Viva 2 : Methods
Sem 1 2024/25

WIX1002: FOP
Thursday Group

1. P
2. YJ
3 S-J
4
5
6

INSTRUCTIONS :

1. Complete all questions in your designated project group.
2. All members must contribute to writing the codes. (i.e. 1 question = 1 person, and share the workload if there's an additional question relative to the actual number of members in your team (i.e. 5)). Ensure that all members must understand and explain codes from any of the questions.
3. During viva, all students in each team will be randomly asked to describe, answer and edit any of the answers provided. Marks will be given to your ability to present the answers.

Lab Report

Prepare a report to solve the above problems. The report should contain all the sections as below for each question:

Section	Description
1. Problem	Description on the problem
2. Solution	Explanation on how to solve the problems below
3. Sample Input & Output	A few sets of input and output (snapshot)
4. Source code	Java source code

Requirements
1. Group Assignment (Grouping is the same as your project group)
2. Cover page that includes all student matric number and full name.
3. Font: Times New Roman 12, Line Spacing: 1 ½ Spacing
4. Submit to Spectrum according to your OCC. **Deadline** : Before your viva session (W7).

Question 1: Banking System

Problem Statement:

Write a Java program that implements a simple banking system capable of performing basic account operations. The system should allow users to:

1. Check their current balance.
2. Deposit money into their account.
3. Withdraw money from their account.
4. View their transaction history.

Method Details:

1. **`checkBalance()`:** Create a method that prints the current balance to the user.

2. **`deposit(double amount)`**: Create a method that takes an amount as a parameter, adds it to the balance, and records the deposit in the transaction history.
3. **`withdraw(double amount)`**: Create a method that takes an amount as a parameter, subtracts it from the balance (if there is enough balance), and records the withdrawal in the transaction history.
4. **`printTransactions()`**: Create a method that prints all the previous transactions (both deposits and withdrawals).

Sample Output:

```
Welcome to the Bank!
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. View Transaction History
5. Exit
Choose an option: 2
Enter amount to deposit: 200
Your balance is now: 1200.00

1. Check Balance
2. Deposit Money
3. Withdraw Money
4. View Transaction History
5. Exit
Choose an option: 1
Current balance: 1200.00

1. Check Balance
2. Deposit Money
3. Withdraw Money
4. View Transaction History
5. Exit
Choose an option: 3
Enter amount to withdraw: RM300
Your balance is now: RM 900.00

1. Check Balance
2. Deposit Money
3. Withdraw Money
4. View Transaction History
5. Exit
Choose an option: 4
Transaction History:
Deposited: RM200.00
Withdrew: RM300.00

1. Check Balance
2. Deposit Money
3. Withdraw Money
4. View Transaction History
5. Exit
Choose an option: 5

Thank you for using our banking system!
Your final balance is: RM900.00
```

Question 2 : Area Shape Calculator

Problem Statement

Write a Java program that calculates the area of different shapes using **methods**. The program should include separate methods to calculate the area of a **circle**, a **rectangle**, and a **triangle**. The user should be prompted to input the necessary dimensions for each shape, and the program should call the appropriate method to compute the area.

Requirements:

- **Method 1**: `calculateCircleArea(double radius)` – This method should take the radius as input and return the area of the circle. The formula for the area of a circle is `π * radius^2`.
- **Method 2**: `calculateRectangleArea(double length, double width)` – This method should take the length and width as input and return the area of the rectangle. The formula for the area of a rectangle is `length * width`.
- **Method 3**: `calculateTriangleArea(double base, double height)` – This method should take the base and height as input and return the area of the triangle. The formula for the area of a triangle is `(base * height) / 2`.

Instructions:

1. Create methods as described above.
2. Ask the user which shape they would like to calculate the area for (circle, rectangle, or triangle).
3. Based on the user's choice, prompt for the necessary dimensions and call the appropriate method.
4. Print the area of the chosen shape.

