("\n Result is %d",p.nn);
}
```

-->SAMPLE INPUT AND OUTPUT:

---

EVALUATION OF POSTFIX EXPRESSION

---

DON'T GIVE SPACE INBETWEEN NUMBERS OR OPERATORS

Enter two single digit numbers and a operator

Enter an expression:23\*

Result is 6

---

EVALUATION OF POSTFIX EXPRESSION

-----  
DON'T GIVE SPACE INBETWEEN NUMBERS OR OPERATORS

Enter two single digit numbers and a operator

Enter an expression:57-

Result is -2

---

EVALUATION OF POSTFIX EXPRESSION

-----  
DON'T GIVE SPACE INBETWEEN NUMBERS OR OPERATORS

Enter two single digit numbers and a operator

Enter an expression:45%

Result is 4