CSE4IP – Lab 3-2

Q1 – Reading data with sentinel

Write a program that reads one or more marks entered by the user. The marks are between 0 and 100, inclusive. The reading ends when the user enter -999.

The program is simply required to echo prints the marks on the screen. Ignore the issue of input validation for the marks.

Here -999 is known as the sentinel. A sentinel is a value that indicates the end of a data set. It is not part of the data set.

Q2 – Reading data with sentinel

Modify the previous program so that if a mark is outside the valid range, display this message

```
The mark is invalid.  It will be ignored!
```

This choice of handling data entry error, when entering a set of data, is not uncommon in practice.

Q3 – Calculating average, minimum and maximum

Modify the previous program to calculate (and display):

a. The average mark

b. The minimum mark

c. The maximum mark

Q4 – Calculating minimum and maximum (Unrestricted range)

Write a program that reads one or more numbers entered by the user. After each iteration, the user is asked if they want to continue or not.

We use the term "unrestricted range" here to indicate that the numbers can take any value.

The program calculates and displays

a. The minimum number and

b. The maximum of the numbers.

Relevant information:

The `math` module defines `math.inf` as a float value representing *positive infinity*, which is bigger than any int or float number. For example:

```
>>> import math
>>> x = 10**20
>>> x < math.inf
True
```

Similarly, `-math.inf` represents *negative infinity*, and it is smaller than any int or float value. For example (continue previous example):

```
>>> y = -x
>>> y > -math.inf
True
```

Q5 – Listing letters A to Z

Write a program to list the letters from A to Z, on the screen, each on a separate line.

Hint:

```
>>> ord("A")
65
>>> chr(65)
'A'
```

Q6 – Temperature conversion

Let F be a temperature in Fahrenheit. The equivalent degrees in Celsius is given by

$$C1 = (F - 32) \times \frac{5}{9}$$

A quick approximation can be calculated as

$$C2 = (F - 30)/2$$

Write a program that display a table with three columns. The first display degrees in Fahrenheit 20, 25, 30, up to 120, inclusive. The second column displays the equivalent degrees in Celsius calculated by the accurate formula, and the third the equivalent degree in Celsius calculated by the approximate formula.

Sample display:

```
20  -6.7  -5.0
25  -3.9  -2.5
30  -1.1   0.0
35   1.7   2.5
40   4.4   5.0
45   7.2   7.5
50  10.0  10.0
55  12.8  12.5
60  15.6  15.0
```

Q7 – Millionaire Plan

Bob has a plan to be a millionaire. In month 1, he saves 1 dollar. In month 2, he saves 2 dollars. In month 3, he saves 4 dollars, and so on. In general, for each month, he saves twice the amount of the previous month.

Write programs to answer these two questions:

a. How much money will Bob save after 1 year (12 months)?

b. When (after how many months) he will become a millionaire?

Q8 – Factorial

Given a number n,

factorial of n = $1 \times 2 \ldots \times n$

Factorial of n is written in Mathematics as n!

Write a program that reads a positive integer n and calculate factorial of n. Do not use the `factorial` function in the `math` library.

Q9 – Adding up terms in a series

Write programs that add up the first 20 terms of the following series:

a. $S_1 = \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \ldots$

b. $S_2 = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \ldots$

It is well known that the first series approaches value 1 as we add more terms. In contrast, the second series just keeps growing and will exceed any given value if we add enough terms (we say that it tends to ∞)

Q10 – Fibonacci Sequence

Fibonacci sequence is shown below:

1, 1, 2, 3, 5, ...

It can defined as follows:

f(1) = 1

f(2) = 1

f(n+1) = f(n) + f(n-1)

That is,

term 1 is 1

term 2 is 1

term 3 = term 2 + term 1 = 2

term 4 = term 3 + term 2 = 2 + 1 = 3

etc.

The terms of Fibonacci sequence and their positions, for the first 10 terms, are shown in the table below:

n	1	2	3	4	5	6	7	8	9	10
f(n)	1	1	2	3	5	8	13	21	34	55

Write a program to the first N terms of the sequence, where N is entered by the user.

Q11 – Prime number

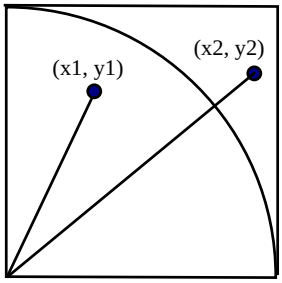
a. Write a program that reads a positive integer N, and count how many proper factor n has.

   An integer n is a proper factor of N if $2 \leq n \leq N - 1$ and N is divisible by n.

b. Write a program that reads a positive integer and determines whether or not it is a prime number.

Q12 – Estimating Pi by Simulation

Consider the diagram below which shows a square of size 1 and a quarter of a circle of radius 1.



Module `random` has method `random` that allows us to generate a random number between 0 and 1. For example:

```
>>> from random import random
>>> random()
0.7740065126012595
>>> random()
0.18210866652361168
```

We can use the `random` function to generate a point inside the square where $0 \leq x < 1$ and $0 \leq y < 1$.

Such a point can be

1. Inside the circle such as (x1, y1) whose distance from the center of the circle is less than 1,
2. Or it can be outside the circle such as (x2, y2), whose distance form the center is greater than 1.

Now, the area of the circle is $A_1 = \frac{\pi}{4}$ and the area of the square is $A_2 = 1$.

Therefore, the *probability* of a point (x, y) to fall inside the circle is equal to the ratio $\frac{A_1}{A_2} = \frac{\pi}{4}$

Hence, we can obtain an estimate of $\frac{\pi}{4}$ by

- generating N points at random,
- counting the number that falls inside the square, and

This method of estimating a certain quantity by simulating a random process is known as the Monte Carlo method.

Then, an estimate of $\frac{\pi}{4}$ can be obtained by

$$\frac{Number\ of\ points\ inside\ the\ circle}{N}$$

and an estimate of π is obtained by

$$4 \times \frac{Number\ of\ points\ inside\ the\ circle}{N}$$

Write a program that generates N point with

```
x = random()
y = random()
```

and uses the points generated to estimate π.

Run your program for various values of N.

Q13 – Convert Number from Base 10 to Base 2, 8 and 16

Write programs to read a positive integers n, and displays on the screen the string representing n in

a. Base 10 (binary format)

   See the provided "Problem Solving Using Python" document.

b. Base 8 (octal format)

c. Base 16 (hexadecimal format)

Q14 – Convert Number from Base 2 to Base 10

Write a program that asks the user for a positive number in binary form, and prints the value of that number in base 10. Sample run:

```
Enter a number in binary form: 110
The value of the number is 6.
```

■