Sample Output:

```
Choose the shape to calculate the area:
1. Circle
2. Rectangle
3. Triangle
Enter your choice: 1
Enter the radius of the circle: 5
The area of the circle is: 78.54

Choose the shape to calculate the area:
1. Circle
2. Rectangle
3. Triangle
Enter your choice: 2
Enter the length of the rectangle: 4
Enter the width of the rectangle: 6
The area of the rectangle is: 24.0

Choose the shape to calculate the area:
1. Circle
```

```
2. Rectangle
3. Triangle
Enter your choice: 3
Enter the base of the triangle: 5
Enter the height of the triangle: 8
The area of the triangle is: 20.0
```

Question 3: Book Management System

Problem Statement:

Write a Java program that implements a simple book management system. The system should allow users to perform the following operations:

1. Add a new book.
2. View the details of a specific book.
3. Display all books.
4. Exit the system.

Sample output:

```

Choose an action:
1. Add a book
2. View book details
3. View all books
4. Exit
Choice of action:
1
Enter book title: One Piece
Enter book author: Eichiro Oda

Choose an action:
1. Add a book
2. View book details
3. View all books
4. Exit
Choice of action:
1
Enter book title: Naruto
Enter book author: Kishimoto

Choose an action:
1. Add a book
2. View book details
3. View all books
4. Exit
Choice of action:
1
Enter book title: AOT
Enter book author: Isayama

Choose an action:
```

```
1. Add a book
2. View book details
3. View all books
4. Exit
Choice of action:
2
Enter book title:
naruto
Book Details: Naruto by Kishimoto

Choose an action:
1. Add a book
2. View book details
3. View all books
4. Exit
Choice of action:
3
One Piece by Eichiro Oda
Naruto by Kishimoto
AOT by Isayama

Choose an action:
1. Add a book
2. View book details
3. View all books
4. Exit
Choice of action:
4
Program ending...
```

Question 4: ISBN Checker

Problem statement:

Write a Java method to validate a list of ISBN-10 numbers. An ISBN-10 number consists of 9 digits followed by a check digit.

Steps to Validate an ISBN-10 Number:

1. The number must have exactly 10 characters.
2. The first 9 characters must be digits.
3. The last character can either be a digit (0-9) or the letter 'X' (representing 10).
4. Compute the check digit using the formula:

   Check digit=(1×digit1 + 2×digit2··· + 9×digit9) mod 11
5. Compare the computed check digit with the 10th character in the string (either '0-9' or 'X').

Hint:

isValidISBN Method:

- This method takes a single ISBN-10 number as input and performs validation.
- It calculates the weighted sum of the first 9 digits and compares the last character (check digit) using the given formula.

- If the last character is 'X', it's treated as the digit 10.

validateISBNList Method:

- This method takes an array of ISBN-10 strings, validates each one using the isValidISBN method, and stores the results in a boolean array.

Main Method:

- Tests the validation logic with sample input and prints the validation results for each ISBN number.
- Sample Input:
  String[] isbnList = {"123456789X", "1234567890", "0471958697"};
- Sample Output:
  The validation results are : true false true

Question 5: Student Assessment Management System

Problem Statement:

You are tasked with developing a program to manage student marks for a viva assessment. The program should handle the following tasks:

a. Store the information of students, including their student ID, name, and marks.
b. Display the list of students with their respective IDs and marks.
c. Find the student with the highest marks.
d. Calculate the average mark of all students.
e. List students whose marks are below the average.

You are provided with the **main method**, which handles basic program flow. Your task is to implement the remaining methods to complete the program.

```
public static void main(String[] args) {

        String[] studentID = {"S0001", "S0002", "S0003", "S0004", "S0005", "S0006"};
        String[] studentName = {"John", "Cindy", "Alex", "Ali", "Rosli", "Roger"};
        int[] mark = {59, 62, 21, 36, 85, 74};

        String[][] studentInfo = getStudentInfo(studentID, studentName, mark);

        System.out.println("List of Students and their Marks: ");
        printStudentInfo(studentInfo);

        System.out.println("Student with highest marks: ");
        findStudentWithHighestMarks(studentInfo);

        double average = findAverage(mark);
        System.out.println("Average mark: " + average);

        System.out.println("Students scoring below the average:");
        listStudentsBelowAverage(studentInfo, average);

}
```

Tip: You can use **`String.valueOf()`**, **`Integer.parseInt()`**, **`Double.parseDouble()`**, and other suitable methods for data conversions.

