

1 [ACJC/PRELIM/9569/2021/P1/Q1]

A famous restaurant only accommodates one seating daily from 6.00 pm to 8.00 pm. It has 10 tables, each with a maximum capacity between 2 and 8 people. Advanced reservation is required to dine in at the restaurant.

The owner of the restaurant decides to write a program to handle reservations. As a trial, it can only take a booking for one evening only.

A procedure to initialise the arrays `MaxSize`, `IsBooked` and `GroupSize` has been defined. The indexes of each array corresponds to the table number.

Index	MaxSize		IsBooked		GroupSize
1	2	1	FALSE	1	
2	2	2	FALSE	2	
3	4	3	FALSE	3	
4	4	4	FALSE	4	
5	4	5	FALSE	5	
6	6	6	FALSE	6	
7	6	7	FALSE	7	
8	6	8	FALSE	8	
9	8	9	FALSE	9	
10	8	10	FALSE	10	

The procedure `BookTable` is shown below. When a booking enquiry is made, the number of customers is keyed in.

```

01 PROCEDURE BookTable
02     DECLARE NumberOfCustomers, TableNumber :  INTEGERS
03     DECLARE Found :  BOOLEAN
04     INPUT NumberOfCustomers
05     TableNumber ← 0
06     FOUND ← False
07     REPEAT
08         TableNumber ← TableNumber + 1
09         IF MaxSize[TableNumber] > NumberOfCustomers AND
IsBooked[TableNumber] = FALSE
10             THEN
11                 Found ← TRUE
12             ENDIF
13     UNTIL Found = TRUE AND TableNumber = 10
14         IF Found = FALSE
15             THEN
16                 OUTPUT "No tables with enough seats available."
17             ELSE
18                 GroupSize[TableNumber] ← NumberOfCustomers
19                 OUTPUT "Booking is successful! Table no:", TableNumber
20         ENDIF
21 ENDPROCEDURE

```

(a) There are two errors and one missing line of code in the procedure above.

(i) Name the type of the errors. [1]

(ii) Describe the errors and the changes required to correct them. [3]

(iii) Write the missing line of code and state where it should be located. [2]

(b) Once the procedure BookTable is able to run correctly, the owner decides to improve its functionality.

The procedure should ask the user to input the name and the mobile number of the person making the reservation when a table with enough seats can be found.

Name and describe two data validation techniques that can be applied to any of the inputs mentioned above. [2]

(c) Explain the difference in the type of memory allocation for an array and a linked list.

[2]

2 [ACJC/PRELIM/9569/2021/P1/Q2]

A hash table has 8 spaces to store strings, indexed from 1 to 8 inclusive.

The hash function finds the ASCII number of the first letter of the string, then counts the number of 1s in its binary representation. This is the index in which the string will be inserted into the hash table.

For example, the string 'Arlington' will have index 2 because the ASCII number of 'A' is 65, which is 1000001 in binary, and there are two 1s.

The following strings are to be inserted into the hash table in the order given.

'Grover',

Horsburgh',

'Island',

'Jordan',

'Kalman'

(a) Find the output of the hash function for each of the strings. [5]

(b) (i) Suppose collisions in the hash table are to be resolved using open hashing.

Draw the hash table after all five strings are inserted. [5]

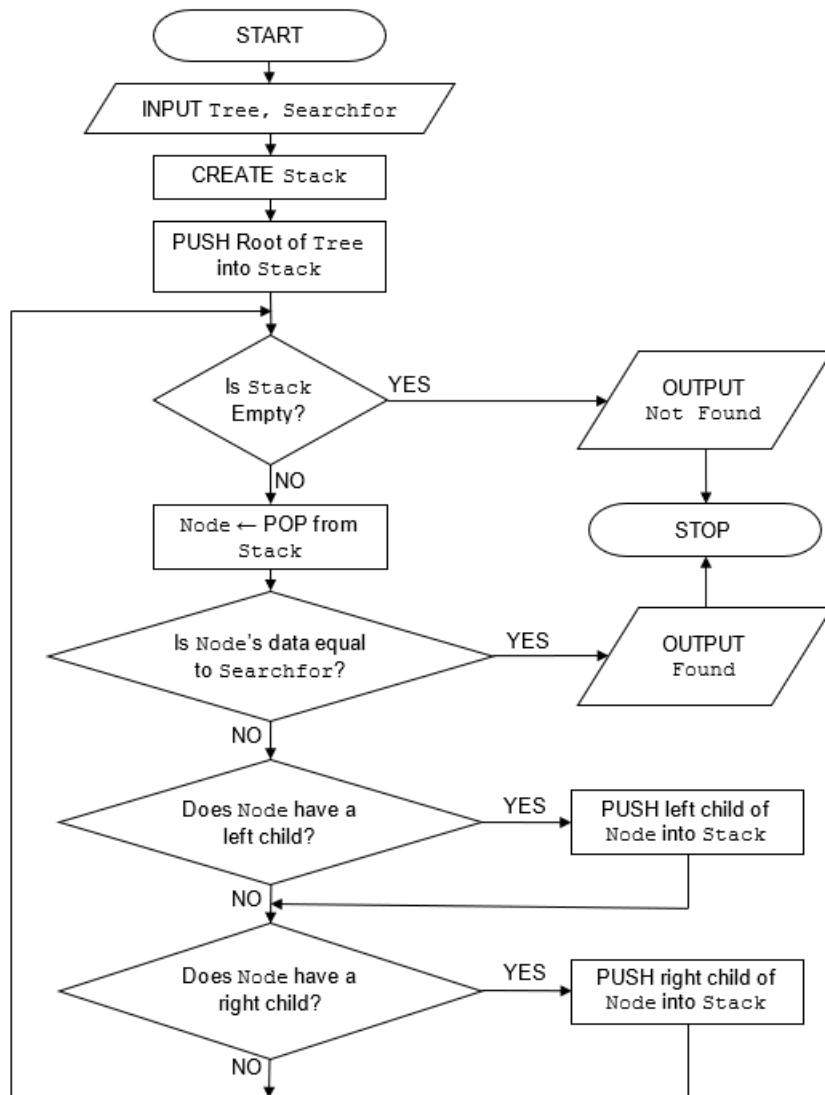
(ii) Suppose instead that collisions in the hash table are to be resolved using closed hashing, where spaces 6 to 8 (inclusive) are used as the overflow storage.

Draw the hash table after all five strings are inserted. [2]

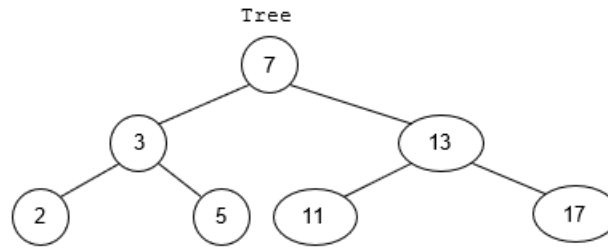
(c) Explain why the space with index 1 in the hash table will never be occupied unless there is a collision. [2]

3 [ACJC/PRELIM/9569/2021/P1/Q3]

The diagram below shows a flowchart for performing a search through a binary tree. The algorithm searches through Tree for Searchfor. If it finds a node whose data is equal to Searchfor, it outputs Found. Otherwise, it outputs Not Found.



(a) Given the input Tree below and a Searchfor value of 5, draw a trace table to illustrate the algorithm. [5]



- (b) State whether this is a depth-first search or a breadth-first search. [1]
- (c) Draw a flowchart to illustrate how the other kind of search in (ii) can be carried out using the same input parameters. [3]
- (d) Given the same input `Tree` and the same `Searchfor` value of 5, draw a trace table to illustrate the algorithm in (iii). [5]

4 [ACJC/PRELIM/9569/2021/P1/Q4]

Merge sort and bubble sort are two sorting algorithms that can be applied to sort a list of integers in ascending order.

- (a) By briefly comparing the operation of merge sort and bubble sort, state which algorithm would be more efficient. [3]

The pseudocode for the recursive MergeSortDesc function is shown below, which takes in a list of integers and returns a new list with the integers sorted in descending order. It makes use of the MergeDesc function in line 10 that merges two lists of integers sorted in descending order into a single list of integers sorted in descending order as well.

```

01 FUNCTION MergeSortDesc(MyList: LIST) RETURNS LIST
02     MaxIndex ← LENGTH(MyList)
03     IF .....A.....
04         THEN
05             Half ← .....B.....
06             LeftList ← LEFT(MyList, Half)
07             RightList ← RIGHT(MyList, Half)
08             SortedLeftList ← MergeSortDesc(LeftList)
09             SortedRightList ← MergeSortDesc(RightList)
10             Result ← MergeDesc(SortedLeftList, SortedRightList)
11         ELSE
12             .....C.....
13         ENDIF
14         RETURN Result
15 ENDFUNCTION

```

- (b) State what is meant by a **recursive** function. [2]

- (c) Write the pseudocode for A, B and C in the algorithm. [3]

- (d) Describe the operation of the MergeDesc function.

Assume that the function does not modify the two input lists. [4]

5 [ACJC/PRELIM/9569/2021/P1/Q5]

A gym organises various classes and runs a loyalty membership programme with four tiers: Bronze, Silver, Gold and Diamond

Upon joining, each member is given a unique membership number and starts with a Bronze membership. Each member can sign up for multiple classes at a reduced rate based on the membership tier.

Each class has a unique class name. Some classes are offered at three different levels: Beginner, Intermediate and Advanced

Each instructor is identified with a unique three-character code and can take one or more classes.

