#include <stdio.h>

#include <malloc.h>

struct node

{

int \*base;

int \*top;

};

void push(struct node \*L,int x)

{

\*L->top=x;

L->top++;

}

int pop(struct node \*L)

{

int x;

L->top--;

x=\*L->top;

return x;

}

int change(char b[])

{

int j,x=0;

for(j=0;b[j]!='\0';j++)

x=x\*10+b[j]-'0';

return x;

}

int main()

{

char a[10000];

char b[8];

int i=0,j,flag;

gets(a);

struct node \*L;

L=(struct node \*)malloc(sizeof(struct node));

L->base=(int \*)malloc(10000\*sizeof(int));

L->top=L->base;

int x,t,p;

while(a[i]!='\0')

{

while(a[i]==' ') i++;

flag=0;

for(j=0;a[i]!=' ';j++,i++)

{

if(a[i]=='/') {flag=1;i++;break;}

else if(a[i]=='\*') {flag=2;i++;break;}

else if(a[i]=='+') {flag=3;i++;break;}

else if(a[i]=='-') {flag=4;i++;break;}

else if(a[i]=='@') {flag=5;break;}

else b[j]=a[i];

}

b[j]='\0';

if(j!=0)

{

x=change(b);

push(L,x);

}

if(flag==1||flag==2||flag==3||flag==4)

{

t=pop(L);

p=pop(L);

if(flag==1) x=p/t;

else if(flag==2) x=p\*t;

else if(flag==3) x=p+t;

else if(flag==4) x=p-t;

push(L,x);

}

if(flag==5) break;

}

printf("%d",\*L->base);

}

**问题 L: 后缀表达式求值**

时间限制: 1 Sec 内存限制: 128 MB  
提交: 1788 解决: 761  
[[提交](http://acm.hnust.cn/JudgeOnline/submitpage.php?cid=1569&pid=11&langmask=0)][[状态](http://acm.hnust.cn/JudgeOnline/problemstatus.php?id=1824)][[讨论版](http://acm.hnust.cn/JudgeOnline/bbs.php?pid=1824&cid=1569)]

**题目描述**

为了便于处理表达式，常常将普通表达式（称为中缀表示）转换为后缀{运算符在后，如X/Y写为XY/表达式。在这样的表示中可以不用括号即可确定求值的顺序，如：(P+Q)\*(R－S) → PQ+RS－\*。后缀表达式的处理过程如下：扫描后缀表达式，凡遇操作数则将之压进堆栈，遇运算符则从堆栈中弹出两个操作数进行该运算，将运算结果压栈，然后继续扫描，直到后缀表达式被扫描完毕为止，此时栈底元素即为该后缀表达式的值。

**输入**

输入一行表示后缀表达式，数与数之间一定有空格隔开（可能不只一个空格），最后输入@表示输入结束。

数据保证每一步的计算结果均为不超过100000的整数。

**输出**

输出一个整数，表示该表达式的值.

**样例输入**

14 3 20 5 / \*8 - + @

**样例输出**

18