You are required to implement the following methods to complete the program:

1. **`getStudentInfo(String[] studentID, String[] studentName, int[] mark)`**
   This method should return a **two-dimensional array** where each student's ID, name, and marks are stored together.

2. **`printStudentInfo(String[][] studentInfo)`**
   This method should print the student details in the format:
   <Student ID> - <Student Name> : <Mark>

3. **`findStudentWithHighestMarks(String[][] studentInfo)`**
   This method should find and print the name of the student with the highest marks. In the case of ties, the first student with the highest mark should be returned.

4. **`findAverage(int[] mark)`**
   This method should return the average mark of the students. The result should be a floating-point number.

5. **`listStudentsBelowAverage(String[][] studentInfo, double average)`**
   This method should print the names of students whose marks are below the a average, showing both their names and marks.

Sample Output:

```
List of Students and their Marks:
S0001 - John     : 59
S0002 - Cindy    : 62
S0003 - Alex     : 21
S0004 - Ali      : 36
S0005 - Rosli    : 85
S0006 - Roger    : 74

Student with highest marks:
Rosli: 85

Average mark: 56.166666666666664

Students scoring below the average:
Alex : 21
Ali : 36
```

Question 6: G101

Problem Statement:

On a clear Saturday morning, after having their brunch in the KK8 cafeteria, Ridwan and Suresh came back to room G101 and found their roommate, Kah Sing, sitting at his desk, looking frustrated.

Ridwan: ‘Hey, Sing, what happened, why do you look so irritated?’

Suresh: ‘Ya, bro, if you have anything just tell us.’

Kah Sing: ‘Wan, Su, you know actually I planned to go out for a walk round Bukit Cinta when it’s sunrise time for exercise. Then, yesterday I got an online meeting for assignment until super late at night.’

Ridwan: ‘No wonder, I saw you just sitting at your desk since like early evening, looking stressed, I’ve gone to bed already you’re still there.’

Kah Sing: ‘Right, I got too many things to do already. Yesterday I just kept doing my work until like 4 am, then I thought I can sleep 3 hours then wake up at 7 am to go out and exercise, then who knows I overslept already.’

Suresh: ‘Of course you overslept. You stayed up so late the last few days, if I were you I also can’t take it.’

Kah Sing: ‘I know, if I don’t rush them though, I’ll have more to do moving forward, and if I don’t go exercise for a long time I’m scared I’ll get some health issues.’

Ridwan: ‘Maybe we can help you track your sleep, so you can have a healthier lifestyle and we can hang out more like last time. I just got an alarm clock.’

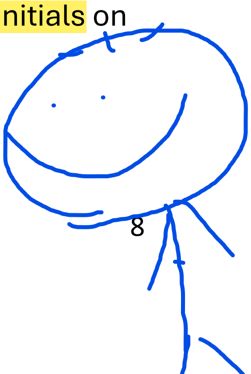
Kah Sing: ‘An alarm clock, don’t we have three already?’

Suresh: ‘No, Wan told me this before, this alarm clock is digital and programmable, you just set the start time of the clock and the end time for the alarm to ring, then the clock will calculate, like accurate to seconds, how much time is there in between for you. We plan to have it do 12-hour and 24-hour formats, but we haven’t written the code yet.’

Kah Sing: ‘That’s a very brilliant idea, I was thinking maybe we can treat this as practice for our upcoming competitive programming competition or hackathon actually, that’d be nice!’

Ridwan: ‘Ya, don’t forget to add 10 more exclamation marks right after Sing's initials on the same line, so he can really see them! This one is only for Sing!’

remove all weird symbol
check g101 -> welcome msg
lee kah sing -> WE KNOW IT’S YOU
kah sing lee -> WE KNOW IT’S YOU -- LEE KAH SING!
time 12am- 6am -> SLEEP NOW!!!!!!!!!!