A relational database is to be created to store data about members, employees and classes.

Part of the table MEMBER, which is a first attempt at the database design, is shown below.

MemberNo	MemberName	MemberTier	ClassName	InstCode
3*5	3*Lindy White	3*Silver	Body Pump	WAY
			Yoga (Beginner)	DAV
			Zumba	ROG
...
78	Derek Davis	Bronze	Muay Thai (Beginner)	CHA
...
4*132	4*John Chua	4*Diamond	Circuits (Intermediate)	JON
			Muay Thai (Intermediate)	LEX
			Yoga (Advanced)	DAV
			Zumba	ROG
...

(a) The table MEMBER is not normalised.

(i) Describe **one** potential issue that may be encountered when the data are maintained in such a non-normalised table. [1]

(ii) Explain why the table is not in first normal form (1NF). [1]

(b) A second attempt at the database design gives rise to two tables:

MEMBER(MemberNo, MemberName, MemberTier)

MEMBERCLASSES(MemberNo, ClassName, Instructor)

The primary keys are not shown.

- (i) State what is meant by a **primary key**. [1]
- (ii) By referring to the relationship between the tables MEMBER and MEMBERCLASSES, state how the relationship is implemented. [2]
- (iii) Write an SQL query to create the table MEMBER with the appropriate constraints. [4]
- (c) Another attempt at the database design needs to be made to ensure that all the tables are in third normal form (3NF).

In addition, the following data need to be recorded in the database:

- the date on which each member signs up for the gym membership;
- the attendance of each member in any classes taken;
- the original fee, i.e. before discount, of each class;
- the name and the salary of each instructor.

- (i) State the total number of 3NF tables required and give their names. [1]
- (ii) Draw the Entity-Relationship (E-R) diagram to show the 3NF tables and the relationships between them. [4]
- (iii) A table description can be written as:

TableName(Attribute1, Attribute2*, Attribute3, ...)

The primary key is indicated by underlining one or more attributes. Foreign keys are indicated by using a dashed underline/asterisk.

Using the information provided, write table descriptions for all the 3NF tables you identified in (c)(i). [8]

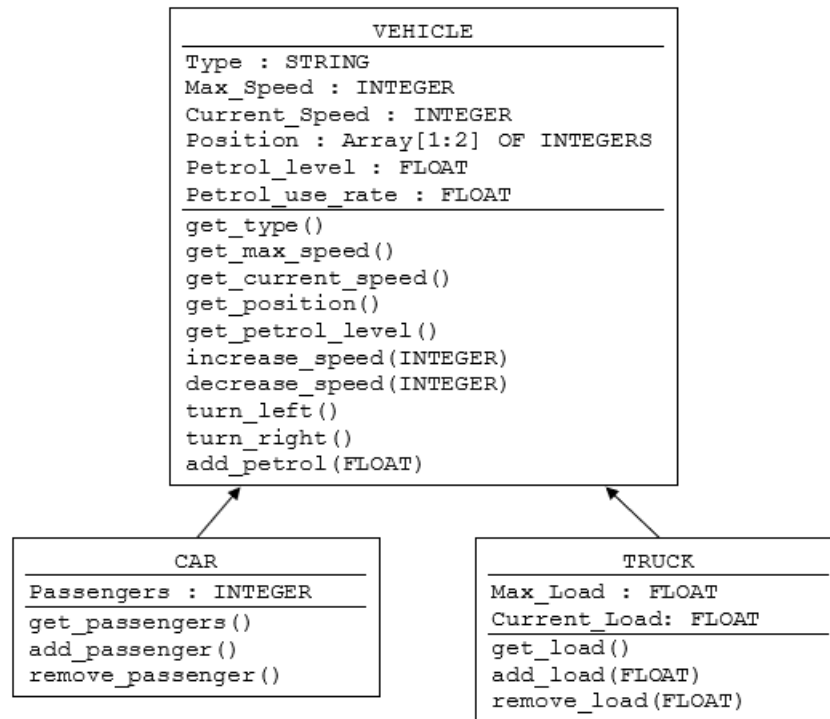
- (d) Making backups and archives are performed to prevent the loss of data.

Explain the difference between a backup and an archive. [2]

6 [ACJC/PRELIM/9569/2021/P1/Q6]

A driving simulator is programmed using Object-Oriented Programming (OOP).

The diagram below shows a UML Class Diagram with **some** of the classes, attributes, and methods used in the simulator.



- (a) State the relationship between the CAR class and the VEHICLE class. [1]
- (b) Explain briefly, in this context, how each of the following features of Object-Oriented Programming help the simulation to be developed more efficiently.
- (i) Abstraction [2]
 - (ii) Inheritance [2]
- (c) The petrol use rate depends on the speed at which the vehicle is travelling, as well as the mass of the vehicle and the contents of the vehicle – the number of passengers in a car, or the mass of the load in a truck. Explain how polymorphism can be used in this case to write the simulation. [2]

7 [ACJC/PRELIM/9569/2021/P1/Q7]

A new social media platform is to be created. In years to come, it is expected to be as popular globally as other trending social media platforms.

- (a) Give two reasons why a NoSQL database is likely to be more suitable than an SQL database for the social media platform. [2]

A basic login page that controls access to user accounts is shown below. The password field masks the user input with a dot (•) replacing each of the characters supplied.

LOGIN PAGE

Username:

Password:

- (b) When the login button is clicked, the program processes the username and password supplied by the user. It displays an error message if the username entered does not exist in the database. If the password entered matches the registered password for the username, login is granted. Otherwise, the program displays an error message to indicate the user of the incorrect password entered.
- The account will be locked if the user enters the correct username, but enters the wrong password three times.

- (i) Create a decision table to show these conditions and actions. [4]

- (ii) Simplify your decision table by removing redundancies. [1]

- (c) It is known that users tend to have different problems associated with passwords.

Besides the error message to tell the user when an incorrect password is entered, describe **two** examples based on usability principles that can be applied to improve the functionality of the login page. [2]

- (d) Explain why the HTTP POST method should be used instead of the HTTP GET method for the login request. [2]

8 [ACJC/PRELIM/9569/2021/P1/Q8]

In a hypothetical scenario, a data security company is helping a client company manage a database of the client company's customers. The data security company notices a possible vulnerability in the database.

Further investigation shows that the vulnerability is obscure and that none or few of the programmers anticipated it. Since the vulnerability is obscure, they determine that the chances of the database being breached are minimal, and decide not to tell the client company about it.

Instead, the data security company waits until the next time the database is due for scheduled maintenance to attempt to fix the vulnerability. By doing so, they can give themselves enough time to learn how to fix the vulnerability and avoid causing unnecessary panic within the client company or among the customers, which could lead to a potential loss of business.

Describe how each of the following ethical guidelines was breached by the data security company.

- | | |
|----------------------------|-----|
| (a) Integrity | [2] |
| (b) Responsibility | [2] |
| (c) Competence | [2] |
| (d) Professionalism | [2] |

9 [ACJC/PRELIM/9569/2021/P2/Q1]

The Universal Product Code (UPC) system is used for tracking trade items in shipping, inventory, and sales. Each item is given a 12-digit identification number. The validity of this identification number can be checked using a checksum. If x_i represents the i th digit (starting with $i = 1$ as the leftmost digit), then a valid identification number satisfies the condition that

$$3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12}$$

has a remainder of 0 when divided by 10.

The identification number can be encoded into a barcode. For this Task, the barcode will be represented as a string of '0's and '1's, where '0' represents a white stripe and '1' represents a black stripe.

The barcode is divided into seven sections. From left to right, they are

- A 'quiet zone' consisting of nine '0's;
- A start pattern which is always '101';
- The first six digits of the identification number are encoded using the table below;
- A middle pattern which is always '01010';
- The last six digits of the identification number are encoded using the table below;
- An end pattern which is always '101';
- A 'quiet zone' consisting of nine '0's.

The table below shows the encoding system for the digits. Note that depending on whether the digit occurs in the first six digits or the last six digits, it would be encoded differently. However, the two encodings are optical inverses of each other – a '0' is changed into a '1', and vice versa.

Digit	First six digits	Last six digits
0	'0001101'	'1110010'
1	'0011001'	'1100110'
2	'0010011'	'1101100'
3	'0111101'	'1000010'
4	'0100011'	'1011100'
5	'0110001'	'1001110'
6	'0101111'	'1010000'
7	'0111011'	'1000100'
8	'0110111'	'1001000'
9	'0001011'	'1110100'

The reason for encoding the first and last six digits differently is that the barcode may inadvertently be scanned upside down. Notice that in the first six digits, the encoding for each digit contains an odd number of '1's, while in the last six digits, the encoding for each digit contains an even number of '1's. This allows the scanning software to detect if the barcode has been placed upside down and correct it.

For example, the UPC identification number 036000 291452 would be encoded as:

000000000	101	0001101	0111101	0101111	0001101
Quiet	Start	0	3	6	0

0001101	0001101	01010	1101100	1110100	1100110
0	0	Middle	2	9	1

1011100	1001110	1101100	101	000000000
4	5	2	End	Quiet



(Notice that in an actual barcode, the stripes for the start, middle and end pattern are usually slightly longer than the surrounding stripes. This is to help humans to read it.)

