#include <iostream>

#include <string>

#include <math.h>

#include <typeinfo>

using namespace std;

int main()

{ string x;

int type;

cout << "Enter a radix system type in integer form : ";

cin>>type;

cout << "Enter a number : ";

cin>>x;

// cout<<x<<endl;

string front\_radix="";

string back\_radix="";

int j=0;

char i=' ';

while (i!='.' & j!=x.length()){

front\_radix+=x[j];

i=x[j];

j+=1;

}

while (j!=x.length()){

back\_radix+=x[j];

j+=1;

}

int length\_of\_back=back\_radix.length();

if(length\_of\_back){

front\_radix.pop\_back();

}

// cout<<front\_radix<<" "<<back\_radix;

int length\_of\_front=front\_radix.length();

int indx\_for\_front=length\_of\_front-1;

int indx\_for\_back=0;

int front\_base\_10=0;

while(indx\_for\_front>-1){

front\_base\_10+= (((int)front\_radix[indx\_for\_front])-48)\*pow(type,length\_of\_front-indx\_for\_front-1);

indx\_for\_front-=1;

}

// cout<<front\_base\_10<<endl;

double back\_base\_10=0.0;

if(length\_of\_back){

while(indx\_for\_back<length\_of\_back){

back\_base\_10+=(((int)back\_radix[indx\_for\_back])-48)\*pow(type,-(indx\_for\_back+1));

indx\_for\_back+=1;

}

}

// cout<<back\_base\_10<<endl;

double num\_in\_10=0.0;

num\_in\_10=back\_base\_10+front\_base\_10;

cout<<num\_in\_10<<endl;

//convert in to y radix

string f\_b\_10="";

string b\_b\_10="";

j=0;

i=' ';

string n\_i\_10=to\_string(num\_in\_10);

while (i!='.' & j!=n\_i\_10.length()){

f\_b\_10+=n\_i\_10[j];

i=n\_i\_10[j];

j+=1;

}

// cout<<j<<n\_i\_10.length()<<endl;

while (j!=n\_i\_10.length()){

b\_b\_10+=n\_i\_10[j];

j+=1;

}

int length\_of\_back\_base\_10=b\_b\_10.length();

front\_base\_10=stoi(f\_b\_10);

if(length\_of\_back\_base\_10){

back\_base\_10=stoi(b\_b\_10)/(pow(10,length\_of\_back\_base\_10));

}

//cout<<back\_base\_10<<endl;

cout<<"Enter the radix type you want to convert into : ";

int convert\_type=0;

cin>>convert\_type;

string front\_convert="";

string back\_convert="";

int r=0;

while(front\_base\_10){

r=front\_base\_10%convert\_type;

front\_base\_10=front\_base\_10/convert\_type;

front\_convert+=to\_string(r);

}

int rev\_str=0;

int rev\_end=front\_convert.length()-1;

while (rev\_str<rev\_end){

swap(front\_convert[rev\_str],front\_convert[rev\_end]);

rev\_str+=1;

rev\_end-=1;

}

int temp=0;

int max\_limit=0;

// cout<<back\_base\_10<<endl;

if(length\_of\_back\_base\_10){

while(max\_limit<6 && back\_base\_10!=1.0){

back\_base\_10=back\_base\_10\*convert\_type;

if (back\_base\_10>=1.0){

while(back\_base\_10>=1.0){

temp+=1;

back\_base\_10-=1;

}

max\_limit+=1;

back\_convert+=to\_string(temp);

temp=0;

}

else{

back\_convert+=to\_string(temp);

}

// cout<<back\_base\_10<<" ";

max\_limit+=1;

}

}

string new\_radix="";

if (back\_convert.length()){

new\_radix=front\_convert+"."+back\_convert;

}

else{

new\_radix=front\_convert+back\_convert;

}

cout<<new\_radix<<endl;

return 0;

}