#include<stdio.h>

#include<stdlib.h>

typedef enum{lparin,rparin,plus,minus,times,divide,mod,eos,operand}precedence;

int x=1;

int top=-1;

char expr[100];

typedef struct

{

int key;

}term;

term \*element;

void create()

{

element=(term\*)malloc(sizeof(element));

}

void stackfull()

{

x=x\*2;

element=realloc(element,x\*sizeof(element));

}

int pop()

{

if (top==-1)

{

return -1;

}

else{

return element[top--].key;

}

}

void push(int item)

{

if (top>=(x-1))

{

stackfull();

element[++top].key=item;

}

else

{

element[++top].key=item;

}

}

precedence gettoken(char\*symbol,int\*n)

{

\*symbol=expr[(\*n)++];

switch(\*symbol)

{

case '(':return lparin;

case ')':return rparin;

case '+':return plus;

case '-':return minus;

case '/':return divide;

case '\*':return times;

case '%':return mod;

case '\0':return eos;

default:return operand;

}

}

int eval()

{

precedence token;

char symbol;

int op1,op2;

int n=0,top=-1;

token=gettoken(&symbol,&n);

while(token!=eos)

{

if(token==operand)

{

push(symbol-'0');

}

else{

op2=pop();

op1=pop();

switch(symbol)

{

case '+':push(op2+op1);break;

case '-':push(op2-op1);break;

case '/':push(op2/op1);break;

case '\*':push(op2\*op1);break;

case '%':push(op2%op1);break;

}

}token=gettoken(&symbol,&n);

}return(pop());

}

void main()

{

printf("Enter the Post Fix expression\n");

gets(expr);

create();

printf("%d",eval());

}