Task 1.1

Write a function to determine the validity of any input string as an identification number. [5]

Task 1.2

Write a function to convert a valid identification number, given as a string, into a barcode (a string of '0's and '1's). [5]

Task 1.3

Write a function that takes in a string, check whether it represents a valid barcode, and converts it to an identification number if it does. Note that the barcode may be upside down. [11]

Download your program code and output for Task 1 as TASK1_<your name>_<centre number>_<index number>.ipynb

10 [ACJC/PRELIM/9569/2021/P2/Q2]

A file compression algorithm reduces file sizes so that files can be sent more quickly. One such algorithm is the Huffman algorithm for text files, which will be implemented in this task.

Unlike ASCII, which assigns a fixed size of 8 bits for each character, the Huffman algorithm assigns fewer bits to more common characters and more bits to less common characters. For example, in a long text written in English, characters such as 'e' and 't' will have fewer bits assigned to them than characters such as 'q' and 'z'. If the text is long enough, this will use fewer bits in total to encode the text compared to ASCII.

To know which sequence of bits to encode for each character, the **frequency** of each character, which is the number of times each character appears in the text file, is tabulated.

The characters are put into a tree. A node is created for each character. The following steps are then repeated until there is only one node without a parent:

- (a) Identify the two nodes, without parents, which have the lowest frequency.
- (b) Create a new node whose left and right children are the two nodes identified in Step 1. The frequency of the new node is the total of the frequency of its children.

The diagram on the following page shows the process of creation of a tree for a file with only five distinct characters ('A', 'E', 'I', 'O' and 'U'), in five stages.

The bit sequence assigned to a character will be the path from the root to the node corresponding to that character, where going left corresponds to '0' and going right corresponds to '1'. For example, 'A' is encoded as '10' and 'O' is encoded as '011'.

Task 2.1

Create a Node class that has the following attributes:

- `data`, which is determined when the node is initialized
- `left`, a pointer to another node,
- `right`, a pointer to another node

When the node is initialised, `left` and `right` do not point to anything. The class also has setter methods for `left` and `right`, and getter methods for all three attributes. [3]

Task 2.2

Write code that takes an input `.txt` file and creates a dictionary whose keys are the characters in the file, including spaces, punctuation and line breaks (`'\n'`), and the value of a key is its frequency in the file. Uppercase and lowercase letters should be considered as different characters.

Create a node for each character in the file, and put the nodes into a list in ascending order of frequency. [11]

Task 2.3

Create a tree using the algorithm described above. [5]

Task 2.4

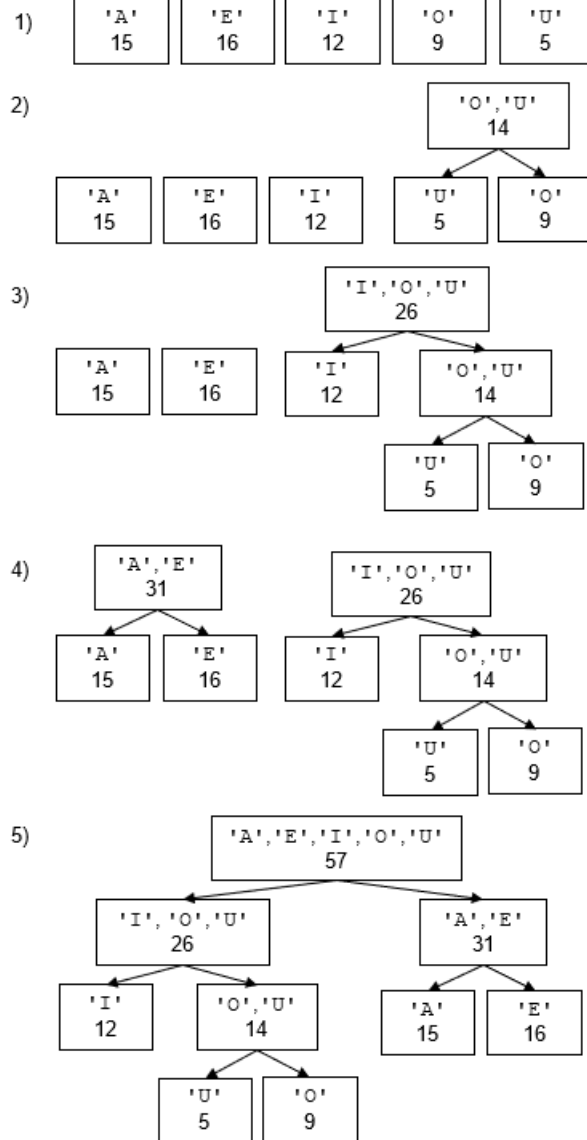
Create a dictionary whose keys are the characters, and the value of a key is the bit sequence of that character, expressed as a string of '0's and '1's.

Carry out Tasks 2.2 to 2.4 on the file `HAMLET.txt`. Compress the file by replacing each character with its bit sequence and writing the output to a new file, `HAMLET_compressed.txt`. [8]

Download your program code and output for Task 2 as `TASK2_<your name>_<centre number>_<index number>.ipynb`

Diagram showing how the tree is created based on the frequency of each character:

Character	Frequency
'A'	15
'E'	16
'I'	12
'O'	9
'U'	5



11 [ACJC/PRELIM/9569/2021/P2/Q3]

A community centre is required to keep COVID-19 vaccination records of its members in a NoSQL database. The CoviDie vaccine, which requires two doses to be taken at least 21 days apart, has been secured for all members of this particular community centre.

As everyone needs to be vaccinated before the end of 2021, we shall only consider the year 2021, which is a non-leap year. The table below shows the number of days available in the twelve months of 2021.

Month	01	02	03	04	05	06
Days	31	28	31	30	31	30

Month	07	08	09	10	11	12
Days	31	31	30	31	30	31

Task 3.1

Write a function `second_dose_date(date)` that:

- Takes a string value `date` in the format `YYYYMMDD`, where `YYYY` represents the year, `MM` represents the month and `DD` represents the day
- determines the date that is 21 days after the input date
- returns the result date in the format `YYYYMMDD`

Assume that the result date does not go beyond 20211231.

[3]

Test the function using the following three calls:

- `second_dose_date('20210105')`
- `second_dose_date('20210212')`
- `second_dose_date('20210919')`

[3]

Save your program code as

`TASK3_1_<your name>_<centre number>_<index number>.py`

Task 3.2

The list of members under the management committee of the community centre is stored in the text file `VACCINATION.txt`. Some members have not taken the vaccination at all, some others have only taken the first dose, while the rest have taken both doses.

Each line of the text file is of the following format: `_id,name,date_first_dose,date_second_dose,remarks`

- `_id` is a unique integer ID assigned to the member
- `name` is the name of the member
- `date_first_dose` and `date_second_dose`, if any, represent the date of the first dose and the date of the second dose respectively in the format `YYYYMMDD`
- `remarks`, if any, shows the pre-existing condition of the member

Write program code to insert the data from `VACCINATION.txt` into a NoSQL database `community_centre` under the collection `management_committee`. The program should clear the collection `management_committee` if it exists inside the database. [7]

Save your program code as `TASK3_2_<your name>_<centre number>_<index number>.py`

Task 3.3

The community centre needs a program to check the vaccination status of its members. The program should also allow for the downloading of vaccination certificates for members who are fully vaccinated, i.e. they have taken the two doses.

Write program code to:

- prompt the user to input a member ID, and keep prompting until the user keys in numeric character(s)
- if the member ID is available in the NoSQL database, perform either one of the following:
 - if the member is fully vaccinated, write the vaccination certificate to an output text file and update the record in the NoSQL database by including a field and an appropriate value to indicate that the certificate has been downloaded
 - if the member has only taken the first dose, output a message to show the date from which the member can take the second dose

- if the member has not taken the vaccination at all, output a message to tell that the member should take the first dose as soon as possible
- otherwise, if the member ID is not available in the NoSQL database, display an appropriate message and terminate the program

The format of the vaccination certificate is as follows.

VACCINATION CERTIFICATE

Name: <name>

Vaccine type: CoviDie

Date of first dose: <date_first_dose>

Date of second dose: <date_second_dose>

[8]

Save your program code as TASK3_3_<your name>_<centre number>_<index number>.py

Test your program for the member with `member_id = 24`.

[2]

The output text file should be saved as TASK3_3_<your name>_<centre number>_<index number>.txt

12 [ACJC/PRELIM/9569/2021/P2/Q4]

A company specialising in bento boxes wishes to trial a relational database management system to manage its data. It is expected that the database should be normalised to third normal form (3NF).

The company owns four kiosks. For each of the kiosks, the following information is to be recorded in the table *Kiosk*:

- *KioskID* – the unique integer assigned to the kiosk
- *Location* – the area where the kiosk is located
- *Rating* – the average rating of the kiosk between 0.0 and 5.0 inclusive

The company offers eight different types of bento boxes. Some of them may contain egg, nut, seafood or a combination of them. For each of the bento boxes, the following information is to be recorded in the table *BentoBox*:

- *BentoName* – the unique name of the bento box
- *ProductionCost* – the cost incurred in producing the bento box in dollars and cents
- *ContainEgg* – an integer 0 for not containing egg and 1 for containing egg
- *ContainNut* – an integer 0 for not containing nut and 1 for containing nut
- *ContainSeafood* – an integer 0 for not containing seafood and 1 for containing seafood

Each of the four kiosks sells all eight bento boxes at different mark-up prices. Another table *KioskBento* is needed to record the following information:

- *KioskID* – the unique integer assigned to the kiosk
- *BentoName* – the unique name of the bento box
- *SellPrice* – the price at which the bento box is sold at the kiosk in dollars and cents

Task 4.1

Create an SQL file called `TASK4_1_<your name>_<centre number>_<index number>.sql` to show the SQL code to create the database `bento_company.db` with the three tables. [5]

Save your SQL code as `TASK4_1_<your name>_<centre number>_<index number>.sql`

Task 4.2

The files `KIOSK.txt` and `BENTOBX.txt` contain information about the company's kiosks and bento boxes respectively for insertion into the database. Each row in the two files is a comma-separated list of information.

For `KIOSK.txt`, information about each kiosk is given in the following order: `KioskID`, `location`, `rating`

For `BENTOBX.txt`, information about each bento box is given in the following order: `BentoName`, `ProductionCost`, `ContainEgg`, `ContainNut`, `ContainSeafood`

The mark-up price for each kiosk has been set as follows:

- `KioskID` = 1 sells each bento box at a price that is \$2.60 higher than the production cost
- `KioskID` = 2 sells each bento box at a price that is \$2.90 higher than the production cost
- `KioskID` = 3 sells each bento box at a price that is \$2.40 higher than the production cost
- `KioskID` = 4 sells each bento box at a price that is \$3.10 higher than the production cost

Write program code to insert all the required information into the database `bento_company.db`.

[6]

Save your program code as `TASK4_2_<your name>_<centre number>_<index number>.py`

Run your program.

Task 4.3

The company wishes to create a form to display the bento boxes sold at a particular kiosk and their prices in a web browser. The form should allow customers to indicate egg, nut and seafood allergies, if any, and filter out the bento boxes that they cannot consume.

Write a Python program and the necessary files to create a web application that:

- receives input from a HTML form that includes:
 - a text box to enter the `location` of the kiosk
 - three checkboxes to indicate egg, nut and seafood allergies, if any
- returns a HTML document to display only the bento boxes that the customers can consume based on the allergies indicated, if any, and their prices for the given `location`

Input validation is not required.

[10]

Save your Python program as TASK4_3_<your name>_<centre number>_<index number>.py

with any additional files / sub-folders as needed in a folder named TASK4_3_<your name>_<centre number>_<index number>

Run and test the web application using the following input:

- 'Woodlands' entered as the location
- checkboxes indicating egg and seafood allergies ticked

[2]

Save the output of the program as TASK4_3_<your name>_<centre number>_<index number>.html

13 [DHS/PRELIM/9569/2021/P1/Q1]

A cosmic ray striking computer memory at just the right time can flip a bit, turning a 0 into a 1 or vice versa.

Bit flips had caused plane accidents, software glitches, and at times, the Blue Screen of Death (BSOD) on personal computers.

- (a) The hexadecimal number D4 had been changed to 9A. Suggest the minimum number of times a bit flip could have affected it and explain why. [2]
- (b) Explain why Unicode characters, as compared to ASCII characters, are more likely to be altered after a cosmic ray strike. [2]
- (c) Define what a backup and archive is, and explain which should be prioritised if a company is warned of a global cosmic ray strike occurring in a week. [4]
- (d) Explain how TCP/IP layers help to fulfil the purpose of the TCP/IP model. [2]
- (e) Suggest and explain a consequence on network communication when a data packet is affected by a bit flip occurring at the Internet Layer [2]
- (f) Explain why the consequence of a bit flip occurring at the Internet Layer and the Transport Layer would be the same as the consequence as a bit flip occurring at the Internet Layer only. [1]

14 [DHS/PRELIM/9569/2021/P1/Q2]

An array stores 16 powers of 2 integers in ascending order:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768

- (a) If the array is searched by means of a binary search, state which elements would be accessed, and in what order,
- (i) when searching for the number 4096 (which is present), and [1]
 - (ii) when searching for the number 3 (which is not present). [1]
- (b) Draw a program flowchart of an iterative binary search algorithm. [6]
- (c) Explain why binary search could be more efficient than linear search. [1]
- (d) State the time complexities of hash table search and binary search and explain which is more efficient with reference to their time complexities in the above scenario. [4]

15 [DHS/PRELIM/9569/2021/P1/Q3]

A mobile network provider's management of customer's overdue bills include an automated emailing and SMS system.

- If a mobile bill is overdue, a daily system generated reminder is emailed to the user indicating the overdue details.
- If the user has 1 mobile bill that is more than 6 days overdue, in place of the daily reminder email, a warning email and SMS will be sent to the user.
- If a user has more than 4 bills which are more than 14 days overdue, the user will incur a penalty fee for every additional week starting from the 14th day of being overdue.
- If a user has incurred more than 3 penalty fees, the user's mobile service will be terminated.
- Penalty fees could be incurred as a result of other actions such as accessing illegal websites.
- Each bill issued to a customer represents the past month's usage.

(a) Create a decision table showing all possible outcomes and results for bill(s) that are overdue. [4]

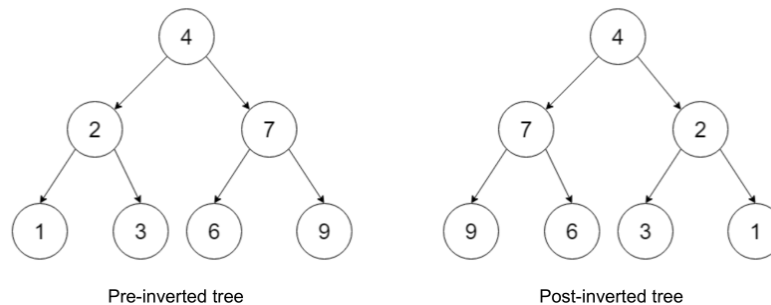
(b) Simplify the decision table by removing redundancies. [3]

(c) A user interface is designed for a program for customers to check their bills and overdue status. State a usability principle and describe how the user interface of the program should be designed to demonstrate this principle. [2]

16 [DHS/PRELIM/9569/2021/P1/Q4]

- (a) Explain how a linked list data structure could be more suitable than an array data structure to implement a binary tree. [2]
- (b) (b) Suggest and justify one circumstance where an array structure is more appropriate than a linked structure. [2]

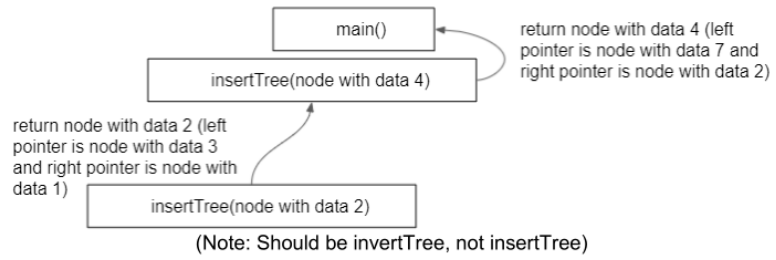
The diagram shows a binary tree before and after an inversion.



Each node has these attributes:

- `right` which is the pointer to right subtree
- `data` which is the value contained in each respective node displayed above
- `left` which is the pointer to the left subtree

- (c) Write pseudocode for procedure `insert` which will add a node to the post-inverted tree (which may be empty) in such a way that if the new value of the node is **less** than the value at the current node, the new node will be added to the **right** subtree, or else it will be added to the left subtree. `insert` takes in values `node_data` and `root_node` which is the value to be added to the tree and the instance of the root node of the tree respectively. [6]
- (d) A function `invertTree` takes in the root node of the above pre-inverted tree, uses recursion to invert it into the post-inverted tree, and returns the root node of the post-inverted tree. Write pseudocode for the function and visualise it in a trace diagram. An incomplete trace diagram is provided for you to begin with. Copy and complete it.



(Note: Should be invertTree, not insertTree)

[6]

17 [DHS/PRELIM/9569/2021/P1/Q5]

Consider the following data which shows a single Civics Group record used in COVID19 vaccination tracking.

Civics Group Name: 6C35 Civics Tutor: Mr Tan Chairperson: Steve Lim				
Register Number	Student Name	Vaccine brand name	CCA Name	CCA Teacher
11	Steve Lim	Moderna	Choir	Mr Lee
4	Melody Tan	Moderna	Choir	Mr Lee
19	Paul Chang	Pfizer–BioNTech	Soccer	Mr Chan
3	Grace Lee	Pfizer–BioNTech	Dance	Ms Deepa
20	Alison Chong	Not Vaccinated	Infocomm Club	Mrs Ho
12	Vincent Eng	Moderna	Soccer	Mr Chan

- (a) Derive a set of tables to show the above data in first, second and third normal form. [6]
- (b) Draw an ER diagram for a normalised database design. [2]
- (c) Using examples in the above context, explain the significance of the following terms:
- (i) primary key [2]
- (ii) foreign key [2]
- (d) With reference to the above context, describe or suggest a scenario where a NoSQL database would be more appropriate. [1]

18 [DHS/PRELIM/9569/2021/P1/Q6]

Long-distance optical fibre lines and submarine cables which are a vital part of the global internet infrastructure are vulnerable to solar superstorms which happen once in a century. The last solar superstorm was in 1921.

- (a) State and explain why websites would or would not be accessible by web browsers if a solar superstorm shuts down all DNS servers. [2]
- (b) Explain packet switching and its importance in ensuring global internet connectivity when some parts of the earth are hit by a solar superstorm. [4]

An international company based in many countries updates its network structure to ensure Internet connectivity during solar superstorms.

Employees can now access files on a shared virtual space which is made up of 3 servers located in the United States, Europe, and Asia. All servers hold identical information (any changes made on one server would update the other servers immediately) so even if one server is affected by the solar superstorm, employees still can access their files on the other servers.

Employees must be connected physically to the company's intranet network at each country's office to access the virtual space.

- (c) Draw a network diagram of the above configuration and label the LAN, internet, router(s), WAN link(s), intranet, servers, and employee laptops. [5]
- (d) Why is version control vital when employees from different countries work in teams? [1]

19 [DHS/PRELIM/9569/2021/P1/Q7]

A divide and conquer approach is used by merge sort to successively divide a list into half, forming two sublists, until each sublist is of length 1. The sublists are then sorted and merged into larger sublists until they are recombined into a single sorted list. An algorithm for merge sort to perform an ascending sort is given below. It will be used to sort large or small data sizes.


```

01 PROCEDURE mergesort(mergelist : ARRAY)
02
03     IF LENGTH(mergelist) > 1 THEN
04
05         mid ← LENGTH(mergelist) DIV 2
06
07         FOR index ← 0 TO (mid - 1)
08             lefthalf[index] ← mergelist[index]
09         NEXT index
10
11         right_len ← LENGTH(mergelist) - mid
12
13         FOR index ← 0 TO (right_len - 1)
14             righthalf[index] ← mergelist[right_len + index]
15         NEXT index
16
17         mergesort(lefthalf)
18         mergesort(righthalf)
19
20         i ← 0
21         j ← 0
22         k ← 0
23         WHILE i < LENGTH(lefthalf) AND j < LENGTH(righthalf)
24             IF lefthalf[i] > righthalf[j] THEN
25                 mergelist[k] ← lefthalf[i]
26                 i ← i + 1
27             ELSE
28                 mergelist[k] ← righthalf[j]
29                 j ← j + 1
30             ENDIF
31             k ← k + 1
32         ENDWHILE

```

33
Generated with Python Script by BRW

```

34         WHILE i < LENGTH(lefthalf)
35             mergelist[k] ← lefthalf[i]
36             i ← i + 1
37             k ← k + 1

```

(a) The following array of numbers is to be sorted using mergesort:

```
mergelist = [2, 4, 2, 8, 2, 8, 9, 1, 3]
```

What are the first two lists to be merged?

[1]

(b) Explain what a logic error is, give the line number for the logic error in the above code, and rewrite the line correctly.

[2]

The procedure `sorting_proc` uses an optimised bubble sort to sort an array `input_array` in an ascending order. It is used within `modified_mergesort` which is a modified version of `mergesort`.

```
01 PROCEDURE sorting_proc(input_array : ARRAY)
02
03     length ← LENGTH(input_array)
04
05     REPEAT
06         swapped ← FALSE
07
08         FOR curr_elem_index ← 1 to length - 1
09
10             IF input_array [curr_elem_index - 1] > input_array [curr_elem_index] THEN
11                 SWAP (input_array [curr_elem_index - 1], input_array [curr_elem_index])
12                 swapped ← TRUE
13             ENDIF
14
15         ENDFOR
16
17         length ← length - 1
18
19     UNTIL NOT swapped
20
21 ENDPROCEDURE
```

```

01 PROCEDURE modified_mergesort(mergelist : ARRAY)
02
03     IF LENGTH(mergelist) > 1 THEN
04
05         IF LENGTH(mergelist) < 5 THEN
06             sorting_proc(mergelist)
07             RETURN
08
09         ELSE
10
11             mid ← LENGTH(mergelist) DIV 2
12
13             FOR index ← 0 TO (mid - 1)
14                 lefthalf[index] ← mergelist[index]
15             NEXT index
16
17             right_len ← LENGTH(mergelist) - mid
18
19             FOR index ← 0 TO (right_len - 1)
20                 righthalf[index] ← mergelist[right_len + index]
21             NEXT index
22
23             mergesort(lefthalf)
24             mergesort(righthalf)
25
26             i ← 0
27             j ← 0
28             k ← 0
29             WHILE i < LENGTH(lefthalf) AND j < LENGTH(righthalf)
30                 IF lefthalf[i] > righthalf[j] THEN
31                     mergelist[k] ← lefthalf[i]
32                     i ← i + 1
33                 ELSE
Generated with Python Script by BRW
34                     mergelist[k] ← righthalf[j]
35                     j ← j + 1
36                 ENDIF
37                 k ← k + 1

```

- (c) Would the above modification of mergesort improve the algorithm's overall efficiency? Support your answer with a description on how and explanation on why its efficiency is affected. [4]

The procedure `insertionSort` is an algorithm which uses insertion sort.

```

01 PROCEDURE insertionSort(input_array: ARRAY)
02
03     current_elem_index ← 0
04
05     REPEAT
06         current_elem_index ← current_elem_index + 1
07         compared_item_index ← -1
08         swapped ← FALSE
09
10         REPEAT
11             compared_item_index ← compared_item_index + 1
12
13             IF input_array[current_elem_index] < input_array[compared_item_index] THEN
14
15                 temp ← input_array[current_elem_index]
16
17                 the value of each element of input_array from compared_item_index to
(current_elem_index - 1) is sequentially assigned to each element of input_array from
(compared_item_index + 1) to current_elem_index
18
19                 input_array[compared_item_index] ← temp
20
21                 swapped ← TRUE
22             ENDIF
23
24         UNTIL swapped ← TRUE
25
26     UNTIL current_elem_index = LENGTH(input_array) - 1
27
28 ENDPROCEDURE

```

- (d) Modify insertionSort and sorting_proc to count and store the number of comparisons made in a variable named comparisons. Instead of copying all the pseudocode statements, state the line number(s) you want

to modify or insert any pseudocode at, followed by the pseudocode statement(s) to be added/modified. [3]

- (e) Trace the modified algorithms `insertionSort` and `sorting_proc` for the array [5, 2, 3, 4] showing the value of all variables for each step by completing the following tables.

Trace table for `insertionSort`:

current_elem_index	compared_item_index	comparisons	input_array	swapped
1	0	1	[2,5,3,4]	TRUE
2	0	2	[2,5,3,4]	FALSE
...	

Trace table for `sorting_proc`:

current_elem_index	compared_item_index	input_array	arr_length	swapped
1	1	[2,5,3,4]	4	TRUE
2	2	[2,5,3,4]	4	FALSE
...

[7]

- (f) In the context of `mergesort`, suggest scenario(s) where using the current optimised bubble sort algorithm for `sorting_proc` would be better than using the `insertionSort` algorithm above. Support your answer by designing 3 test cases (normal and boundary) and comparing the number of `comparisons` made by each algorithm for each test case. Display your output. [7]

20 [DHS/PRELIM/9569/2021/P2/Q1]**Task 1**

For this task, submit all code as T1_<index number>_<name>.py.

The Taliban's takeover of Afghanistan had seen the United States (US) evacuating people. The Republic of Singapore Air Force offers its A330 Multi-Role Tanker Transport plane (A330 MRTT) to help the US airlift groups of evacuees from Afghanistan.

Task 1.1

To prevent Taliban interference, messages sent between evacuees and the A330 MRTT are encrypted.

The encryption converts each character of the message into its ASCII number representation, adds the ASCII number representation of a character of a secret key which is in the same position, and then converts it back into ASCII text.

Secret keys are generated from passwords known only by the sender and receiver. Passwords that are shorter in length than the message are lengthened to match by repeating the password until the same length as the message is achieved.

For example, given that the inputs are

```
password : "cat"
```

```
message  : "hello"
```

Step 1 - Extend password by repeating until matches length of message

```
"cat" -> "catca"
```

Step 2 - Convert each character of password and message into ASCII number representation

```
"hello" -> 104, 101, 108, 108, 111
```

```
"catca" -> 99, 97, 116, 99, 97
```

Step 3 - Add the ASCII number representation of same positions together

```
104+99, 101+97, 108+116, 108+99, 111+97 -> 203, 198, 224, 207, 208
```

Step 4 - Convert each number to its ASCII character

```
203, 198, 224, 207, 208 -> "ËÆäİĐ"
```

Therefore, the results are

Secret key : "catca"

Encrypted message : "ËÆàİĐ"

Write a function `generateKey(password, message)` that takes in two argument strings `message` and `password` and returns the secret key.

Test your function using `generateKey("cat", "hello")` and show your output. [4]

Task 1.2

Write functions `encrypt(password, message)` and `decrypt(password, message)` that takes in `password` and `message` strings and returns the encrypted or decrypted message. It should use the above function `generateKey(password, message)` to obtain the secret key before performing the encryption.

Test your function with `decrypt("cat", encrypt("cat", "hello"))`. Show your output. [6]

Task 1.3

Since the A330 MRTT plane can only take a maximum of 266 passengers at once, implement a `Queue` to manage incoming pickup requests and dropoffs. Write a class `Queue` with the following methods:

- `enqueue(string)` which takes in a string and adds it to the queue or returns "Queue is full!" if queue is full
- `dequeue` which removes the next item for the queue or returns "Queue is empty!" if the queue is empty
- `display` which returns a string of queue items in sequence from head to tail or returns "Queue is empty!" if the queue is empty
- `current_size` which returns an integer of the number of items in the queue. [8]

Task 1.4

Using socket programming, write the client program for the evacuee teams (client) in Afghanistan and the A330 MRTT plane (server) to communicate. The program should

- use the above queue data structure to manage the requests
- request the user to set a password for encryption at the start of the program

- present the user with incoming pickup requests and pickup details (Team name and group size) as well as queue details (size of current queue)
- allow the user(server) to accept/reject pickup requests
- encrypt all information sent and received using `encrypt(password, message)` and `decrypt(password, message)` functions.

You are provided with the client program in `client.py`. Write and submit the server program as `T1_<index number>_<name>.py`.

Study the following sample program output to determine your code design, output format and socket protocol.

User inputs are underlined.

- 2- 2Sample client programs (in sequence):

CLIENT 1

Please set a password.

Answer: WDIB4

What is your Team Name?

Answer: Team1

Group size?

Answer: 143

Establishing connection...

Connection established!

Data sent!

Waiting for the server to confirm your request...

Pickup confirmed! Please wait for pickup.

Connection disconnected.

CLIENT 2

Please set a password.

Answer: WDIB4 solver("((1*7)+6)")

What is your Team Name?

Answer: Team2

Group size?

Generated with Python Script by BRW

Establishing connection...

Connection established!

Data sent!

21 [DHS/PRELIM/9569/2021/P2/Q2]**Task 2**

Name your Jupyter Notebook and save all parts for this task as

TASK2_<index_number>_<name>.ipynb

You will be writing a Math game program for Form Teachers to play with students in school. The game auto-generates math expressions and tracks scoring.

Task 2.1

Using a stack data structure, write a function `solver(expr)` that takes in a string of a mathematical expression `expr` such as `"((1*7)+6)"` and returns 13. You may assume that the entire expression would never have spaces, and would always be enclosed in an opening and closing parenthesis `"()"`. Do not use the built-in function `eval()`. Test your function with `solver("((1*7)+6)")` and show your output. [6]

Task 2.2

Write a function `generate_expression` that takes in an integer `operator_count` and returns a string of a mathematical expression which has the specified number of operators (i.e. `+` `-` `*` `/` `)` in `operator_count`. The function should use **recursion** to form up the operators and operands. The operands, operators and positions of operands and operators should be random.

For example, `generate_expression(5)` would output `"(4*(6-(2+((1*7)+6))))"` and `generate_expression(5)` would output `"((8+(6+((2*4)-3)))+6)"`.

Test your function with `generate_expression(5)` and show your output. [7]

Task 2.3

Implement the following using object-oriented programming:

- Person, a class, which
 - initialises with these attributes
 - * `name:` `string`
 - * `gender:` `string` where male is "M" and female is "F"

- * `score: integer`
- has the following methods
 - * `display_info()` which displays the Person's name, gender and score
 - 1. eg "Nelson(M)'s score is 3."
 - * `attempts()` which
 - 1. uses the function `generate_expression` from Task 2.2 to generate and display a random math expression of 2 operators
 - 2. queries the user to give an answer rounded up to the nearest integer
 - 3. displays "Good job!" if the input is correct or "Wrong answer. (Correct answer: <answer>)" where <answer> is the correct answer.
 - 4. increases the score of the student by 1 if the answer is correct
 - 5. displays the user's latest score
- `Student`, a subclass of `Person`, which
 - also has the following attribute
 - * `role: string` which is "no role" by default unless the `Student` has a class committee role such as "chairperson"
 - also has the following methods
 - * `student_role()` which returns a string describing the role of the `Student`
- `FormTeacher`, a subclass of `Person`, which
 - also has the following methods
 - * `display_info()` which uses polymorphism to display the `FormTeacher`'s information with salutation to the `FormTeacher`'s name
 - 1. eg: "Ms. Norah's score is 0." where "Norah" is her name, and "Ms." corresponds to her gender.

[14]

Task 2.4

Write driver code to test the earlier class you created. Also, create `groups` which is a list that uses a 2-dimensional array to store and associate each instance created below with his/her civics group. Use this 2-dimensional array to display the scores of all persons in each civics group indicating the student chairperson's name (if any). Test your code with the following steps in order:

- Create an instance of `Student` with name "Melvin" in civics group 5C35
- Create an instance of `Student` with name "Susan" in civics group 5C35 whose role is "chairperson"
- Create an instance information of `FormTeacher` with name "Norah" in civics group 5C35
- Create an instance of `Student` with name "Ben" in civics group 6C35
- Create an instance of `FormTeacher` with name "Jimmy" in civics group 6C35
- Display the information of Melvin
- Display the information of Susan
- Display the information of Norah
- Melvin attempts a math question
- Susan attempts a math question
- Jimmy attempts a math question
- Display the scores of all persons in each civics group with a header for each class

Here is a sample of an expected output:

Generated with Python Script by BRW

Melvin(M)'s score is 0.

Susan(F)'s score is 0.

Ms. Norah's score is 0.

To Melvin : $((5/7)+8)$?

Answer:1

Wrong answer. (Correct answer: 9)

Total score is still 0.

To Susan : $(3*(5*6))$?

Answer:90

Good job!

New total score for Susan is 1

To Jimmy : $((6-2)-6)$?

Answer:3

Wrong answer.

(Correct answer: -2)

Total score is still 0.

5C35's scores:

Melvin(M)'s score is 0.

Susan(F)'s score is 1. (Chairperson)

Ms. Norah's score is 0.

6C35's scores:

Ben(M)'s score is 0.

Mr. Jimmy's score is 0.

Run your program and save your output.

[7]

22 [DHS/PRELIM/9569/2021/P2/Q3]**Task 3**

In 2021, Singapore's Health Science Authority (HSA) recalled 18 brands of hand sanitisers due to high levels of acetaldehyde and/or methanol. The HSA keeps information on current hand sanitisers and uses it to monitor the types of chemical ingredients used to make the sanitisers.

Task 3.1

Create an SQL file to show the SQL code to create database `sanitisers.db` with the single table, `sanitisers`.

The table will have the following fields:

- `product_name` which is the primary key
- `active_ingredient`
- `alcohol-based`

Save your SQL code as

`TASK3_<index_number>_<name>.sql`

[3]

Task 3.2

The text file, `sanitisers.txt`, contains data items for a number of sanitisers. It contains a header line. Each data item is separated by a comma, with each item data on a new line, as follows:

- product name
- active ingredient used to make the sanitiser product
- "Yes" or "No" to indicate if the product is alcohol-based

Write program code to read in the information from the text file, `sanitisers.txt`, and insert all the information into the `sanitisers.db` database.

[3]

Run the program.

Save your program as

`TASK3_<index_number>_<name>.py`

Task 3.3

The information is to be displayed in a web browser.

Write a python program and the necessary files to create a web application that enables the list of sanitisers to be displayed.

For each record the web page should include the:

- product name
- ingredients used to make the sanitiser product
- "Yes" or "No" to indicate if the product is alcohol-based

Save your program as

TASK3_<index_number>_<name>.py

with any additional files/subfolders as needed in a folder named

TASK3_<index_number>_<name>

Run the web application and save the output of the program as

TASK3_OUTPUT_<index_number>_<name>.html

[6]

Task 3.4

HSA wants a form on the web page that allows users to enter in the name of an active ingredient and, upon submission, will display all the information of the products with the matching active ingredient.

Update your application to include this form feature so that users will be able to use the form after seeing the list of sanitisers displayed as required in Task 3.3.

Run the web application, test your program with the ingredient "Triclosan" and save the output.

[4]

Save, zip up and submit your program code and all related files for Task 3 as

TASK3_<index_number>_<name>.zip

23 [DHS/PRELIM/9569/2021/P2/Q4]**Task 4**

Name your Jupyter Notebook and save all parts for this task as

TASK4_<index_number>_<name>.ipynb

Task 4.1

Write a program to help staff of an events company to insert data into a NoSQL database products under the collection balloons.

The data is provided for you in balloons.json as well as in the table below where the first row are headers for the fields.

design	amount	helium	colours
car	88	no	red, yellow
cloud	14		blue, green
flower	75	yes	red, blue
bag	38	no	red, blue, black

Each colour in colours field should be an item in an array.

[6]

Task 4.2

Write code to print the amount of the product with the design "car".

[2]

Task 4.3

Write code to update the field helium to have the value "no" for all documents which do not have a field or value for helium.

[3]

Task 4.4

Write code to display the design(s) which do not contain helium and have colours that either contain green or do not contain black.

[3]

Run the program.

24 [HCI/PRELIM/9569/2021/P1/Q1]

An E-Commerce company stores the following data of customers in the system.

- Name
- Contact
- Address

It categories its customers into 2 types of loyalty programs.

- Spend-based loyalty program
- Paid loyalty program

Customers of Spend-based loyalty program earn one point for every block of \$10 spent in a single order, whereas customers of Paid loyalty program pay a monthly or annual fee. Customers of Paid loyalty program will enjoy the benefits of having early access to sales events and free delivery for purchases above \$30.

For Spend-based loyalty program, the additional data stored include:

- Points earned

For Paid loyalty program, the additional data stored include:

- Payment schedule (monthly or annually) and corresponding fee
- Next payment date, computed based on payment schedule and the date of enrollment to the program

Object-oriented programming will be used to model the customers.

(a) Draw a class diagram that shows the following for the requirement described above.

- the superclass
- any subclasses
- inheritance
- properties
- appropriate methods

[6]

The company makes changes to the Paid loyalty program to allow the customer in the program to earn ten points for every block of \$20 spent in a single order, in addition to the current benefits. The points earned do not expire. For Spend- based loyalty program, all points earned will expire on the anniversary of the date of enrolment to the program.

(b) Suggest changes required to the class diagram to enable the changes. [3]

(c) Explain why inheritance is an important feature of object-oriented programming. [1]

To attract customers to enrol to its Paid loyalty program, the company launches an invitation campaign to invite Spend-based loyalty program customers who qualified the following conditions:

- Customer who earned more than 2000 points in a year and has an average of at least one order per month will be contacted by staff.
- Customer who has enrolled for at least a year and has an average of at least one order per month will be sent an invitation email.
- Otherwise, no invitation will be sent.

