#include<iostream>

#include<cstring>

using namespace std;

string prec\_plus(string plus\_s1,string plus\_s2){

int plus\_i1[10100],plus\_i2[10100];

int l1=plus\_s1.length(),l2=plus\_s2.length();

string ans="";

int len=max(l1,l2);

memset(plus\_i1,0,sizeof(plus\_i1));

memset(plus\_i2,0,sizeof(plus\_i2));

for(int i=l1-1;i>=0;i--)

plus\_i1[l1-i-1]=plus\_s1[i]-'0';

for(int i=l2-1;i>=0;i--)

plus\_i2[l2-i-1]=plus\_s2[i]-'0';

for(int i=0;i<len;i++){

plus\_i1[i]+=plus\_i2[i];

plus\_i1[i+1]+=plus\_i1[i]/10;

plus\_i1[i]%=10;

}

if(plus\_i1[len]!=0) len++;

while(plus\_i1[len-1]==0 and len>1)

len--;

for(int i=len-1;i>=0;i--)

ans=ans+char(plus\_i1[i]+'0');

return ans;

}

string prec\_minus(string minus\_s1,string minus\_s2){

int minus\_i1[10100],minus\_i2[10100];

int l1=minus\_s1.length(),l2=minus\_s2.length();

string ans="";

int len=max(l1,l2);

memset(minus\_i1,0,sizeof(minus\_i1));

memset(minus\_i2,0,sizeof(minus\_i2));

for(int i=l1-1;i>=0;i--)

minus\_i1[l1-i-1]=minus\_s1[i]-'0';

for(int i=l2-1;i>=0;i--)

minus\_i2[l2-i-1]=minus\_s2[i]-'0';

for(int i=0;i<len;i++){

minus\_i1[i]-=minus\_i2[i];

if(minus\_i1[i]<0){

minus\_i1[i]+=10;

minus\_i1[i+1]--;

}

}

while(minus\_i1[len-1]==0 and len>1)

len--;

for(int i=len-1;i>=0;i--)

ans=ans+char(minus\_i1[i]+'0');

return ans;

}

string prec\_multiply(string multiply\_s1,string multiply\_s2){

int multiply\_i1[1010],multiply\_i2[1010],multiply\_i3[1010];

int l1=multiply\_s1.length(),l2=multiply\_s2.length();

string ans="";

int len=(l1+l2);

memset(multiply\_i1,0,sizeof(multiply\_i1));

memset(multiply\_i2,0,sizeof(multiply\_i2));

memset(multiply\_i3,0,sizeof(multiply\_i3));

for(int i=l1-1;i>=0;i--)

multiply\_i1[l1-i-1]=multiply\_s1[i]-'0';

for(int i=l2-1;i>=0;i--)

multiply\_i2[l2-i-1]=multiply\_s2[i]-'0';

for(int i=0;i<l1;i++){

for(int j=0;j<l2;j++){

multiply\_i3[i+j]+=multiply\_i1[i]\*multiply\_i2[j];

multiply\_i3[i+j+1]+=multiply\_i3[i+j]/10;

multiply\_i3[i+j]%=10;

}

}

while(multiply\_i3[len-1]==0 and len>1)

len--;

for(int i=len-1;i>=0;i--)

ans=ans+char(multiply\_i3[i]+'0');

return ans;

}

string prec\_division(string div\_s1,int div\_i2){

int div\_i1[10100];

memset(div\_i1,0,sizeof(div\_i1));

int l1=div\_s1.length();

for(int i=0;i<l1;i++)

div\_i1[i]=div\_s1[i]-'0';

int div\_t=0;

for(int i=0;i<l1;i++){

div\_t=div\_t\*10+div\_i1[i];

div\_i1[i]=div\_t/div\_i2;

div\_t%=div\_i2;

}

bool div\_f=false;

string ans;

for(int i=0;i<l1;i++){

if(div\_i1[i]) div\_f=true;

if(div\_f or i==l1-1) ans=ans+char(div\_i1[i]+'0');

}

return ans;

}