BASIC PROGRAMMING CONTEST

Problem A: Perfect Square

Problem Statement

An integer is a perfect square if it is the square of another integer. For example, 4, 9, and 16 are perfect squares because they are the squares of 2, 3, and 4, respectively.

Write a program that reads an integer `n` and determines if `n` is a perfect square.

Input

A single line containing one integer `n`.

Output

Print `Yes` if `n` is a perfect square, otherwise print `No`.

Constraints

- `-1,000 <= n <= 1,000`

Sample Cases

Input	Output
25	Yes
10	No
0	Yes
-9	No

Problem B: Smallest and Largest Index

Problem Statement

Given a sequence of `n` integers, your task is to find the index of the smallest and largest elements in the sequence. The indices should be 0-based. If there are multiple occurrences of the smallest or largest value, output the index of the first occurrence.

Input

- The first line contains an integer `n`, the number of elements in the sequence.
- The second line contains `n` space-separated integers.

Output

- The first line should be: `Index of largest element: <index>`
- The second line should be: `Index of smallest element: <index>`

Constraints

- `1 <= n <= 100`
- The value of each element will be between `[-1000, 1000]`.

Sample Cases

Input	Output
5 3 8 1 9 4	Index of largest element: 3 Index of smallest element: 2
7 10 20 30 5 30 20 10	Index of largest element: 2 Index of smallest element: 3

Problem C: Conditional Sum

Problem Statement

Given a sequence of `n` integers, your task is to calculate a sum based on a special condition involving the first element of the sequence.

- If the first element is a prime number, calculate the sum of all **even** numbers in the sequence.
- If the first element is not a prime number, calculate the sum of all **odd** numbers in the sequence.

Note: A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.

Input

- The first line contains the integer `n`.
- The second line contains the `n` integers of the sequence.

Output

A single line containing the calculated sum.

Constraints

- `1 <= n <= 100`
- The value of each element will be between `[0, 1000]`.

Sample Cases

Input	Output	Explanation
5 7 2 3 4 5	6	7 is prime, so the sum of even numbers (2+4) is 6.
6 9 1 2 3 4 5	18	9 is not prime, so the sum of odd numbers (9+1+3+5) is 18.

Problem D: The Next Day

Problem Statement

Write a program that reads a valid day and month. Your program must calculate and print the next day and month. The calendar is cyclical and resets annually (i.e., the day after December 31 is January 01).

For the purpose of this problem, you must follow a simplified calendar rule:

- **February will always have 28 days.**

Input

A single line containing two space-separated integers: `day` and `month`.

Output

Print the next day and month in the format `dd/mm`. For example: `01/03`.

Constraints

- `1 <= month <= 12`
- `1 <= day <= maximum number of days in that month` (The input date will always be valid).

Sample Cases

Input	Output
28 02	The next day is: 01/03
31 12	The next day is: 01/01
30 04	The next day is: 01/05
15 06	The next day is: 16/06