(d) Create a decision table showing all the possible outcomes and results. [4]

(e) Simplify your decision table by removing redundancies. [2]

25 [HCI/PRELIM/9569/2021/P1/Q2]

Merge Sort is a Divide and Conquer algorithm. It divides the unsorted array $A[\text{low}.. \text{high}]$ into two halves, calls itself for the two halves, until each half is of length 1. It then merges the two sorted halves. An algorithm for Merge Sort is given below.

```
PROCEDURE MergeSort(A, low, high)
```

```
    IF low < high
```

```
        mid ← (low + high) DIV 2
```

```
        MergeSort(A, low, mid)
```

```
        MergeSort(A, mid+1, high)
```

```
        Merge(A, low, mid, high)
```

```
    ENDIF
```

```
ENDPROCEDURE
```

- (a)** Write in **pseudocode**, an algorithm for the merge procedure, $\text{Merge}(A, \text{low}, \text{mid}, \text{high})$ that is called by the MergeSort algorithm. The merge procedure should merge the sorted subarrays in $A[\text{low}.. \text{mid}]$ and $A[\text{mid}+1.. \text{high}]$ into a single sorted subarray in $A[\text{low}.. \text{high}]$. [6]
- (b)** Give and justify the time complexity of Merge Sort. [2]

26 [HCI/PRELIM/9569/2021/P1/Q3]

An abstract Data Type (ADT) consists of both data type and associated operations.

A stack ADT has the following operations defined:

- Create(S) --- creates an empty stack S,
- Insert(S, Item) --- inserts new value, Item, onto stack S,
- Retrieve(S) --- removes and returns item from the stack S,
- EmptyStack(S) --- returns true if stack S is empty.

(a) Devise an algorithm that converts a non-negative integer from decimal to hexadecimal, by making use of the stack operations given above. [4]

(b) Three items, L1, L2 and L3, are to be inserted into a stack in its original order, but the output would be in the order of L1, L3 and L2.

Write an algorithm, using the operations given above, that would use a stack R to carry this out. [4]

27 [HCI/PRELIM/9569/2021/P1/Q4]

Some algorithms can be written using recursion.

- (a) State **two** features of recursion. [2]
- (b) Explain the use of a stack when the recursive procedure executes. [3]
- (c) Write a recursive function using **pseudocode** that returns the sum of the digits in an integer. For example, the sum of the digits of the integer 12345 is $5+4+3+2+1=15$. [4]

28 [HCI/PRELIM/9569/2021/P1/Q5]

(a) Vaccination centres are located across the island to facilitate the national vaccination programme. At each vaccination centre, data is uploaded to the central system of Ministry of Health.

(i) State the name of this network structure. Describe **one** disadvantage and suggest **one** method to resolve it. [3]

(ii) Describe **two** rules of conduct for the staff handling data. [2]

(b) Explain each of the following terms and how it works:

(i) Digital signature [7]

(ii) Transmission Control Protocol [3]

(iii) Domain Name System [2]

29 [HCI/PRELIM/9569/2021/P1/Q6]

Check digit is one technique of data validation.

(i) Give **two** other techniques of data validation. [2]

(ii) With **one** example of data verification, explain the difference between data verification and data validation.

[3]

A student ID consists of 5 digits and a check digit.

(iii) One way to calculate the check digit is to use the unit's digit of the sum of all 5 digits. For example, suppose the 5 digits are 50879. Since $5 + 0 + 8 + 7 + 9 = 29$, the check digit is 9, and the student ID is 508799.

Explain, with **two** examples, why this method is inadequate. [2]

The check digit is calculated from the 5 digits using the modulus 11 system. It can be digits 0 - 9 or character 'X'.

(iv) Showing your working, determine the check digit for 30526. [3]

(v) Write an algorithm to check if a student ID is valid. [5]

(vi) A function is designed to read a student ID and determine if it is valid. State the data types of its input parameter and justify. [2]

30 [HCI/PRELIM/9569/2021/P1/Q7]

- (a) (i) What is a flowchart? [1]
- (ii) Draw a flowchart to find the factorial of a given positive integer N . [2]
- (b) You have a row of $2n$ disks of two colors, n black and n white. They alternate: black, white, black, white, and so on. You want to get all the black disks to the right-hand end, and all the white disks to the left-hand end. The only moves you are allowed to make are those that interchange the positions of two neighboring disks.



Assume that there is an array A of size $2n$ representing the alternating disks. Write, in **pseudocode**, an algorithm to solve this puzzle and determine the number of moves it takes. [5]

31 [HCI/PRELIM/9569/2021/P1/Q8]

The school is designing a website to allow ordering of meal. The database stores data about

- students
- meal information
- order information

An order contains one meal only.

Each meal can be purchased by different students.

A student never places more than one meal on any day.

The data is stored in a relational database.

Table: ORDER

Student ID	Student Name	Class ID	Class Name	OrderDate	Meal ID	MealDescription	Price
67	John	55	21S66	15-04-2021	16	Chicken rice with orange juice	4.00
				21-04-2021	23	Japanese Bento with green tea	5.00
				13-05-2021	30	Fried Mee with apple juice	4.00
54	Peter	57	21S67	18-04-2021	32	Fried rice with orange juice	4.00
				25-04-2021	30	Fried Mee with apple juice	4.00
32	Mary	59	21S69	16-04-2021	23	Japanese Bento with green tea	5.00
				30-04-2021	5	Big burger meal set	5.00
				15-05-2021	16	Chicken rice with orange juice	4.00
73	Jean	62	21S68	18-04-2021	23	Japanese Bento with green tea	5.00
				28-04-2021	16	Chicken rice with orange juice	4.00
				14-05-2021	23	Japanese Bento with green tea	5.00

(a) Explain why the table is not in first normal form (1NF).

[1]

The following is an attempt to reduce data redundancy:

Table: Student

StudentID	StudentName	ClassID	ClassName
67	John	55	21S66
54	Peter	57	21S67
32	Mary	59	21S69
73	Jean	62	21S68

Table: Meal

MealID	MealDescription	Price
16	Chicken rice with orange juice	4.00
23	Japanese Bento with green tea	5.00
30	Fried Mee with apple juice	4.00
32	Fried rice with orange juice	4.00
5	Big burger meal set	5.00

Table: Order

StudentID	MealID	OrderDate
67	16	15-04-2021
67	23	21-04-2021
67	30	13-05-2021
54	32	18-04-2021
54	30	25-04-2021
32	23	16-04-2021
32	5	30-04-2021
32	16	15-05-2021
73	23	18-04-2021
73	16	28-04-2021
73	23	14-05-2021

- (b) State suitable primary key(s) for each table. [3]
- (c) Explain the reasons for reducing data redundancy in a relational database. [2]
- (d) Draw an entity-relationship (E-R) diagram showing the degree of the relations. [2]
- (e) State which table is not in third normal form (3NF) and explain why. [2]

A table description can be expressed as:

TableName (Attribute1, Attribute2*, Attribute3, ...)

The primary key is indicated by underlining one or more attributes. Foreign keys are indicated by using a dashed underline/asterisk.

- (f) Write table descriptions for the required tables in the databases so they are in third normal form (3NF). [4]
- (g) Write an SQL query to output the student names and date of order of all the orders for the meal "Japanese Bento with green tea". [3]

32 [HCI/PRELIM/9569/2021/P2/Q1]

Name your Jupyter Notebook as

Task1_<your name>_<centre number>_<index number>.ipynb

For each of the sub-tasks, add a comment statement, at the beginning of the code using the hash symbol '#', to indicate the sub-task the program code belongs to, for example:

2*In[1]:	# Task 1.1 Program Code
2*In[2]:	# Task 1.2 Program Code
2*In[3]:	# Task 1.3 Program Code
Output:	

Task 1.1

The file `INTEGERS.txt` stores 100 integers. Write a program to read the integers, arrange them in ascending order using quick sort, and write the sorted integers to a file called

`SORTED_<your name>_<centre number>_<index number>.txt` [15]

Task 1.2

Write a function `BinarySearch(list_of_integers, target)` that

- takes a list of ascending integers, `list_of_integers` and an integer `target`
- performs a binary search
- prints out if `target` is found in `list_of_integers`
- returns the number of comparisons during the binary search

[8]

Task 1.3

Write a program to read the list of integers from

`SORTED_<your name>_<centre number>_<index number>.txt`

obtained in Task 1.1. Generate 50 random integers between 1 and 200 (inclusive) and perform a binary search for each of these random integers in this sorted list. Output the average number of comparisons of these 50 binary searches. [2]

Save your Jupiter Notebook for Task 1.

33 [HCI/PRELIM/9569/2021/P2/Q2]

FlexiMSG provides messaging services. Information of the messages are logged into a file. The log records contain the phone numbers or IP address of the sender, the date which the service is being accessed, the status indicating whether the message has been sent and the type of application used. There are two different formats used:

<IP address> <DD/MMM/YYYY> <Status> <App>

or

<Phone number> <DD/MMM/YYYY> <Status>

Below is the log records in the file, LOG.txt:

```
- 2- 254.36.149.41 22/Jan/2021 200 WA
    188.226.164.216 22/Jan/2021 0 FB
    92783423 22/Jan/2021 200
    188.226.164.216 23/Jan/2021 0 FB
    88188293 23/Jan/2021 0
```

Task 2.1

Write the SQL code to create database ServiceLog.db with the single table, Log.

