MH1403 Algorithms and Computing
Lab 1 Python Programming
(Week 4, 30.01.2023 – 03.02.2023)

Submission Instructions:

1. This lab is 4% of the final grade of this course.
2. The submission deadline is 11:59PM, 6 February (Monday).
3. Please submit the codes task5.py (2 marks) and task6.py (2 marks) in NTULearn (click Lab 1 in NTULearn → MH1403 → Labs, you will find the submission instructions).

There are six tasks. You need to submit the codes of Task 5 and 6.

Task 1. Write a Python program to get an integer n from the user, then print the integers from 1 to n to the screen. Write the code in the file task1.py (you can simply assume that the user will type an integer on the keyboard).

Task 2. Write a Python program to get an integer n from the user, then create a list with n elements. The value of the element at the index i is the integer 3*i+7. Print this list to the screen. Write the code in the file task2.py

Task 3. Write a Python program to get an integer n from the user, then print a square. Each side of the square consists of n characters '*'. For example, if n = 4, the following square is printed:

```
****
*  *
*  *
****
```

Write the code in the file task3.py

Task 4. Define a python function f(x) that returns the sum of all the positive integers which are smaller than x and are a multiple of 11 or 17.

Get an integer n from the user, then call the function f(n) to compute the sum of all the positive integers which are smaller than n and are a multiple of 11 or 17. Print the returned value (the sum) to the screen.

Write the code in the file task4.py

(For example, if n is 18, the sum is 28. If n is 23, the sum is 50.)

Hint. To get an integer n from the user, we may us the following code:

```
n = input('Please input a number: ')
n = int(n)
```

Task 5. (2 marks) Triangle class
Create a class named Triangle. The class has five instance variables a, b, c, peri, and valid. a, b, c are the side lengths, peri is the perimeter of the triangle, valid is True if the three sides can form a triangle; otherwise, it is False. This class Triangle has the following methods:

1. __init__()
   Initialize instance variables a, b, c to the inputs when an object is created (i.e., this constructor will take input parameters, and initialize a, b and c as those inputs).

2. is_valid()
   This method sets the instance variable valid to False if the length of any one side is larger than or equal to the sum of the lengths of another two sides; otherwise, this method set the instance variable valie to True.

3. computePeri()
   This method compute the perimeter of the triangle, then update the instance variable peri.

4. printTriangle()
   It prints all the instance variables of Triangle to the screen.

After creating the Triangle class, create an object triA of Triangle with three input parameters (2.1, 3.4, 5.2). After creating the object triA, call triA.is_valid(), followed by triA.computePeri(), then triA.printTriangle() to print all the instance variables to the screen.

Then create an object triB of Triangle with three input parameters (2, 3, 5). After creating the object triB, call triB.is_valid(), followed by triB.computePeri(),

then triB.printTriangle() to print all the instance variables to the screen.

Write your code in the file task5.py

Task 6. (2 marks) The Python list is a class, and it has a number of methods, such as append(), remove(), reverse(), insert(), etc. A list, such as X = [2, 3, 4], is an object (instance) of the Python list class. Define a class named myList inheriting from Python's list class, and introduce a new method power() in the class myList. When the method power() is called with input parameter x, every element of an object (instance) of myList is raised to the power of x (assume that every element is a number).

After defining the class myList, we execute the following code:

```
B = myList()
for i in range(5):
    B.append(i)

print(B)
B.power(2)
print(B)
B.reverse()
print(B)
```

We get the following output:

```
[0, 1, 2, 3, 4]
[0, 1, 4, 9, 16]
[16, 9, 4, 1, 0]
```

Write your code in the file task6.py (Hint: in the class myList, we can use self[i] to access the ith element)