

## Problem A. Fibonacci

Input file: `standard input`  
Output file: `standard output`  
Time limit (C++): 1 second  
Time limit (Java): 2 seconds  
Time limit (Python): 10 seconds  
Memory limit: 64 megabytes  
Java Class Name: `fibonacci`  
Maximum Points Available: 100

From the beginning of time, 1337 hackers have calculated Fibonacci numbers just because they can. Now it is your turn.

The Fibonacci numbers are defined by  $F_1 = 1$ ,  $F_2 = 1$  and  $F_n = F_{n-1} + F_{n-2}$  for  $n > 2$ . The first few are 1, 1, 2, 3, 5, ...

Given the number  $N$ , calculate the  $N$ th Fibonacci number,  $F_N$ .

### Input

An integer  $N$ .

### Output

The integer  $F_N$ .

### Examples

standard input	standard output
1	1
6	8

### Note

In the examples given, the first fibonacci number is 1 and the sixth is 8.

## Problem B. Palindromania

Input file: `input_xx.txt`  
Output file: `output_xx.txt`  
Maximum Points Available: 100

Palindromes are fun! We love palindromes! Unfortunately, sometimes words pretend to be palindromes, but they're not, and we don't like that. Help check whether a given line of text is a palindrome.

A palindrome is a sentence that reads the same both forward and backward, ignoring whitespace.

### Input

A line of text with under 100 characters.

### Output

If the line of text is a palindrome, output `palindrome`. If not, output `not a palindrome`.

### Note

This is an output only problem. That is, you will not submit any code, only answers. The input files are available on your computer, or on the submission system.

### Examples

<code>input_xx.txt</code>	<code>output_xx.txt</code>
<code>racecar</code>	<code>palindrome</code>
<code>bernays</code>	<code>not a palindrome</code>
<code>madam im adam</code>	<code>palindrome</code>

## Problem C. Primes

Time limit (C++):	1 second
Time limit (Java):	2 seconds
Time limit (Python):	10 seconds
Memory limit:	64 megabytes
Java Class Name:	<code>primes</code>
Maximum Points Available:	100

Primes are interesting numbers that are only divisible by themselves and 1. Finding primes is difficult, which is why we are asking you to do it!

### Input

An integer  $N$ .

### Output

The smallest prime number larger than  $N$ .

### Interaction Protocol

This task is interactive. You are thus required to compile your code against a *stub* that will perform the interaction with the grader for you.

You are required to write one function, called `nextPrime`. This function must have the following signature:

```
Python def nextPrime(N)
C++    int nextPrime(int N);
Java   public static int nextPrime(int N)
```

The function must take an integer,  $N$  and return the smallest prime larger than  $N$ .

### Notes

Testing your program is made easier with the supplied stub code and manager.

To compile and test your program, run the script `run.sh` with the name of an input file and the name of your source code. The script will automatically compile your code against the stub, and run it against the grader. It will output your code's responses to the input.

The format of the input file is as follows. Each line of the input file must contain one integer,  $N$  that will be put to your code. The last line must contain the number `-1`. You are supplied with a sample input file.

Thus to test a solution written in java against the sample input, you must run

```
./run.sh primes.java sample.in
```

If your stubs, graders or compilation scripts are corrupted during the contest, you may re-download them from the problem statement on the web interface.