**A blue and black text

Description automatically generated**

**Programming I**

Year 1 (2024/25), Semester 1

## SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in Cybersecurity & Digital Forensics

Diploma in Data Science

Diploma in Information Technology

Diploma in Immersive Media

Common ICT Programme

**COMMON TEST REVISION PAPER**

**INSTRUCTIONS:**

**Prior to test**

* Create a folder on your desktop with your student id as the name
* Create 6 python files in that folder with the following naming convention
  1. CT\_REV\_Q1\_S12345678.py
  2. CT\_REV\_Q2\_S12345678.py
  3. CT\_REV\_Q3\_S12345678.py
  4. CT\_REV\_Q4\_S12345678.py
  5. CT\_REV\_Q5a\_S12345678.py
  6. CT\_REV\_Q5b\_S12345678.py
* In each file, enter your **name**, **id** and **group** in the first line as comment:

e.g. # *John Tan (S12345678) – IT01*

**Submission**

* In POLITEMall, navigate to Assessment > Common Test Revision Submission. Upload all the 6 Python files that you have created and click the “Submit” button.

Note: It is **your responsibility** to ensure that the files are submitted correctly

|  |
| --- |
| **Plagiarism Warning:**  **If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this practical test. Disciplinary action may also be taken.**  **Similar action will be taken for student who allows other student(s) to copy his/her work, or posting any solutions or code related to the practical test before the end of the hour for the test.** |

**QUESTION 1**

Tom is training for a swimming competition and records the time taken for each of 2 laps.

Tom has written a Python program to help him find his best and average times. He has made a couple of mistakes in his program. There may be more than one mistake per line. Help Tom debug and resolve the errors.

The Python program that Tom has created is shown below.

|  |
| --- |
| times = input('Enter time taken of 2 laps separated by semi-colon (seconds):'  times\_list = times.split(';')  firstlap\_time = times\_list[0]  secondlap\_time = times\_list[1]  if firstlap\_time > secondlap\_time:  best = firstlap\_time  else:  best = secondlap\_time  total = firstlap\_time + second\_time  print('Tom's best time is {:.1f} s and average time is {total} s'.format(best)) |

Create a new .py file with IDLE, and copy-paste the code above into the file. Correct the program and submit the corrected file in POLITEMall at the end of the paper.

The following is a sample output of the corrected program (values underlined are the user’s input):

|  |
| --- |
| Enter time taken of 2 laps separated by semi-colon(seconds): 2;3  Tom's best time is 2.0 s and average time is 2.5 s |

**QUESTION 2**

In a card game, the cards 2 to 10 are counted at their face values, regardless of suit (clubs, diamonds, hearts or spades). All face cards (jack, queen and king) are counted as 10. An ace is counted as a 1 or an 11, depending on the total count of all cards in a player’s hand. The ace is counted as 11 only if the resulting total value of all cards does not exceed 21. Otherwise, it is counted as 1.

A program will be used to calculate the total count in a player’s hand. The program will prompt the user to enter three card values, calculate the total value of the hand and display the value of the three cards. An input of 1 corresponds to an ace, an input of 2 corresponds to a two, …, an input of 11 corresponds to a jack, an input of 12 corresponds to a queen, and so on.

Two sample runs of the program are as follows:

|  |
| --- |
| Enter card 1: 5  Enter card 2: 6  Enter card 3: 1  Total value is 12 |

|  |
| --- |
| Enter card 1: 12  Enter card 2: 1  Enter card 3: 1  Total value is 12 |

Lists and other data structures are not allowed for this program.

**QUESTION 3**

In an experiment to find the relationship between a student’s BMI and his t-shirt size, the following data were collected:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Height(cm) | Weight(kg) | Size |
| Sharon | 172 | 59.5 | M |
| Mic | 166 | 65.6 | L |
| Josh | 187 | 49.8 | S |
| Hannah | 200 | 64.2 | M |
| Hanzel | 166 | 47.5 | S |
|  |  |  |  |

(Assume that only 3 sizes: S, M and L are being considered)

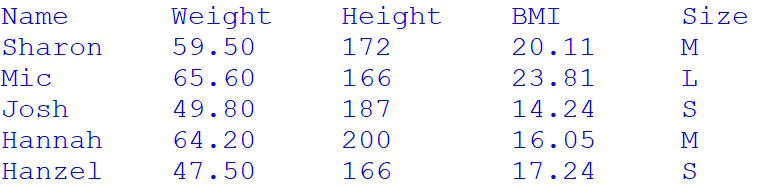
Write a Python program to perform the following tasks:

1. Calculate BMI and print data using lists:

* Create 4 lists (name\_list, height\_list, weight\_list, size\_list) to store the data given in the table, one list for each column.
* Create another new list (bmi\_list), to calculate and store the student’s corresponding BMI.

