Name:		Index Number:		Class:	

DUNMAN HIGH SCHOOL
Preliminary Examination
Year 6

COMPUTING

9597

(Higher 2)

2 September 2019
3 hours 15 minutes

Paper 1

Additional Materials: Data files

READ THESE INSTRUCTIONS FIRST

Type in the EVIDENCE.docx document the following:

- Candidate details
- Programming language used

Answer **all** questions.

All tasks must be done in the computer laboratory. You are not allowed to bring in or take out any pieces of work or materials on paper or electronic media or in any other form.

All tasks and required evidence are numbered. The marks is given in brackets [ ] at the end of each task.

Copy and paste required evidence of program code and screenshots into the appropriate cells in EVIDENCE.docx.

Data files

Q1 – IOI19.TXT
Q2 – SGQR.TXT
Q3 – PSLE21.TXT
Q4 – IMAGE1.IN

1. The file `IOI19.TXT` stores the scoreboard of the 31st International Olympiad in Informatics (IOI) held in Baku, Azerbaijan from 4 to 11 August 2019. The first line contains the header with contestant's rank, name, team (country), and overall score. The remaining lines contain the corresponding data. IOI 2019 eventually awarded 28 Gold, 54 Silver and 81 Bronze medals. A country typically sends up to 4 contestants to participate in IOI.

Task 1.1

Determine the top 3 teams in the competition and display their country names as well as the number of Gold, Silver and Bronze medals attained. Teams which are tied will be ordered by their country names in alphabetical order and will share the same rank.

Sample output:

```
Top 3 teams:
1 ABC 4G
2 DEF 3G1S
2 GHI 3G1S
4 JKL 2G1S1B
4 MNO 2G1S1B
```

Evidence 1

Program code. [9]

Evidence 2

Screenshot. [1]

Task 1.2

Write program code to prompt a user to enter a team name and display the results of its participants. Participants who did not receive a medal will be denoted with `P`. The program will terminate when a user enters the text string '`ZZZ`' (without the quotes).

Sample interaction:

```
Enter team: SGP
Eu-Shaun Leong          G
Lee Jeffrey Chun Hean   S
Benson Zhan Li Lin      B
Daniel Zhenghao Choo    P

Enter team: ZZZ
Bye
```

Evidence

Program code. [4]

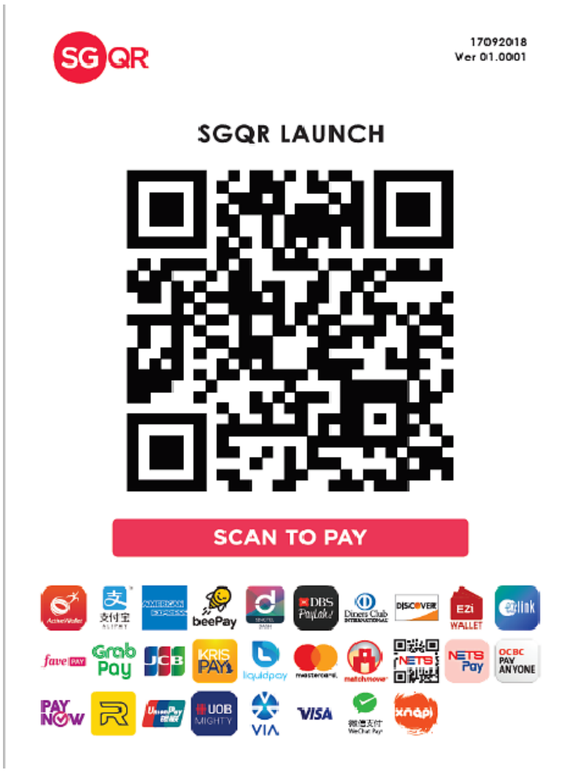
Evidence 4

Screenshot. [1]

2. Dubbed first of its kind globally, the Singapore Quick Response Code (SGQR) is an infrastructure-light technology that will help to simplify QR e-payments in Singapore for both consumers and merchants.

