

# Write a function

Problem

Submissions

An extra day is added to the calendar almost every four years as February 29, and the day is called a *leap day*. It corrects the calendar for the fact that our planet takes approximately 365.25 days to orbit the sun. A leap year contains a leap day.

In the Gregorian calendar, three conditions are used to identify leap years:

- The year can be evenly divided by 4, is a leap year, unless:
  - The year can be evenly divided by 100, it is NOT a leap year, unless:
    - The year is also evenly divisible by 400. Then it is a leap year.

This means that in the Gregorian calendar, the years 2000 and 2400 are leap years, while 1800, 1900, 2100, 2200, 2300 and 2500 are NOT leap years. [Source](#)

## Task

Given a year, determine whether it is a leap year. If it is a leap year, return the Boolean `True`, otherwise return `False`.

Note that the code stub provided reads from STDIN and passes arguments to the `is_leap` function. It is only necessary to complete the `is_leap` function.

## Input Format

Read *year*, the year to test.

## Constraints

$$1900 \leq \text{year} \leq 10^5$$

## Output Format

The function must return a Boolean value (True/False). Output is handled by the provided code stub.

## Sample Input 0

```
1990
```

## Sample Output 0

```
False
```

## Explanation 0

1990 is not a multiple of 4 hence it's not a leap year.



Contest ends in 1 day 6 hours 21 minutes 33 seconds



Submissions: 1096




Max Score: 50

Rate This Challenge:



[More](#)

Current Buffer (saved locally, editable)  

Python 3   


```
1 n = int(input())
2 if n % 400 == 0:
3     print("True")
4 elif n % 4 == 0 and n % 100 !=0:
5     print("True")
6 else:
7     print("False")

```

```
9 year = int(input())
10 print(is_leap(year))

```

Line: 7 Col: 19

 [Upload Code as File](#) ☐ Test against custom input 

Run Code Submit Code