Suresh: ‘I think we should add another line between the initials line and the time interval line, for any user, if he types in the start time of after 12:00:00 am and the end time of before 06:00:00 am exclusive, we should print out “SLEEP NOW!!!!!!!!!!” right away!’

Kah Sing: ‘Sure thing. Since some of our friends could use it, I have friends who have special characters in their names, like Brian O’Brien, Chiang Hsiao-hsin, oh, and Suresh, your full name is Suresh a/l Subramaniam, right? Since Malaysian Indians don’t count the “a/l”, “a/p”, “al”, “ap” in their initials, the full name initials should be “BOB”, “CHH”, “SS”, in the order.'

Suresh: 'And, just in case some people use underscores, full stops to separate words in their names we should separate into parts with them as well, like Sing's Instagram username “lee_kah_sing”, and Wan's "ridwan.faiz", the outputs should be the initials of “LKS” and "RF" respectively! The name section only has characters which are uppercase, lowercase letters, and special characters namely the whitespace ( ), apostrophe('), dash (-) and underscore (_) characters!’

Ridwan: ‘Yes, then since for us Malaysian Malays, our names have “bin” or “binti”, and my full name is Ridwan Faiz bin Mohamad Hassan, then my initials should be “RFMH”, am I right?’

Suresh: ‘That’s right, I also think, for us three only we should print a welcome message before the initials, like only if Sing enters "Kah Sing", "Lee Kah Sing", "Kah Sing Lee; if I enter "Suresh" on top of my full name; if Wan enters "Ridwan", "Ridwan Faiz", "Ridwan Faiz Mohamad Hassan" on top of his full name?’

Ridwan: ‘I think we just print “Welcome to G101, Kolej Kediaman Kinabalu, Universiti Malaya!” like that.’

Kah Sing: ‘Wow, you guys have so many ideas!’

Suresh: ‘Eh, one more thing, haha. We originally wanted to make this clock for Sing. Wan, don’t forget he can get a bit blur sometimes! Like last time, on the Residential College Facilitator Registration form, he writes his name first, then his surname on the application form. The staff at the counter kept calling him “Sing Lee”, so funny!’

Kah Sing: ‘Wah, I remember, a bit embarrassing, haha... Luckily you guys went with me and spotted the mistake before I submitted the form. Hope we can still keep our room here next semester.’

Ridwan: ‘Ya, I think no matter it’s “Kah Sing Lee” or “Lee Kah Sing”, and any other variant of the order (with delimiters other than a whitespace character, namely the apostrophe (‘), the dash (-), and the underscore (_) characters), as long as it has “Kah Sing” and “Lee” only and in any order in the name to be input to the clock, we should just treat it as our brother, Sing, like that, so we just output “LKS” for the matter.’

Suresh: ‘Also, if he enters “Kah Sing Lee” or any other variant of the order (with delimiters other than a whitespace character, namely the apostrophe (’), the dash (-), and the underscore (_) characters), we just print all capitals “WE KNOW IT’S YOU -- LEE KAH SING!”; if he enters “Lee Kah Sing” or any other variants of the order (with the same requirements for the delimiters as when he inputs “Kah Sing Lee”), we just print all capitals “WE KNOW IT’S YOU!”, like that.’

Kah Sing: ‘I see, so if all of the requirements are met, in this order, we first print a line of, say, 60 ‘+’ characters; a line of the welcoming message if needed; a line of the initials of the user; a line of the special all capitals message for me only if needed, a line of the message to sleep now; a line of the time interval; a line of the 60 ‘+’ characters in the end?’

Ridwan: ‘Correct, I think that should be it, the clock should work just fine after we finish coding!’

Kah Sing: ‘Wow, thanks so much... I’m super touched to have brothers like you two!’

Ridwan, Suresh: ‘Ya, bros help bros, Sing!’