The SGQR is based on the QR Code Specification for Payment System - Merchant-Presented Mode standard issued by EMVCo, which has the benefits of international interoperability, multi-tenancy of QR schemes and non-sensitive data presented for payments.

According to the specification, the parsed SGQR text string contains data items, with each data item adhering to the following structure: id, length, value. Two such data items are highlighted in bold in the following diagram:



00020101021126810011SG.COM.NETS01231198500065G9912312359000211111686614000308686614019908604108C251800007SG.SGQR01121809072DD85C020701.00010306079027040201050206060400000708201809155204581253037025802SG5912SOBAEXPRESS6009Singapore**630457B3**

Thus for the first data item **000201**, 00 is the id, 02 is the length, and 01 is the value.

And for the last data item **630457B3**, 63 is the id, 04 is the length, and 57B3 is the value.

The value 57B3 is also a hexdecimal number to verify the integrity of the SGQR data.

Task 2.1
Write program code to extract the last data item of the SGQR stored in `SGQR.TXT`. For the example above, it will be the data item with id 63 and length 4 i.e. 630457B3.

Evidence 5

Program code. [3]

Evidence 6
Screenshot. [1]

Task 2.2
Write a `hex2oct` function which takes in a hexadecimal number string and returns its equivalent octal number string. For example `hex2oct('A')` returns '`12`'. You may not use Python's built in `int(num, 8)`, `int(num, 16)`, `bin()`, `oct()` or `hex()` functions. Use the hexadecimal number string '`4F63A`' to to test your program code.

Hint: One hexadecimal digit can be expressed as four binary digits and one octal digit can be expressed as three binary digits.

Evidence 7
Program code. [5]

Evidence 8
Screenshot. [1]

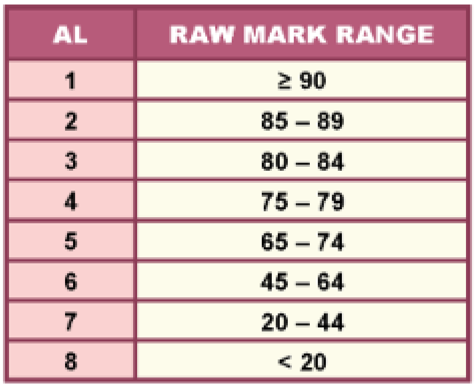
Task 2.3
Write program code to perform input validation for a hexadecimal number string. Test your program with suitable test data.

Evidence 9
Program code. [3]

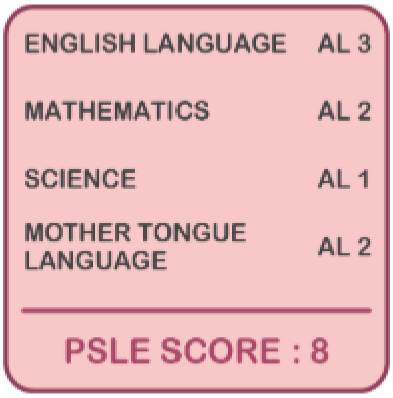
Evidence 10
Screenshots. [2]

3. From 2021 onwards, the Primary School Leaving Exam (PSLE) will be scored with wider bands, replacing the current T-scores.

Each subject will be scored using 8 bands known as Achievement Levels (AL), with AL 1 being the best score and AL 8 being the lowest score. The student's PSLE Score will be the sum of the four subject scores. The PSLE Score will range from 4 (best) to 32.



AL	RAW MARK RANGE
1	≥ 90
2	85 – 89
3	80 – 84
4	75 – 79
5	65 – 74
6	45 – 64
7	20 – 44
8	< 20



ENGLISH LANGUAGE	AL 3
MATHEMATICS	AL 2
SCIENCE	AL 1
MOTHER TONGUE LANGUAGE	AL 2

PSLE SCORE : 8

Secondary 1 posting will continue to be based on academic merit, using the PSLE Score.

Each student will submit a list of 6 schools in order of preference. If two students with the same score are being considered for the last place in a school, the following tie-breakers will be used:

- Citizenship - priority given to Singapore Citizens (SC), then Singapore Permanent Residents (PR), then International Students (IS)
- Choice order of schools - priority given to the student who indicates a certain school as a higher choice
- Computerised balloting

The file `PSLE21.txt` contains the application information of 400 Primary 6 students of a primary school with the following structure:

```
<StudentID>,<EnglishLanguageMark>,<MathematicsMark>,<ScienceMark>,<MotherTongueLangueMark>,<Citizenship>,<SchoolChoice1>,<SchoolChoice2>,<SchoolChoice3>
```

You may assume that in this school all students study subjects at the standard level. Also, all of them have made up their mind to only apply to 3 schools of their choice. If a student is unable to get admission to a school of their choice, they will be posted to SchoolD.

It is decided to process and store the following application information about the student in 4 linked lists. Each linked list pertain to the vacancy positions of the 4 schools. Schools A, B and C have 120, 150 and 80 available places. The data to be stored in each linked list node include: PSLE score, student ID, citizenship and the 3 school choices.

Task 3.1

Write program code to read in and store the contents of the file `PSLE21.txt` in a dictionary `students` with key `StudentID` and value the computed PSLE score, citizenship and three school choices. Display the first 10 dictionary entries in `students`.

Evidence 11
Program code. [6]

Evidence 12
Screenshot for first 10 dictionary entries in `students`. [1]

Task 3.2
Using OOP where appropriate, write program code to declare and initialise the necessary classes. Insert the 400 students from the `students` dictionary in Task 3.1 to the appropriate linked lists in your main program driver code.

Evidence 13
Program code. [16]

Evidence 14
Screenshots for the first 5 entries in each linked list. [4]

Task 3.3
Students `P351` and `P365` who were previously Singapore Permanent Residents (PR) have successfully become Singapore Citizens (SC). Write the necessary program code to update their citizenship status and new secondary 1 posting order.

Evidence 15
Program code. [5]

Evidence 16
Screenshots. [2]

Task 3.4
Student `P286` has decided to emigrate to another country with his/her parents. Write the necessary program code to remove him/her from his/her existing allocation and perform the necessary adjustments to fill up the vacancy.

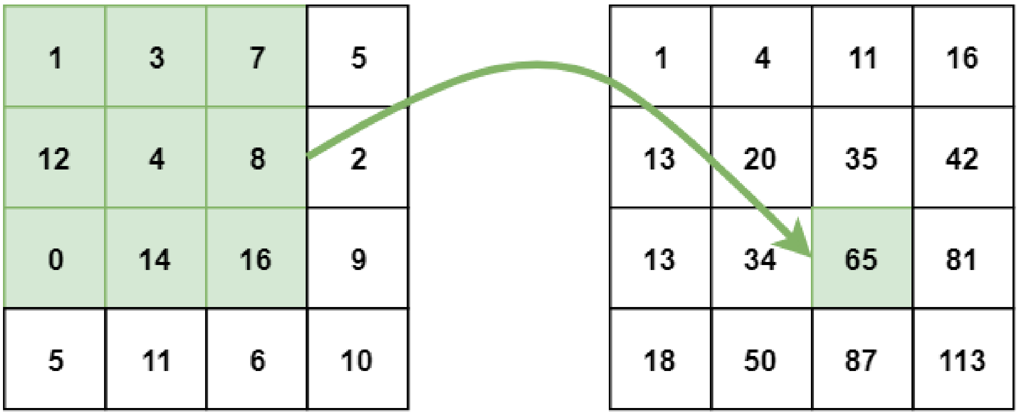
Evidence 17
Program code. [5]

Evidence 18
Screenshot. [1]

4. The Viola-Jones object detection algorithm, named after two computer vision researchers Paul Viola and Michael Jones, uses integral images to detect the presence of facial features in an image efficiently.

An integral image (also known as a summed-area table) is the name of both a data structure and an algorithm used to obtain this data structure. It uses a quick and efficient way to calculate the sum of pixel values in a rectangular part of an image.

