CSC1001: Introduction to Computer Science Programming Methodology
Assignment 1

Contact: xiaoyuanliu@link.cuhk.edu.cn

Assignment description

This assignment will be worth **10%** of the final grade.

You should write your code for each question in a `.py` file (please name it using the question name, e.g., `q1.py`). Please pack all your `.py` files into a single `.zip` file, name it using your student ID (e.g., if your student ID is `123456`, then the file should be named as `123456.zip`), and then submit the `.zip` file via Blackboard.

Please also write **one text file** providing necessary details about how to run your code for each question (this file is not graded). The text file should be included in the `.zip` file as well.

Please note that the teaching assistant may ask you to explain the meaning of your program to ensure that the code is indeed written by yourself. Please also note that we may check whether your program is too similar to your fellow students' code using Blackboard.

Deadline: This assignment is due at **11:59 PM, 17 October**. For each day of late submission, you will lose **10%** of your mark in this assignment. If you submit more than three days later than the deadline, you will receive **zero** in this assignment.

Question 1 (10% of this assignment)

Write a program that prompts the user to enter a binary number (a string consisting only of 0s and 1s), and then converts it into its hexadecimal representation.

Sample Run:

```
>>> Enter a binary number: 101101
2D
```

Question 2 (10% of this assignment)

Write a program that prompts the user to enter a *positive* integer, and then calculates the sum of its digits.

Sample Run:

```
>>> Enter an positive integer: 3125
The sum of the digits of 3125 is: 11
```

Question 3 (20% of this assignment)

Write a program that implements a 24h → 12h **Time Converter**.

1. Prompts the user for two integers:
   - hour in 24-hour format (0–23)
   - minute (0–59)
2. If either input is out of range, print exactly:

`Invalid time`

3. Otherwise, display the time in 12-hour format as:

   `The time is: H:MM AM or The time is: H:MM PM`

   where:

   - H is 1–12 without a leading zero.
   - MM is 00–59 with a leading zero if needed.
   - Midnight case. Input: `00:MM` → Output: `12:MM AM`.
   - Noon case. Input: `12:MM` → Output: `12:MM PM`.

Sample Runs:

```
>>> Enter hour (0-23): 0
>>> Enter minute (0-59): 0
The time is: 12:00 AM

>>> Enter hour (0-23): 12
>>> Enter minute (0-59): 5
The time is: 12:05 PM

>>> Enter hour (0-23): 23
>>> Enter minute (0-59): 59
The time is: 11:59 PM

>>> Enter hour (0-23): 24
>>> Enter minute (0-59): 0
Invalid time
```

Question 4 (20% of this assignment)

Write a program that prompts the user to enter a positive integer n and return the largest prime factor of n.

Sample Run:

```
>>> Enter an positive integer: 97
The largest prime factor of 97 is: 97
```

Question 5 (20% of this assignment)

Write a **function** named `largest_k_evens(x, k)` that takes a list of numbers `x` and an integer `k`, and returns the k largest even integers in ascending order. If there are fewer than k even integers, return all available evens. Ignore non-integers and booleans.
Note that in this program, you do not need to prompt users to input. We will send various inputs to test your function, so you just need to test your function yourself before submitting.

Sample Run:

```
>>> largest_k_evens([10, 23, 5, 4, 11, 8], 3)
[4, 8, 10]
```

```
>>> largest_k_evens([12, 12, 11, 10, 14], 3)
[12, 12, 14]

>>> largest_k_evens([3.5, 2, 2.0, 6, "8", True], 2)
[2, 6]
```

Question 6 (20% of this assignment)

Write a **function** named `trapezoid_area(points)` that accepts a list of four (x, y) coordinate pairs of a trapezoid, where each pair is a two-element tuple. The order of the points in the list is arbitrary. The function should return the area as a floating-point number. *Hint:* find the parallel sides first.

Sample Run:

```
>>> trapezoid_area([(0,0), (4,0), (3,2), (1,2)])
7.0
```

You do not need to handle improper inputs (e.g., the coordinates do not form a trapezoid, or there are not exactly four points). You are **not allowed** to use any external packages such as `math` or `numpy`; all calculations must be implemented directly using basic arithmetic operations and built-in functions.