(The formula for calculating BMI is weight/(height/100)2)

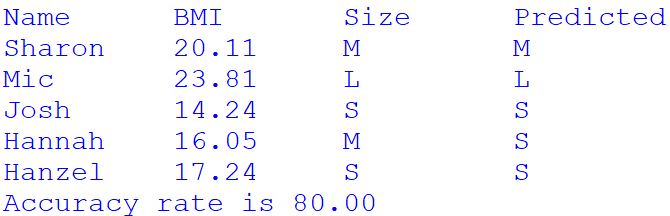
* Print a table using the 5 lists, to present the data as follows:



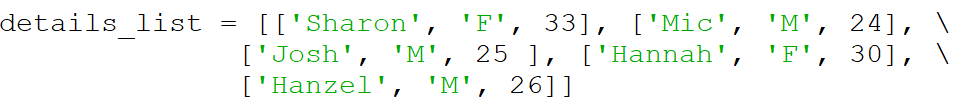
1. It is proposed that the following guideline be used to predict a student’s t-shirt size based on his BMI:

|  |  |
| --- | --- |
| bmi | size |
| Less than or equal to 18 | S |
| Between 18 and 21 inclusive | M |
| More than 21 | L |

* Create a list (predicted\_list), to predict based on the above guideline and store the student’s t-shirt size.
* Print a table using the lists created so far, followed by the accuracy rate of the prediction, which is the percentage of correct predictions, in 2 decimal places.



1. Assuming that a nested list, containing the following information (name, gender, age) was given.



Given weight, height, gender and age, the BMR (Basal Metabolic Rate) of a student can be calculated:

BMR for Men = 66.47 + (13.7 \* weight) + (5 \* height) − (6.8 \* age)

BMR for Women = 655.1 + (9.6 \* weight) + (1.8 \* height) − (4.7 \* age)

Print a table using the lists created in part (a) and the details\_list to present the following information.



**QUESTION 4**

Triangular Numbers are a series of numbers that can be represented in the form of a triangular grid of points where the first row contains a single element and each subsequent row contains one more element than the previous one as shown in the figure below:

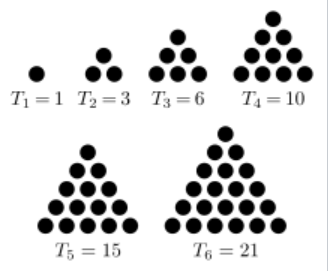


Figure 1: the first 6 triangular numbers (1, 3, 6, 10, 15, 21)

The *n*th triangular number is the number of dots in the triangular arrangement where *n* is the number of dots on the base of the triangle. The formula to calculate the nth triangular number, Tn is:

Tn = ((n\*\*2) + n)/2

Your task is to write a Python program that:

1. Ask the user to enter a number between 0 and 5000.
2. If the number entered is a triangular number, display the value of n
3. Otherwise, display the message that the number entered is not a triangular number

You are required to use loop(s) in your program.

Sample outputs:





**QUESTION 5**

(a) Tom has the text file “marks.txt” that contains some students’ names and their marks for 3 tests, separated by ‘;’ as shown below. He would like to find the average of the 3 tests marks for each student as well as the overall average for the whole class.

|  |
| --- |
| Name;Test1;Test2;Test3  David Lim;80;85;85  Peter Tan;90;98;89  Asli Fan;67;50;45  Simon Chan;77;70;80 |

Write a Python program to read the data from the file and store the values read in a nested list called **test\_marks**. Calculate the average mark for each student and append into the sub-list for each student. Print each student’s name, test marks and average mark. Print also the overall average for all students as shown in sample output below.

|  |
| --- |
| Name Test 1 Test 2 Test 3 Average  ---- ------ ------ ------ -------  David Lim 80 85 85 83.33  Peter Tan 90 98 89 92.33  Asli Fan 67 50 45 54.00  Simon Chan 77 70 80 75.67  Average Mark: 76.33 |

(b) After looking at the output produced in Question 5(a), Tom would like to ensure that it is able to sort the names in alphabetical order.

Tom learnt that he can use the following statement to perform this sorting:

**test\_marks.sort(key=lambda x:x[0], reverse=False)**

Modify your previous program to produce the following output:

|  |
| --- |
| Name Test 1 Test 2 Test 3 Average  ---- ------ ------ ------ -------  Asli Fan 67 50 45 54.00  David Lim 80 85 85 83.33  Peter Tan 90 98 89 92.33  Simon Chan 77 70 80 75.67  Average Mark: 76.33 |

\*\* END OF PAPER \*\*\*