CS 112 – Fall 2021 – Programming Assignment 2

Decision Making

Due Date: Monday Sept. 20, Noon (12 pm)

The main purpose of this assignment is to gain experience using selection statements effectively. See the **Assignment Basic** file for more detailed information about getting assistance, running the test file, grading, commenting, and many other extremely important things. Each assignment is governed by the rules in that document.

Needed files: download the attached file to use the tests we will use when grading your assignment

- tester2.py

Background

Selection statements (**if/elif/else** combinations) allow us to write code that can execute different statements based on the current values seen in a particular run of the program. We will use this to write a program to create a "tentative" title for the movie of your life.

Guidelines

- Think carefully about the order you will check different properties. You might want to write some pseudocode or perhaps draw a flowchart if this works better for you. Think first, then implement.
- Be careful what kinds of selection statements you use and be sure to test your code with many examples. For instance, multiple if statements will not necessarily behave the same as a chain of **if-elif-else**.
- When a specific test case is not working, plug your code into the visualizer to watch what the code does, which lines run, which branches are taken.
- From built-in functions, you are allowed to call **abs()**, **int()**, **float()**, **str()** only if needed.
- You are not allowed to import anything.
- You are not allowed to use loops, lists, sets, dictionaries and any feature that hasn't been covered in class yet.
- Insert comments in the lines you deem necessary

Assumptions

You may assume that the types of the values that are sent to the functions are the proper ones.

Testing

In this assignment testing will be slightly different from the previous one. The tester you'll be provided with uses "unit testing" with a large number of tests per function. A template with the functions is provided to you. The tester will be calling your functions with certain arguments and will be examining the return values to decide the correctness of your code. Your functions should not ask for user input and should not print anything, just return a value based on the specification of the tasks. When you run your code with the tester or submit your code to **Gradescope**, you can see the results of the tests.

Note: You **DO NOT** need to modify the tester, just test on your own any way you like!

Grading Rubric

Submitted correctly (file is named correctly)	2
Code is well commented	5
Math operation* in the `months()` function	3
Correct outputs:	40
TOTAL:	**50**

Note:
If your code does not run (immediately crashes due to errors), it will receive at most 12 points. No exceptions. As stated on our syllabus, turning in running code is essential.

*The math operation implemented in the `months()` function must determine the correct number of months for any number of `days>=1` even for the cases where `days>729`

Reminders from the **Assignment Basics** file:

- Don't forget to name your file **netID_2xx_PA2.py**, where **2xx** is your lab section number.
- Don't forget your honor code statement at the top of your **.py** file
- Don't forget to comment your code
- No hard coding!
- Submit to **Gradescope** under Programming Assignment 2

Task

You are trying to create a fun application to have fun with your friends. The app can create a movie title based on some input from a user.

Functions

In this assignment things are a bit different from the previous one. You won't write a monolithic program as you did in Programming Assignment 1, but you're going to implement only functions.

Be reminded that the generic structure of a function is the following:

```
def function_name(arg1, arg2, etc.):  #This line is the signature
      # commands go here
      # more commands
      # result = ...
      return result           #This line returns the result of your computations
```

The signature of each function is provided below, do ***not*** make any changes to them otherwise the tester will not work properly. Keep in mind that you must ***not*** write a main body for your program in this assignment. You should only implement these functions; it is the tester that will be calling and testing each one of them.

The following are the functions you have to implement:

```
def myMovieLife(lastname_fstletter, birth_month, cell_digit):
```

Description: This function creates a "*Title of the Movie of your Life*", which is a string formed by the concatenation of return value (rv) of the call of the following 5 functions.

Function name	Return Value
status	True/False
adjective	string
subject	string
complement	string
months	int

Parameters: **lastname_fstletter** (string) is the first letter of your last name, **birth_month** (string) is the first 3 letters of your birth month, **cell_digit** (int) is the last digit of your cellphone.

Return value: A composed string with the return value of the call of 5 functions as follows:
"The" + rv status + rv adjective + rv subject + rv complement + "in"+ rv months + **M**

Where **M** is "month" when is 1; and "months" when is >1
Note: place a blank space after each string except the last one (M)

Examples:

```
myMovieLife('A',"Jan",0) →
      "The True awesome biography of an adventurer in 1 month"

myMovieLife('Z',"Dec",9) →
      "The False overwhelming fairy tale of an ogre in 25 months"
```

```
def status(cell_digit):
```

Description: This function gets the last digit of your cellphone and returns a boolean (True/False).
Parameters: **cell_digit** (int) is the last digit of your cellphone.

Return value: It returns True for values: 0,2,4,6,8; OR False for values: 1,3,5,7,9

Examples:

```
status(0) → True
status(7) → False
```

```
def adjective(lastname_fstletter):
```

Description: This function gets the first letter (Uppercase) of your last name and returns a string which is an adjective for your "Title Movie".

Parameters: **lastname_fstletter** (string) is the first letter of your last name

Return value: A string based of the following values:

A	awesome	J	wonderful	S	astonishing
B	shocking	K	dramatic	T	interesting
C	hilarious	L	intriguing	U	unexpected
D	fascinating	M	courageous	V	surprising
E	marvelous	N	beautiful	W	lovely
F	unbelievable	O	bracing	X	electrifying
G	funny	P	lively	Y	commoving
H	epic	Q	dangerous	Z	overwhelming
I	thrilling	R	impressive		

Examples:

```
adjective('P') → "lively"
adjective('D') → "fascinating"
```

```
def subject(birth_month):
```

Description: This function gets the birth month and returns a string which is a subject for your "Title Movie".

Parameters: **birth_month** (string) is the first 3 letters of your birth month.

Return value: A string based of the following values:

Jan	biography	Jul	mission
Feb	history	Aug	existence
Mar	legend	Sep	battle
Apr	life	Oct	chronicle
May	anecdote	Nov	combat
Jun	revenge	Dec	fairy tale

Examples:

```
subject("Jan") → "biography"
subject("Dec") → "fairy tale"
```

```
def complement(cell_digit):
```

Description: This function gets the last digit of your cellphone and returns a string which is a complement for your "Title Movie".

Parameters: **cell_digit** (int) is the last digit of your cellphone.

Return value: A string based of the following values:

0	of an adventurer	5	of a scientific
1	of a warrior	6	of a dreamer
2	of a genius	7	of a cowboy
3	of a movie star	8	of a jedi
4	of a hero	9	of an ogre

Examples:

```
complement(1) → "of a warrior"
complement(8) → "of a jedi"
```

```
def months(cell_digit):
```

Description: This function gets the last digit of your cellphone and returns an integer that will be months used for your "Title Movie", where months >= 1.

First you need to determine a number of days with the following formula:

$$days = (cell\_digit)^{((cell\_digit\%2)+1)} * (cell\_digit)$$

After, you need to transform the number of days to months. Assume a month has 30 days, and return the value based on the range of days. Note that 1 is the minimum number of months.

Range of days	Months
0 – 30	1
31 – 60	2
61 – 90	3
…	…
721 – 750	25
…	…

Parameters: **cell_digit** (int) is the last digit of your cellphone.

Return value: An integer that represents a number of months, where months >= 1.

YOU MUST implement a math operation to determine the number of months, NO hard coding allowed

Examples:

```
months(0) → 1    # 0^1 * 0 = 0 days >> 1 month
months(9) → 25   # 9^2 * 9 = 729 days >> 25 months
```