PECANWOOD

COLLEGE

Prepared for Life

INFORMATION TECHNOLOGY GRADE 11
JULY PRACTICAL EXAMINATION

NAME: _____ **GRADE:** _____

DATE: 22 JULY 2023 **MARKS: 65**
MODERATOR: MR C SEEWALD **TIME: 3 HOURS**
EXAMINER: MR SC EILERTSEN

INSTRUCTIONS:

1. This examination is made up of 5 pages including a page for making notes. Please ensure that your paper is complete.
2. You will be provided with two text files i.e. "questionOne.txt" and "questionTwo.txt"
3. You may not sit at your usual IT terminal but will have to log into your exam account using your ESD number.
4. You must use the same names as are used in the examination paper for projects, packages, and classes.
5. Credit is given for good layout, indentation, and well-chosen variable names.

Section One **Reading from a text file – No delimiters.**

A sensor at the side of a marathon reads in each runner's race number as they run past and saves them into a simple text file - one number per line. For the purposes of this exam there will be a small number of runners. You need to code a program that checks these numbers for problems.

Here is the file with the runner's numbers. You have been provided with this text file – it is called "questionOne.txt". The first two letters indicate the country of origin e.g. "ZA" indicates South Africa. The number that follows is a unique number for identification purposes and can be any number of digits.

ZA-78
ZA-45
ZA-451
UK-37
CH-8
US-437
ZA-36
USA-43
ZA-67
FR-34
UK-37
ZA-67
US-133
PK-6
ZA-34

The types of problems we are looking for are race numbers in the wrong format, and runners from countries that are unknown to the database.

The program must also determine the number of men and the number of women – men are provided with even numbers and women with odd numbers.

1.1) Create a new project called **JulyExams2023**. Create a new package called **questionOne**. Create a UI class called **QuestionOneUI**. Create a manager class called **QuestionOneManager**. (1)

1.2) Credit is given for good layout, indentation, and well-chosen variable names. (2)

1.3) In your UI class create a manager object. (1)

1.4) In your manager class create a method to open and read the race numbers into an array – ignore any potential problems in the text file – read in the race numbers regardless. If the file is not found a useful error message must be provided. (15)

1.5) Create a method to print out the runner's array. Your output should look like this . (3)

Runner Array ZA-78
Runner Array ZA-451
Runner Array ZA-45
Runner Array UK-37
Runner Array CH-8 etc etc etc

1.6) In your manager class create a method to determine the number of men and the number of women. Your output should look like this . . . (9)

```
Number of men is: 6
Number of women is: 9
Total number of runners is: 15
BUILD SUCCESSFUL (total time: 0 seconds)
```

1.7) Create a method to check for errors in the format of the runner's numbers. The format is as follows.

The country can only be 2 letters in length.
The number can be any valid number. (5)

Your output should look like this . . .

```
Error with runner code: 8
BUILD SUCCESSFUL (total time: 0 seconds)
```

1.8) Create a method to check for unknown countries. Known countries are as follows . . .

ZA, UK, CH, US and FR
Any other country code is unknown and should be reported. (5)

Your output should look like this . . .

```
Error with country code: PK
BUILD SUCCESSFUL (total time: 0 seconds)
```

Total: 41

Section Two **Reading from a text file - With delimiters**

Use the same project. Create a new package called **questionTwo**. Create a UI class called **QuestionTwoUI**. Create a manager class called **QuestionTwoManager**.

Here we are looking for runners that do not meet the age requirements i.e. runners must be 16 years or older and 65 years or younger. The program must report any runner that does not meet this requirement.

We will be using the file called "questionTwo.txt" for this question. This is what the text file looks like where the first token is the runners race number, the second token is their age, and the third token is their name.

ZA-457#23#Andrew Alpha
UK-7#76#Betty Bravo
CH-654#56#Chunk Charlie
US-99#8#David Delta
ZA-765#21#Erin Echo
UK-54#25#Francois French
ZA-23#29#Gretha Golf

2.1) In your manager class, create a method to read in the text file. If a runner does **NOT** meet the requirements their runner's number, their age and their name must be stored in a String array. (There is no need to read all the runners into a runner's array – we only need an array for runners that do **not** meet the age requirements.) (21)

2.2) Create a method to print out the array of runners that do not meet the requirements. (3)

Your output should look like this . . .

```
run:
Does not qualify: UK-7$76$Betty Bravo
Does not qualify: US-99$8$David Delta
BUILD SUCCESSFUL (total time: 0 seconds)
```

Total: 24

GRAND TOTAL: 65 Marks

Additional Paper 1: