

Programming Assignment 02

-

Conditions

Instructions

This programming assignment consists of **2 programming exercises**.

You have to:

1. **create** Python files on your computer (be careful of filenames)
2. **edit** them according to the assignment
3. **verify** on your computer that it works (run them and check the output in the shell)
4. **upload** the files to Gradescope (upload directly the .py files, not a .zip file)
5. **check** the autograder report on Gradescope
6. **go back to step 2** if necessary

The autograder will evaluate your code for a few testcases. If some testcases fail, the autograder should show you what is your code output, and what is the expected output.

Note:

The use of concepts that were not yet covered in class yet is prohibited. You cannot use **for** loops, lists, functions, ...

Exercise 1 - Which full moon?

There are (at least) three special types of full moons:

- **Super Moon:** the full moon occurs when the moon is at its closest approach to earth (less than 230,000 km away).
- **Blue Moon:** the second full moon in a calendar month. In other words, any full moon on the 29th, 30th, or 31st of a month.
- **Blood Moon:** a lunar eclipse during a full moon.

Write a program (in the file `exercise1.py`) that ask the user to answer questions and that will print out the type of moon – "Full Moon", "Super Moon", "Blue Moon", "Blood Moon", based on the values of the variables below. Note that for the moon to be any of these special kinds of moons, it must also be full.

```
phase = "Full"  
distance = 228000  
date = 1  
eclipse = False
```

Note, though, that multiple modifiers can be true at the same time. We could have a Super Blue Moon, a Blue Blood Moon, or even a Super Blue Blood Moon. Always print those modifiers in that order. If any of these special modifiers is present, do not include the word "Full". If none of them are present, but the moon is Full, then print "Full Moon". If none of them are present at all, print "Moon".

Sample example 1 (the user input is in red, the printed output is in blue):

```
Is the moon full? If yes enter Full if no enter No: Full  
Enter the distance of the moon from the earth: 230000  
Enter the date of the month: 1  
Is there an eclipse? Enter True for yes and False for no: False  
Full Moon
```

Sample example 2:

```
Is the moon full? If yes enter Full if no enter No: No  
Enter the distance of the moon from the earth: 240000  
Enter the date of the month: 8  
Is there an eclipse? Enter True for yes and False for no: False  
Moon
```

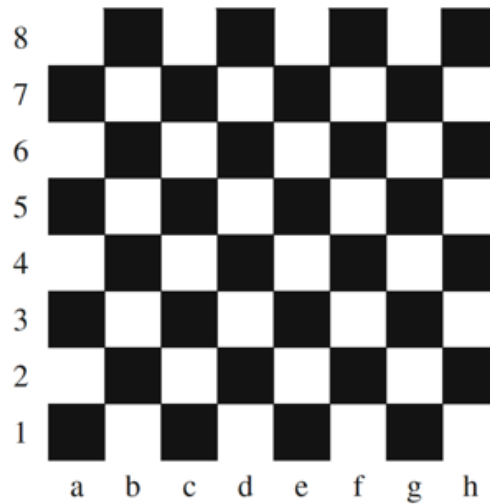


Sample example 3:

```
Is the moon full? If yes enter Full if no enter No: Full
Enter the distance of the moon from the earth: 228000
Enter the date of the month: 30
Is there an eclipse? Enter True for yes and False for no: True
Super Blue Blood Moon
```

Exercise 2 - Chess board

Positions on a chess board are identified by a letter and a number. The letter identifies the column, while the number identifies the row, as shown below:



Write a program (in the file `exercise2.py`) that reads a position from the user. Then use modular arithmetic to report the color of the square in that row. Your program may assume that a valid position will always be entered. It does not need to perform any error checking.

Sample example 1 (the user input is in red, the printed output is in blue):

```
Enter a chess board position: d5
This case is white
```

Sample example 2:

```
Enter a chess board position: g7
This case is black
```

Exercise 3 - Leap year and soccer

Write a program (in the file `exercise3.py`) that does the following (in the specified order):

1. asks the user to **input a year** (type: `int`)
2. **prints the message** `Leap year` or `Not leap year` on a first line
3. then, **only** if applicable, **prints the message** `World Cup year` or `Euro Cup year` on a second line

Definitions

Definition of **leap year** (also known as an intercalary year or bissextile year):

- Every year that is **exactly divisible by 4** is a **leap year**, ...
 - **except** for years that are **exactly divisible by 100**, ...
 - but these **centurial years** are **leap years** if they are **exactly divisible by 400**

For example, the years 1700, 1800, and 1900 were not leap years, but the years 1600 and 2000 were.

Definition of **World Cup year** (FIFA Soccer World Cup):

- Happens **every 4 years**
- Starting **from 1950**

For example, the years 1950, 1982, or 2018 are World Cup years, but the years 1946 or 2020 are not.

Definition of **Euro Cup year** (UEFA Soccer European Championship):

- Happens **every 4 years**
- Starting **from 1960**

For example, the years 1960, 1984, or 2020 are Euro Cup years, but the years 1956 or 2018 are not.

Sample examples (the user input is in **red**, the printed output is in **blue**):

Year: **2000**
Leap year
Euro Cup year

Year: **1956**
Leap year

Year: **2017**
Not leap year