Data Structures
Homework #1

Due: September 30, 2025

This assignment is to practice Python programming. There are four major problems and please finish each problem right after the problem description in `HW1.ipynb` file, provided on the **i-school(Plus)** (`https://istudy.ntut.edu.tw/learn/index.php`) platform. More detail about each problem is also in `HW1.ipynb` file.

1. Write a short Python function `all_tuples()` that, given an unsorted n integer array A and an integer S, prints all distinct *four-element tuples* (*quadruplets*) from A having sum S. For example, suppose $A = [12, 21, 8, 7, 2, 5, 16, 19, 25, 14, 10]$ and integer $S = 34$. Then, the output will be all four-element tuples: (2, 5, 8, 19), (2, 8, 10, 14), (5, 7, 8, 14), and (5, 7, 10, 12).

2. Python's `random` module includes a function `choice` that returns a random element from a non-empty sequence. The `random` module includes a more basic function `randrange`, with parameterization similar to the built-in `range` function, that returns a random choice from the given range. Using only the `randrange` function, implement your own version of the `choice` function. Please run these functions 10, 100, 1000, and 10000 times respectively and provide a simple comparison on the results of these two versions of functions.

3. The *birthday paradox* says that the probability that two people in a room will have the same birthday is more than half, provided n, the number of people in the room, is more than 23. This property is not really a paradox, but many people find it surprising. Design a Python program that can test this paradox by a series of experiments on randomly generated birthdays, which test this paradox for $n = 5, 10, 15, 20, \ldots, 100$. We consider two directions for the test.

   a (Simulation Experiments)
   One way is to do the experiments by simulation. For each value of n, one do the experiments many times, say 1000 times. In each experiment, one can pick one day `bd` in a year first as a birthday to compare with the other $n - 1$ birthdays. All the birthdays are selected randomly, so please use the function `choice(data)` you implement in Problem 2 for selecting the birthdays. If the selected day `bd` is the same with a day in the remaining randomly generated birthdays, then the trial is successful (true); otherwise, it fails (being false). We count the number of successful trials and can get a percentage that two people in a room have the same birthday for n people. Pleasse provide a Python script that performs the simulation experiments.

   b (Combinatorial Computation)
   The other way is to derive a formula using *combinatorial computation* for the probability that two people have the same birthday in a room having n people. Please first derive the formula and the provide the Python script. Then, use this formula to have the script for the computation and list the results.

4. A *polygon* is a piecewise-linear, closed *curve* in the plane; *i.e.*, a curve ending on itself that is formed by a sequence of straight line segments, called *sides* (or *edges*) of the polygon. The curve is also called the *boundary*. A point joining two consecutive sides is call a *vertex* of the polygon. If a polygon is *simple*, the curve does not cross itself. We consider simple polygons

here. Furthermore, if a polygon is *convex*, then for any two vertices, the line segment drawn between these two vertices is inside or on the boundary. Develop a class, `Polygon`, where each polygon object maintains two instance variables, which we name as: `_numOfSides`, and `_listOfVertices`. With a polygon, please provide a python function `isConvex(Polygon)` to check if the given polygon is convex.

Homework Submission

- Please upload the completed `.ipynb` file with the filename as `HW1_studentID.ipynb` to **i-school(Plus)** (`https://istudy.ntut.edu.tw/learn/index.php`).
- The **deadline** is the **midnight of September 30**, 2025 and **Late work** is not acceptable.
- Honest Policy: We encourage students to discuss their work with the peer. However, each student should write the program or the problem solutions on her/his own. Those who copy others work will get 0 on the homework grade.