#include <iostream>

using namespace std;

class GCD {

public:

static int gcd(int numbers[],int len) {

int gcd=1;

int index=2;

if(len==1){

gcd=numbers[1];

}

if(len>1){

gcd=euclidGcd(numbers[0],numbers[1]);

}

while(index<len){

gcd=euclidGcd(gcd,numbers[index]);

index++;

}

return gcd;

}

static int euclidGcd(int num1,int num2){

int temp=0;

while(num2!=0){

temp=num2;

num2=num1%num2;

num1=temp;

}

num1=num1<0 ? num1 \* (-1):num1;

return num1;

}

};

int main()

{

int numbers[] = {140,130};

int len=sizeof(numbers)/sizeof(numbers[0]);

cout<<"\*\*\* Greatest Common Divisor \*\*\*"<<endl;

cout<<"GCD(Greatest Common Divisor) of N numbers using Associative law and euclid's method"<<endl;

cout<<GCD::gcd(numbers,len)<<endl;

return 0;

}