



Started on	Wednesday, 10 April 2024, 6:58 PM
State	Finished
Completed on	Wednesday, 10 April 2024, 7:42 PM
Time taken	43 mins 56 secs
Grade	10.00 out of 10.00 (100%)

### Question 1

Correct

Mark 10.00 out of 10.00

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

- $1700 \leq y \leq 2700$

For example:

Input	Result
2017	13.09.2017
2016	12.09.2016

Answer:

Reset answer

```
20 // Special case for the transition year
21 return "26.09.1918";
22 }
23 else if ((year <= 1917 && year % 4 == 0) ||
24          (year > 1918 && (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0))))
25 {
26     // Leap year in both Julian and Gregorian calendars
27     return "12.09." + to_string(year);
28 }
29 else
30 {
31     // Non-leap year in both Julian and Gregorian calendars
32     return "13.09." + to_string(year);
33 }
34 }
35
36 int main()
37 {
38     string year_temp;
39     getline(cin, year_temp);
40
41     int year = stoi(ltrim(rtrim(year_temp)));
42
43     string result = dayOfProgrammer(year);
44
45     cout << result << "\n";
46
47     return 0;
48 }
49
50 string ltrim(const string &str)
```

```

50 string ltrim(const string &str)
51 {
52     string s(str);
53
54     s.erase(
55         s.begin(),
56         find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace))));
57
58     return s;
59 }
60
61 string rtrim(const string &str)
62 {
63     string s(str);
64
65     s.erase(
66         find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
67         s.end());
68
69     return s;
70 }
71

```

	Input	Expected	Got	
✓	2017	13.09.2017	13.09.2017	✓
✓	2016	12.09.2016	12.09.2016	✓

Passed all tests! ✓

► [Show/hide question author's solution \(C++\).](#)

Correct

Marks for this submission: 10.00/10.00.