NYU Tandon School of Engineering

CS-UY 1114 Spring 2023

Homework 01

Due: 11:59pm, Thursday, February 9th, 2023

Submission instructions

1. You should submit your homework on **Gradescope**.
2. For this assignment you should turn in 4 separate `.py` files named according to the following pattern: `hw1_q1.py`, `hw1_q2.py`, etc.
3. Each Python file you submit should contain a header comment block as follows:

```
"""
Author: [Your name here]
Assignment / Part: HW1 - Q1 (etc.)
Date due: 2023-02-09, 11:59pm
I pledge that I have completed this assignment without
collaborating with anyone else, in conformance with the
NYU School of Engineering Policies and Procedures on
Academic Misconduct.
"""
```

No late submissions will be accepted.

REMINDER: *Do not use any Python structures that we have not learned in class.*

For this specific assignment, you may use everything we have learned up to, **and including**, variables, types, mathematical and boolean expressions, user IO (i.e. `print()` and `input()`), number systems, and the `math` / `random` modules. Please reach out to us if you're at all unsure about any instruction or whether a Python structure is or is not allowed.

Do **not** use, for example, selection statements (i.e. `if`, `elif`, `else`), `for`- and `while`-loops, modules, user-defined functions (*except* for `main()` if your instructor has covered it during lecture), strings and string methods, file i/o, exception handling, dictionaries, lists, tuples, and/or object-oriented programming.

Failure to abide by any of these instructions will make your submission subject to point deductions.

Question 01: *Hello, You!*

Write a program that asks for the user’s name, their age and prints a personalized welcome message for them. For example, an execution could look like this:

```
Please enter your name: Aika
Please enter your age: 29
Aika, 29, is taking CS-UY 1114.
```

The format of your output need not look exactly the same as ours, but it must still ask the user for their name and age, and print them out as part of some sort of sentence.

Question 02: *Some Have Gone and Some Remain*

(Textbook P84, Q4): The U.S. Census provides information on its **web page** about the current U.S. population as well as approximate rates of change. Four rates of change are provided:

- There is a birth every 7 seconds.
- There is a death every 15 seconds.
- There is a new immigrant every 42 seconds.
- There is a new emigration (i.e. somebody moves out of the country) every 1.25 minutes.

These are, naturally, approximations of birth, death, and immigration rates, but they can assist in providing population estimates in the near term.

Write a program that takes a year as input (an integer—you may assume that this will value will always be greater than or equal to `2023`) and prints out an estimated population (as an integer). Assume that the current population is 330,109,174, and assume that there are always exactly 365 days in a year.

Sample execution:

```
Please enter a year greater than 2023:
2042
The population in year 2042 will be 382038453
```

The format of your output need not look exactly the same as ours, but it must still ask the user to enter an integer representing a year, and it must always output the estimated population of that year.

Question 03: *Penny Pinching*

Write a program that asks the user to enter a number of quarters, dimes, nickels and pennies and then outputs the monetary value of the coins in the format of dollars and remaining cents.

Your program should interact with the user, and output its results, ***exactly*** as it is shown in the following example:

```
Please enter number of coins:
Number of quarters: 13
Number of dimes: 4
Number of nickels: 11
Number of pennies: 17
The total is 4 dollar(s) and 37 cent(s)
```

Question 04: *Mad as a Hatter, Thin as a Dime*

Write a program that asks the user to enter an amount of money in the format of dollars and remaining cents. The program should calculate and print the minimum number of coins (quarters, dimes, nickels and pennies) that are equivalent to the amount input by the user.

Your program should interact with the user, and display its results, **exactly** as it shows in the following example:

```
Please enter your amount of dollars and cents, in two separate lines.
4
37
4 dollars and 37 cents are: 17 quarters, 1 dimes, 0 nickels and 2 pennies
```

***Hint**: In order to find the minimum number of coins, first find the maximum number of quarters that fit within the amount of money input by the user, then find the maximum number of dimes that fit in the remaining amount, and so on.*