Decimal Expansion

Problem Statement

Remember how to do long division? Before you try this question you should try dividing 1000 by 13. See how it works? Now, write a recursive method that takes in a numerator, a divisor and a number *n* that prints out the *n*th digit after the decimal place. For example, 1,000/13 = 76.9230769231, so decimal(1000,13,2) should return 2, while decimal(1000,13,1) should return 9.

Input Format

The main method, which is already written, takes in the numerator as an int, the divisor as an int and the *n*th digit as an int.

Output Format

An int, which is the *n*th digit after the decimal place when the numerator is divided by the divisor.

Constraints

$1 \leq numerator \leq 1000$

$1 \leq divisor \leq 1000$

$0 \leq n \leq 10{,}000$

Sample input

1000

17

4

Sample output

5

```
import java.util.Scanner;

public class Solution {
    public static void main(String args[] ){
        Scanner myscanner = new Scanner(System.in);
        int numerator = myscanner.nextInt();
        int divisor = myscanner.nextInt();
        int n = myscanner.nextInt();
        System.out.println(decimal(numerator,divisor,n));
    }

    public static int decimal(int numerator, int divisor, int n){
        if(n==0){
            return((numerator-numerator%divisor)/divisor);
        }
        return (decimal((numerator%divisor)*10,divisor,n-1));
    }
}
```

Compound Interest

Problem Statement

Write a recursive method that takes in a bank account balance x, an interest rate i and a number of years y, and outputs how much the bank account will be worth after y years.For example, if you invest €100 at 2% annual interest, then after year 1 you have €102.00, after year 2 you have €104.04, after year 3 you have €106.12 and so forth.

Input Format
The main method, which is already written, takes in a bank account balance as a double, x, interest percentage rate as a double, i, and the number of years for which it is being invested, y.

Output Format

A double representing how much the bank account is worth after y years, correct to two decimal places.

Constraints

1<=y<=30

0.0%<=i<=100.0%

0<=x<=1000000

Sample input

7

1.3

100

Sample output

109.46

```
import java.util.Scanner;

public class Solution {
    public static void main(String args[] ){
        Scanner myscanner = new Scanner(System.in);
        int y = myscanner.nextInt();
        double i = myscanner.nextDouble();
        double x = myscanner.nextDouble();
        System.out.println(String.format( "%.2f", compound(y,i,x)));
    }

    public static double compound(int y, double i, double x){


        if(y==0){
            return x;
        }
        double factor=(i/100)+1;
        int one=y-1;
        double two=i;
        double three=x;


        return (factor*compound(one, two, three));
    }
}
```

Is it a palindrome? (recursive)

Problem Statement

Write a recursive method that takes in a String and outputs "TRUE" if it is a palindrome and "FALSE" if it is not

Input Format

The main method reads in a String

Output Format

The main method outputs either TRUE or FALSE

Constraints

0<=length(String)<=100

Sample input

hello

Sample output

FALSE

```
import java.util.Scanner;

public class Solution {
    public static void main(String args[] ){
        Scanner myscanner = new Scanner(System.in);
        String mystring = myscanner.nextLine();
        System.out.println(check(palindrome));
    }

    public static double check(String mystring){
```

```
    /* Enter your code here. Define substring below. */


        if(mystring.length()<2){
            return "TRUE";
        }
        if(mystring.charAt(0)!=mystring.charAt(mystring.length()-
1)){
            return "FALSE";
        }
        String substring=mystring.substring(1,mystring.length()-1);



        return (check(substring));
    }
}
```

Greatest Common Divisor (Euclid's Algorithm)

Problem Statement

Euclid's algorithm is the oldest algorithm ever written down. It is named after the ancient Greek mathematician Euclid, who first described it in his Elements around 300 BC. It involves a recursive algorithm for deriving the Greatest Common Divisor (GCD) of a pair of numbers. Let the first number be x and the second number be y. If either x or y is a zero, then the GCD is the other number. Otherwise the GCD of (x, y) is equal to that of (y , x modulo y).

Complete the method that implements Euclid's algorithm.

Input Format

A pair of integers x and y

Output Format

An integer which is the GCD of x and y

Constraints

1<=x<=100000000

1<=y<=100000000

Sample input

16

12

Sample output

4

```
import java.util.Scanner;

public class Solution {

    public static void main(String args[] ){
        Scanner myscanner = new Scanner(System.in);
        long x = myscanner.nextLong();
        long y = myscanner.nextLong();
        System.out.println(GCD(x,y));
    }

    public static long GCD(long x, long y){


        /* Enter your code here. Define one and two below. */



        if(x==0){
            return y;
        }
        if(y==0){
            return x;
        }
        long one=y;
        long two=x%y;



        return(GCD(one,two))
   }
}
```