

Day of the Programmer

 by [_mfv_](#)

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Marie invented a [Time Machine](#) and wants to test it by time-traveling to visit Russia on the [Day of the Programmer](#) (the **256th** day of the year) during a year in the inclusive range from **1700** to **2700**.

From **1700** to **1917**, Russia's official calendar was the [Julian calendar](#); since **1919** they used the [Gregorian calendar](#) system. The transition from the Julian to Gregorian calendar system occurred in **1918**, when the next day after January **31st** was February **14th**. This means that in **1918**, February **14th** was the **32nd** day of the year in Russia.

In both calendar systems, February is the only month with a variable amount of days; it has **29** days during a *leap year*, and **28** days during all other years. In the Julian calendar, leap years are divisible by **4**; in the Gregorian calendar, leap years are either of the following:

- Divisible by **400**.
- Divisible by **4** and *not* divisible by **100**.

Given a year, ***y***, find the date of the **256th** day of that year *according to the official Russian calendar during that year*. Then print it in the format `dd.mm.yyyy`, where `dd` is the two-digit day, `mm` is the two-digit month, and `yyyy` is ***y***.

For example, the given ***year* = 1984**. **1984** is divisible by **4**, so it is a leap year. The **256th** day of a leap year after **1918** is September 12, so the answer is **12.09.1984**.

Function Description

Complete the *dayOfProgrammer* function in the editor below. It should return a string representing the date of the **256th** day of the year given.

dayOfProgrammer has the following parameter(s):

- year*: an integer

Input Format

A single integer denoting year ***y***.

Constraints

- $1700 \leq y \leq 2700$

Output Format

Print the full date of *Day of the Programmer* during year ***y*** in the format `dd.mm.yyyy`, where `dd` is the two-digit day, `mm` is the two-digit month, and `yyyy` is ***y***.

Sample Input 0

```
2017
```

Sample Output 0

```
13.09.2017
```

Explanation 0

In the year ***y* = 2017**, January has **31** days, February has **28** days, March has **31** days, April has **30** days, May has **31** days, June has **30** days, July has **31** days, and August has **31** days. When we sum the total number of days in the first eight months, we get **31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 = 243**. Day of the Programmer is the **256th** day, so then calculate **256 − 243 = 13** to determine that it falls on day **13** of the **9th** month (September). We then print the full date in the specified format, which is `13.09.2017`.

Sample Input 1

```
2016
```

Sample Output 1

```
12.09.2016
```

Explanation 1

Year ***y* = 2016** is a leap year, so February has **29** days but all the other months have the same number of days as in **2017**. When we sum the total number of days in the first eight months, we get **31 + 29 + 31 + 30 + 31 + 30 + 31 + 31 = 244**. Day of the Programmer is the **256th** day, so then calculate **256 − 244 = 12** to determine that it falls on day **12** of the **9th** month (September). We then print the full date in the specified format, which is `12.09.2016`.

Sample Input 2

```
1800
```

Sample Output 2

```
12.09.1800
```

Explanation 2

Since 1800 is leap year. Day lies on 12 September.



[f](#) [t](#) [in](#)


Contest ends in 3 days


Submissions: 4
Max Score: 15

Rate This Challenge:
☆☆☆☆☆

[More](#)

Current Buffer (saved locally, editable)  

Python 3 

1 `#!/bin/python3`

2

3 `import math`

4 `import os`

5 `import random`

6 `import re`

7 `import sys`

8

9 `# Complete the dayOfProgrammer function below.`

10 `def dayOfProgrammer(year):`

11

12 `if __name__ == '__main__':`

13 `fptr = open(os.environ['OUTPUT_PATH'], 'w')`

14

15 `year = int(input().strip())`

16

17 `result = dayOfProgrammer(year)`

18


19 `fptr.write(result + '\n')`

20

21 `fptr.close()`

22

Line: 1 Col: 1

 [Upload Code as File](#)

☐ [Test against custom input](#)

Run Code

Submit Code