Requirements

1. Under the class G101,
    a. Create a method named ***generateInitials*** to generate the initials of the user(s) of the computer program.

    b. Create a method named ***isPrintingWelcomeMessage*** to determine whether to print the welcome message as follows:
    ‘Welcome to G101, Kolej Kediaman Kinabalu, Universiti Malaya!’.

    c. Create a method named ***calculateInterval*** to calculate the interval between the start and end times.

2. Ensure there is no error in compiling, running and handling invalid inputs in the code, and when sample inputs and outputs are passed in, there is no error found by the grader.

3. Ensure your report is done and everything you include in the report aligns with all the aspects of your code.

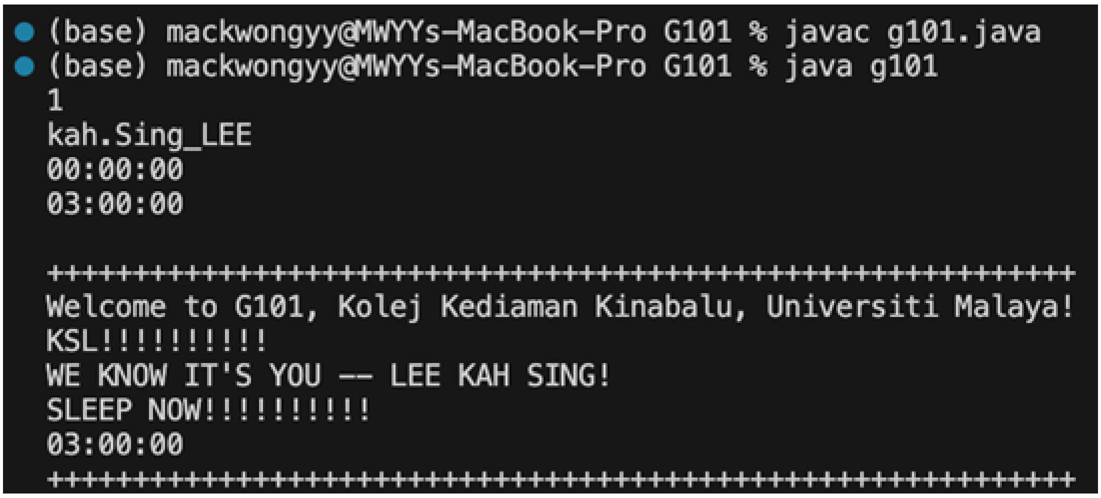
For all notes on input and output, refer to the following pages.

Input
The first line contains the number of test cases which ranges from 1 to 10 inclusive.

In each of the test cases, the first line contains the name of the user of the alarm clock. The second and third lines contain the current time and the target time each in the format of (hours, minutes, seconds) respectively, as both lines can either be in 12-hour or 24-hour format. The hours will be between 0 and 23 inclusive and the minutes and seconds between 0 and 59 inclusive.

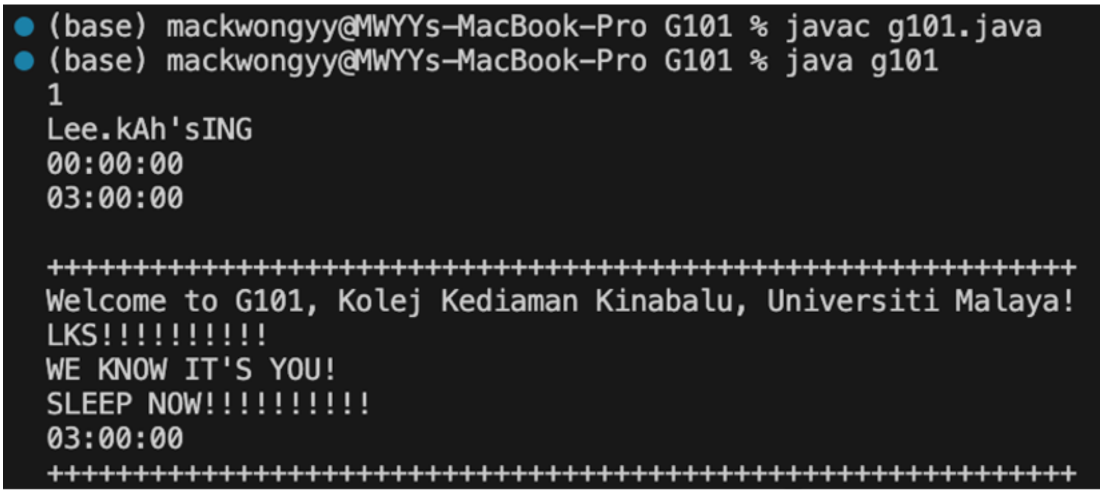
Output
For each of the test cases, output at least four, at most seven lines, depending on the requirements mentioned in the problem statement. The first line shall be sixty '+' characters, whereas the second line shall be the initials of the name of the user of the alarm clock. For the next few lines, if any, if certain conditions in the problem statement are met, output accordingly. The last second line shall be the time interval in the hh:mm:ss format, whereas the last line shall be sixty '+' characters. Note the range of output interval is between '00:00:00' and '23:59:59'.