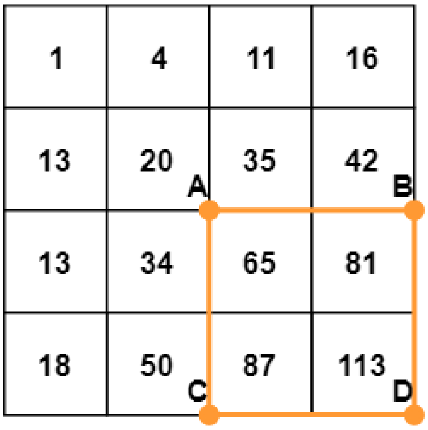
In an integral image, the value of each point is the sum of all pixels above and to the left, including the target pixel:



1	3	7	5
12	4	8	2
0	14	16	9
5	11	6	10

1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

The integral image can be calculated in a single pass over the original image. This reduces summing the pixel intensities within a rectangle into only three operations with four numbers, regardless of rectangle size:



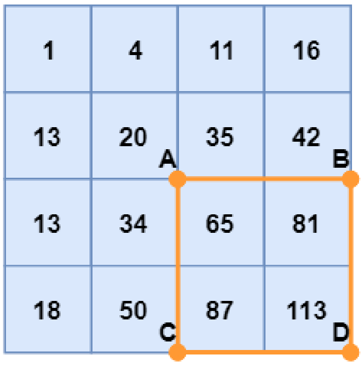
1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

ABCD = ?

The sum of pixels in the rectangle ABCD can be derived from the values of points A, B, C, and D, using the formula D - B - C + A. It is easier to understand this formula visually:

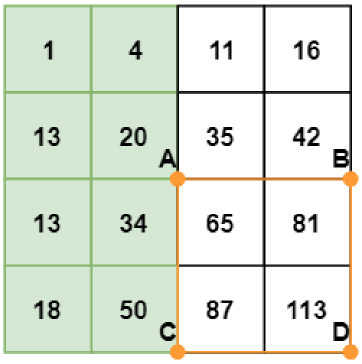
Note that subtracting both B and C means that the area defined with A has been subtracted twice, so we need to add it back again.

Thus D - B - C + A = 113 - 50 - 42 + 20 = 41.



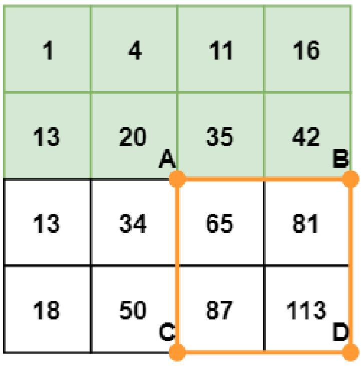
1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

D = 113



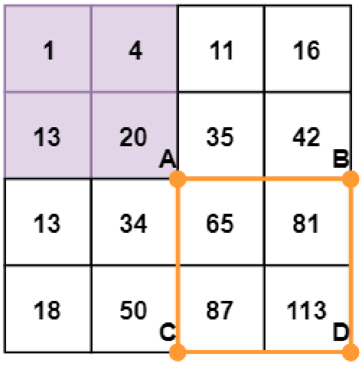
1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

C = 50



1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

B = 42



1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

A = 20

Task 4.1
Write an `integral_image()` function which reads in the data from the file `IMAGE1.IN` into a 2D array and computes and outputs the integral image to a file `IMAGE1.OUT` using the algorithm described above, and also displays the result of D - B - C + A to the screen.

Evidence 19
Program code. [13]

Evidence 20
Screenshots of `IMAGE1.OUT` and output of D - B - C + A. [2]

Task 4.2
Write a `magic()` function which is a generalisation of your `integral_image()` function which will work for any `m * n` rectangular 2D array and any rectangle ABCD. Programmatically randomise your image with suitable values (`m`, `n` >= 8) in `IMAGE2.IN` and work your magic on this pseudo-randomly generated file to produce `IMAGE2.OUT` and the updated computed value of D - B - C + A.

Evidence 21
Program code. [13]

Evidence 22
Screenshots of `IMAGE2.OUT` and output of D - B - C + A. [2]

***** THE END *****