

# Algorithm Development and Programming Fundamentals

## Termwork

1. The Collatz function is defined for a positive integer  $n$  as follows.

$$f(n) = \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even} \end{cases}$$

We consider the repeated application of the Collatz function starting with a given integer  $n$ , as follows:

$$f(n), f(f(n)), f(f(f(n))), \dots$$

It is conjectured that no matter which positive integer  $n$  you start from, this sequence eventually will have 1 in it.

e.g. If  $n=7$ , the sequence is

1)	$f(7)$	=	22
2)	$f(f(7))$	=	$f(22) = 11$
3)	$f(11)$	=	34
4)	$f(34)$	=	17
5)	$f(17)$	=	52
6)	$f(52)$	=	26
7)	$f(26)$	=	13
8)	$f(13)$	=	40
9)	$f(40)$	=	20
10)	$f(20)$	=	10
11)	$f(10)$	=	5
12)	$f(5)$	=	16
13)	$f(16)$	=	8
14)	$f(8)$	=	4
15)	$f(4)$	=	2
16)	$f(2)$	=	1

Thus if you start from  $n=7$ , you need to apply  $f$  16 times in order to first get 1.

In this question, you will be given a positive number  $\leq 32,000$ . You have to output how many times  $f$  has to be applied repeatedly in order to first reach 1.

	Input	Expected Output
Test Case 1	101	25
Test Case 2	100	25
Test Case 3	2463	208

2. Write a recursive program that inputs a line of characters from the user. The line may contain blanks. It outputs the line with the characters reversed. The input ends with EOF (end of file).

NOTE: You have to use recursion to solve this, and are NOT allowed to use array to store the input!!

**Example:**

**INPUT**

This is easy

**OUTPUT**

ysae si sihT

	Input	Expected Output
Test Case 1	visible	elbisiv
Test Case 2	xyzyx	xyzyx

3. We say that a string 's' is an anagram of another string 't' if the letters in 's' can be rearranged to form 't'.

For example, "butterfly" is an anagram of "flutterby", since a rearrangement of the first word results in the second.

We say that a position 'i' in 's' and 't' match, if 's' is an anagram of 't', and  $s[i] == t[i]$ .

In this question, you will be given two words, 's' and 't'. You have to output the number of matching positions if s is an anagram of t, and -1 if s is not an anagram of t.

#### Input

The input consists of two lines. The first line contains the first string, with length  $\leq 100$  characters. The second line contains the second string, with length  $\leq 100$  characters.

#### Output

If the first string is an anagram of the second string, then output the number of matching positions. Otherwise, print -1.

#### Sample Input 1

```
butterfly
flutterby
```

#### Sample Output 1

```
2
```

#### Sample Input 2

```
home
come
```

#### Sample Output 2

```
-1
```

	Input	Expected Output
Test Case 1	anarchy anerchy	-1
Test Case 2	cyclonepic enolcyccpi	1
Test Case 3	turingmachine turingmachime	-1
Test Case 4	abacbstuvab baabctsuavb	3

4. In a string, a "run" is a substring consisting of consecutive occurrences of the same character. For example, the string "mississippi" contains the following runs - "ss", "ss" and "pp".

In this question, given a string, you have to output the length of the longest run in the string.

#### Input

A string, having length at most 100. The string is guaranteed to have at least one run.

#### Output

The length of the longest run in the string.

#### Sample Input

abbaaacccc

#### Sample Output

4

	Input	Expected Output
Test Case 1	pqrsssspppqqppttttt	5
Test Case 2	pprdfgeerjimcndddgeejkcj jdjsssssrtrthsa	5
Test Case 3	ppqqqyrtdgfdreeennnnnnssg grrjfhg	6

5. In this question, you are given two positive integers  $M$  and  $N$ , where  $M < N$ . You may assume that  $N$  is less than or equal to 100.

The orbit of  $M$  with respect to  $N$  is defined to be the sequence

$$M, (2 \cdot M) \bmod N, (2^2 \cdot M) \bmod N, \dots$$

There are at most  $N$  elements in the sequence, but for some  $M$ , the number of elements in this sequence may be fewer.

You have to output the maximum number of distinct integers in the orbit of  $M$ .

For example, if  $M=5$  and  $N=8$ , then the orbit of 5 with respect to 8 is

$$5, 2 \cdot 5 \bmod 8, 4 \cdot 5 \bmod 8, 8 \cdot 5 \bmod 8$$

which is equal to

$$5, 2, 4, 0.$$

Hence the number of distinct integers in the orbit of 5 is 4.

	Input	Expected Output
Test Case 1	2 5	4
Test Case 2	4 6	2

6. You will be given an  $N \times N$  matrix. You have to determine whether the matrix is a triangular matrix.

The diagonal of the matrix  $M$  of size  $N \times N$  is the set of entries  $M(0,0)$ ,  $M(1,1)$ ,  $M(2,2)$ , ...,  $M(N,N)$ .

A matrix is upper triangular if every entry below the diagonal is 0. For example,

```
1 1 1
0 0 1
0 0 2
```

is an upper triangular matrix. (The diagonal itself, and the entries above and below the diagonals can be zeroes or non-zero integers.)

A matrix is lower triangular if every entry above the diagonal is 0. For example,

```
2 0 0
3 1 0
4 2 2
```

is a lower triangular matrix.

A matrix is triangular if it is either upper triangular or lower triangular or both.

**You may not use arrays for this program.**

### Input

First, you will be given  $N$ , which is the size of the matrix.

Then you will be given  $N$  rows of integers, where each row consists of  $N$  integers separated by spaces.

### Output

If the input matrix is triangular, then print yes. Otherwise, print no.

Sample Test Cases	Input	Output
-------------------	-------	--------



Test Case 1	3 1 0 0 0 1 0 1 1 2	
		yes
Test Case 2	7 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1	
		yes
Test Case 3	7 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1	
		no
Test Case 4	2 1 1 0 1	
		yes

7. Write Program to generate following pattern for input size N.

For N = 5 output is:

```

* * * * * * * * * *
* * * *   * * * *
* * *     * * *
* *       * *
*         *
*         *
* *       * *
* * *     * * *
* * * *   * * * *
* * * * * * * * *

```

8. Write Program to generate following pattern for input size N.

For N=3 output is:

```

*
* 1 *
* 1 2 1 *
* 1 2 3 2 1 *
* 1 2 1 *
* 1 *
*

```

9. Write Program to generate the following pattern for input size N(rows).

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1

```

10. Write a C program to find G.C.D of a Number - N using Recursion.

11. Write a C program to print Fibonacci Series up to N terms using Recursion.