**Q4:**

- This program contains an example of a Python dictionary which contains list and tuples.
- The purpose of the dictionary is to store information about an apartment building.
- The program demonstrates various operations that can be performed on a python dictionary and lists and tuples embedded inside of the them.
- This program uses the PrettyPrint library. If it is not on your system, then the program will not run.

**Q5:**

- "HW1-Q5.py" file contains a function definition called "find_median" which takes two arrays as arguments and finds the median between them.
- "HW1-Q5-TESTCODE.py" is the testbench for the "find_median" function.
    - It imports the "find_median" function so the "HW1-Q5.py" file must be in the same directory as the testbench code.
    - When run, the testbench code will prompt you to enter how many tests you want to run.
    - Each test uses a pseudo-random number generator to generate one large array of random length (0-2000 items) and randomly populates it with integers in the range $-10^6$ to $10^6$. This array is sorted and its median is determined.
    - The generated array is then split into two sorted arrays.
    - The sorted arrays are passed to the "find_median" function.
    - The median determined by the "find_median" function is compared to the known value of the median determined when the test array was generated.
    - The test details and the result of the test, "PASS" or "FAIL", is printed to the console.

**Q6:**

The method used in the program for swapping variables uses plus and minus operators and is based on the code under the "Method 1 (Using Arithmetic Operators)" section at the following website: https://www.geeksforgeeks.org/swap-two-numbers-without-using-temporary-variable/

When this method for swapping variables would fail:

1. When the variables are of data types that the plus and minus operators do not work on, e.g. strings.
2. If both variables are very large (e.g. very large integers), adding them together could cause overflow. Thus, the original values of the two variables would be lost and the swap would not work.

**Q7:**

First, the program prompts the user to input the length of the array they would like to enter.

To accommodate for variable length arrays, I used dynamic memory allocation. The code for collecting input and performing dynamic memory allocation was taken from the answer to this question on Stack Overflow: https://stackoverflow.com/questions/2539796/c-reading-multiple-numbers-from-single-input-line-scanf

The user is then prompted to enter the items of the array. The user should type one integer per line in the terminal and then press enter each time.

The program then prints out a new array of the same length, where each item is a product of all the other integers in the inputted array, not including the integer at the current index.