Assignment 03
Due: Check Moodle

General Assignment Notes. Please read carefully.

Your assignments must follow the following requirements:

- Name your project with:
  - your login,
  - an underscore,
  - 'a' (for 'assignment'),
  - the assignment number: *login_a#.*

- For example, if a student with login *barn4520* submits Assignment 1, the name should be: *barn4520_a1* or *barn4520_a01*.
-
- In your program, use the variable naming style given in Coding and Documentation Style https://peps.python.org/pep-0008/

- Zip the entire project. Give your .zip file the same name as your project when exporting your project, e.g. barn4520_a1.zip .

- Unless otherwise indicated by the question you may only use the built-in functions and special forms introduced in the lecture slides.

- The solutions you submit must be entirely your own work. Do not look up either full or partial solutions on the Internet or in printed sources.

- Where a sample run of the program is provided, the underlined values are entered by the user.

General Marking Expectations

- Although the marking scheme is tailored for each question, you can use the following as an indication of what we are looking for when marking your assignments.
  - General:
    - project and zip files named correctly. Using the wrong names is an automatic zero.
    - uses the variable naming style given in Coding and Documentation Style Standards : i.e. lower case variable names, underscores between words
  - main:
    - identification template included and filled in correctly
    - inputs as required for program
    - outputs as required for program

Task 01:
Write and test a program named **t01.py** to divide the number of balloons evenly among children coming to a birthday party. The program should ask for the number of balloons and the number of children joining the party in that order. Output the number of balloons each child will receive and the number of balloons leftover that won't be distributed. You cannot use conditional statements in this task.

Sample Run:
```
Enter number of balloons: 400
Enter number of children: 11

Balloons per child: 36
Balloons left over: 4
```

Task 02:
Write a python program **t02.py** that asks the user to enter a positive three-digit integer N. The program outputs the product of the three digits. If the user enters a number `251`, the product will be `10` as `(2x5x1 = 10)`.
You must use integer division and modulus, not strings, to extract the three digits. You cannot use conditional statements in this task.

Sample Run:
```
Enter a positive three digit integer: 251

The product of the three digits in the integer (251) is: 10
```

Task 03: Write and test a program **t03.py** to calculate charges for a shipping company.

The Fast Freight Shipping Company charges the following rates:

Weight of Package	**Rates Per Pound**
2 pounds or less	$1.50
Over 2 pounds but not more than 5 pounds	$3.00
Over 5 pounds but not more than 11 pounds	$4.00
Over 11 pounds	$4.75
Priority shipping charges	The rate is multiplied by 1.5

Sample Run of the program:
```
Enter weight of the package: 1
Is it a priority package(Y/N): N
The cost for your non-priority package is:   $1.50
```

Task 04:
Write and test a program t04.py that asks the user to enter a number of seconds and works as follows:

- There are 60 seconds in a minute. If the number of seconds entered by the user is greater than or equal to 60, the program should convert the number of seconds to minutes and seconds.
- There are 3,600 seconds in an hour. If the number of seconds entered by the user is greater than or equal to 3,600, the program should convert the number of seconds to hours, minutes, and seconds.
- There are 86,400 seconds in a day. If the number of seconds entered by the user is greater than or equal to 86,400, the program should convert the number of seconds to days, hours, minutes, and seconds.

Sample Run of the program:

```
Enter time in seconds:   65
The time is:             1 min(s) and 5 sec(s)
```

Task 05:
Write and test a program t05.py to display the number of days in February for a given year.

The month of February normally has 28 days. But if it is a *leap year*, February has 29 days. Write a program that asks the user to enter a year. The program should then display the number of days in February that year. Use the following criteria to identify leap years:

- Determine whether the year is divisible by 100. If it is, then it is a leap year if and only if it is also divisible by 400. For example, 2000 is a leap year, but 2100 is not.
- If the year is not divisible by 100, then it is a leap year if and only if it is divisible by 4. For example, 2008 is a leap year, but 2009 is not.

Sample Run of the program:

```
Enter a year: 2008
In the February of 2008, the number of days are 29.
```

Note:
Exact naming of files, functions and variable names is critical.
You cannot use any concept not yet taught in the class.