Sample Input/Output



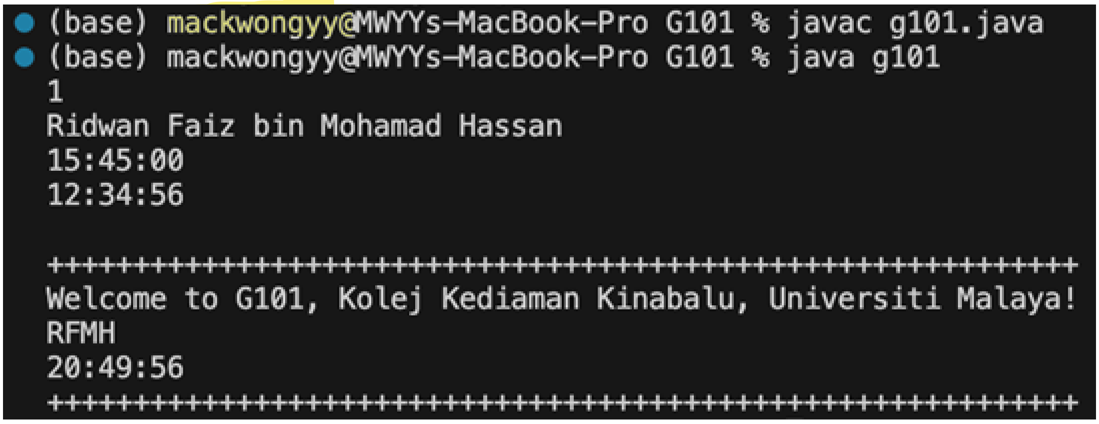
```
● (base) mackwongyy@MWYYs-MacBook-Pro G101 % javac g101.java
● (base) mackwongyy@MWYYs-MacBook-Pro G101 % java g101
  1
  kah.Sing_LEE
  00:00:00
  03:00:00

  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  Welcome to G101, Kolej Kediaman Kinabalu, Universiti Malaya!
  KSL!!!!!!!!!!
  WE KNOW IT'S YOU -- LEE KAH SING!
  SLEEP NOW!!!!!!!!!!
  03:00:00
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```



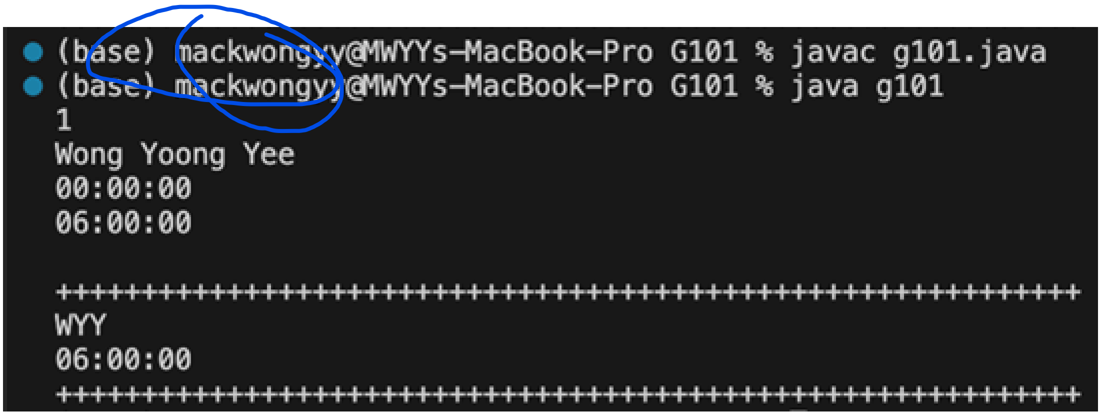
```
● (base) mackwongyy@MWYYs-MacBook-Pro G101 % javac g101.java
● (base) mackwongyy@MWYYs-MacBook-Pro G101 % java g101
  1
  Lee.kAh'sING
  00:00:00
  03:00:00

  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  Welcome to G101, Kolej Kediaman Kinabalu, Universiti Malaya!
  LKS!!!!!!!!!!
  WE KNOW IT'S YOU!
  SLEEP NOW!!!!!!!!!!
  03:00:00
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```



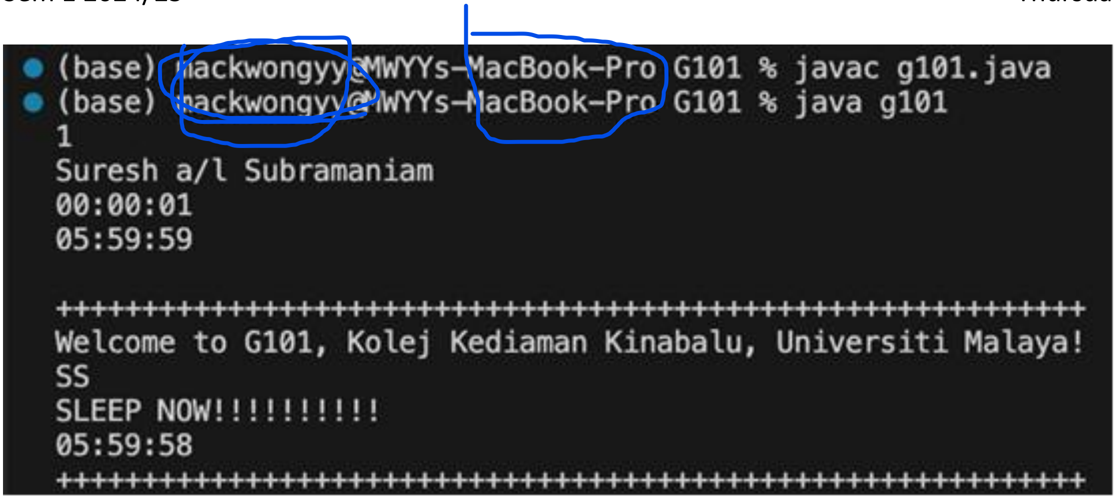
```
● (base) mackwongyy@MWYYs-MacBook-Pro G101 % javac g101.java
● (base) mackwongyy@MWYYs-MacBook-Pro G101 % java g101
  1
  Ridwan Faiz bin Mohamad Hassan
  15:45:00
  12:34:56

  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  Welcome to G101, Kolej Kediaman Kinabalu, Universiti Malaya!
  RFMH
  20:49:56
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```



```
● (base) mackwongyy@MWYYs-MacBook-Pro G101 % javac g101.java
● (base) mackwongyy@MWYYs-MacBook-Pro G101 % java g101
  1
  Wong Yoong Yee
  00:00:00
  06:00:00

  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  WYY
  06:00:00
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```



```
● (base) mackwongyy@MWYYs-MacBook-Pro G101 % javac g101.java
● (base) mackwongyy@MWYYs-MacBook-Pro G101 % java g101
  1
  Suresh a/l Subramaniam
  00:00:01
  05:59:59

  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  Welcome to G101, Kolej Kediaman Kinabalu, Universiti Malaya!
  SS
  SLEEP NOW!!!!!!!!!!
  05:59:58
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```