The table will have the following fields of the given SQLite types:

- LogID - primary key, an auto-incremented integer
- Sender - the client internet address or phone number, text
- AccessDate - the access date, text
- Status - the status, integer
- AppType - application type, text

Save your SQL code as

Task2_1_<your name>_<center number>_<index number>.sql

[2]

Task 2.2

FlexiMSG wants to use Python programming language and object-oriented programming to update the information in the log file into the database.

Create the class `ServiceRecord` that will store the following:

- `Sender` - stored as a string
- `AccessDate` - stored as a string
- `Status` - stored as integer, 0 or 200
- `AppType` - stored as string value 'WA' or 'FB'

The class has the following methods:

- `isSuccess()` - returns a Boolean value to indicate whether the message has been sent.
 - returns `True` if the `Status` is 200, otherwise returns `False`
- `getAppType()` - returns a string value to indicate the type of messaging application.
 - returns the value of `AppType`

Write program code to read in the information from `LOG.txt`, creating an instance of the `ServiceRecord` class for each record and insert the information into `ServiceLog.db` database.

Save your program code as `Task2_2_<your name>_<center number>_<index number>.py`

[8]

Task 2.3

FlexiMSG wants to publish the database content on a web page.

Create class `AppServiceRecord` which inherits from `ServiceRecord`, such that:

- `getAppType()` - returns the following values based on the value of `AppType`
 - WA - returns 'WHATSAPP'
 - FB - returns 'FACEBOOK MESSENGER'
- `getSuccess()` - returns the following values based on the returned value of `isSuccess()`
 - `True` - returns 'SUCCESS'
 - `False` - returns 'FAILED'

Create class `SmsServiceRecord` which inherits from `ServiceRecord`, such that:

- `getAppType()` - always returns 'SHORT MESSAGE SERVICE'
- `getSuccess()` - returns the following values based on the returned value of `isSuccess()`

- True - returns 'SUCCESS'
- False - returns 'FAILED'

Save your program code to Task2_3_<your name>_<center number>_<index number>.py

[4]

Task 2.4

Write a Python program and the necessary files to create a web application that enables the list of log records to be displayed.

For each record, the web page should include the:

- Sender
- AccessDate
- AppType (either WHATSAPP, FACEBOOK MESSENGER or SHORT MESSAGE SERVICE)
- Status (SUCCESS or FAILED)

Save your program code as

Task2_4_<your name>_<center number>_<index number>.py

with any additional files/sub-folders as needed in a folder named

Task2_4_<your name>_<center number>_<index number>.

[9]

Run the web application and save the output of the program as

Task2_4_<your name>_<center number>_<index number>.html

[1]

34 [HCI/PRELIM/9569/2021/P2/Q3]

Name your Jupyter Notebook as

TASK3_<your name>_<centre number>_<index number>.ipynb

The task is to implement a priority queue using Object-Oriented Programming.

A priority queue is an extension of queue with the following properties.

- Every element has a priority associated with it. Smaller integer value has a higher priority.
- An element with high priority leaves the queue before an element with low priority.
- If two elements have the same priority, they are served according to their order in the queue, i.e. the earlier element will be served before the later element (FIFO).

For example, the emergency room in a hospital assigns patients with priority numbers. The patient with the highest priority is treated first, regardless of the order of arrival.

An example of operations on a priority queue is shown below:

Initial state of priority queue:

Data	Ben	May	Anne	Jim
Priority	2	1	3	1

↑
↑
front
rear

Remove from priority queue: May is removed

Data	Ben	Anne	Jim
Priority	2	3	1

↑
↑
front
rear

Remove from priority queue: Jim is removed

Data	Ben	Anne
Priority	2	3

↑
↑
front
rear

Insert 'Ken' with priority 2

Data	Ben	Anne	Ken
Priority	2	3	2

↑
↑
front
rear

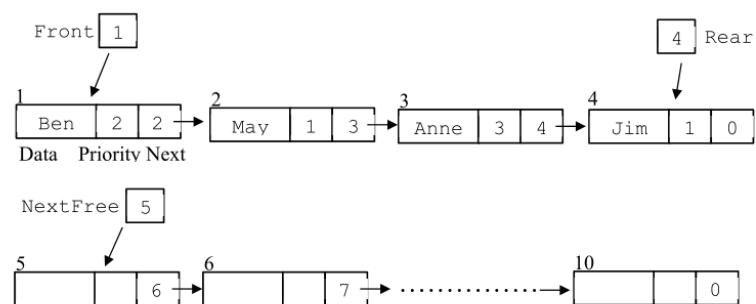
A **priority queue** abstract data type (ADT) is to be implemented as a linked list using object- oriented programming. Two classes Node and PQueue have been identified.

Class: Node		
Identifier	Data Type	Description
Data	STRING	The node data
Priority	INTEGER	Indicates priority of node. Smaller value has higher priority
Next	INTEGER	Pointer to next node in queue.

Class: PQueue		
Identifier	Data Type	Description
ThisPQueue	ARRAY[1:10] Of Node	The priority queue data.
Front	INTEGER	Index for front node of queue.
Rear	INTEGER	Index for rear node of queue.
NextFree	INTEGER	Index for the next unused node.
2*Initialise()	2*PROCEDURE	- Set pointers to indicate all nodes are unused and linked. - Initialise values for Front, Rear and NextFree.
2*PQInsert(NewItem, Priority)	2*PROCEDURE	- Assign NewItem and Priority passed as parameters to a node. - Insert the node to the rear of the priority queue.
2*PQDelete()	2*FUNCTION	- Remove a node of highest priority from the priority queue. - Return the Data attribute of the node that is removed.
DisplayPQueue()	PROCEDURE	Display the values of Front, Rear, NextFree and the content of the queue.

The diagram shows the linked list with:

- the data items 'Ben', 'May', 'Anne' and 'Jim' (inserted in that order) in the priority queue
- the unused nodes linked together



For each of the sub-tasks, add a comment statement at the beginning of the code using the hash symbol '#', to indicate the sub-task the program code belongs to, for example:

2*In[1] :	# Task 3.1 Program Code
2*In[2] :	# Task 3.2 Program Code
2*In[3] :	# Task 3.3 Program Code
Output :	

Task 3.1

Write program code for the classes `Node` and `PQueue`, including the `Initialise`, `PQInsert`, `PQDelete` and `DisplayPQueue` methods. The code should follow the specification given. [17]

Task 3.2

The program is to be tested.

Write a main program to:

- create a `PQueue` object
- read from file `PATIENTS.txt` all the data items with its priorities into the priority queue by calling `PQInsert` method.
- output the priority queue by calling `DisplayPQueue` method. [2]

Task 3.3

Write additional code in your main program to do the following in order by calling the appropriate methods from `PQueue` class.

No.	Operation	Data	Priority
1	Remove patient	-	-
2	Remove patient	-	-
3	Add patient Carol	Carol	4
4	Remove patient	-	-
5	Remove patient	-	-
6	Display priority queue	-	-

Save your Jupiter Notebook for Task 3.

[3]

35 [HCI/PRELIM/9569/2021/P2/Q4]

Name your Jupyter Notebook as

TASK4_<your name>_<centre number>_<index number>.ipynb

The task is to write program code for a Tic-Tac-Toe-Tomek game for two players.

Tic-Tac-Toe-Tomek is a game played on a 4 x 4 square board. The board starts empty, except that a single 'T' symbol may appear in one of the 16 squares. There are two players: X and O. They take turns to make moves, with X starting. In each move a player puts her symbol in one of the empty squares. Player X's symbol is 'X', and player O's symbol is 'O'.

After a player's move, if there is a row, column or a diagonal containing 4 of that player's symbols, or containing 3 of her symbols and the 'T' symbol, she wins and the game ends. Otherwise the game continues with the other player's move. If all of the fields are filled with symbols and nobody won, the game ends in a draw.

Given the 4 x 4 board description containing 'X', 'O', 'T' and '.' characters (where '.' represents an empty square). The following examples show the various winning positions.

X X X T	X O X O	O O X X	X X X O	O X X X
. . . .	X X O O	O X X X	. . O .	X O . .
O O . .	O X T X	O X . T	. O O .
. . . .	X X O O	O . . O	T T . O
X won	Draw	O won	O won	O won

For each of the sub-tasks, add a comment statement at the beginning of the code using the hash symbol '#', to indicate the sub-task the program code belongs to, for example:

2*In[1]:	# Task 4.1 Program Code
2*In[2]:	# Task 4.2 Program Code
2*In[3]:	# Task 4.3 Program Code
In[4]:	# Task 4.4 Program Code
In[5]:	# Task 4.5 Program Code

Output:

Task 4.1

Write program code to:

- initialize the data structure to represent the 4 x 4 square board, using the identifier `board`
- generate a pair of random numbers between 1 and 4
- place 'T' at that random position on the board

[3]

Task 4.2

Write a function `displayBoard` that will display the game board clearly to the players. You should use the `board` as a parameter in `displayBoard`.

[2]

Task 4.3

Write a function `getPlayerMove` to get players to make their move (by marking 'X' or 'O') on the board. You should include validation on player's input and check that the space is not already occupied. Use `board` as a parameter. You may include any other suitable parameters.

[4]

Task 4.4

Write a function `checkWin` that checks all the conditions for winning a game and returns `True` if a player has won the game, otherwise returns `False`. Use `board` as a parameter. You may include any other suitable parameters.

[5]

Task 4.5

Write a `main` function that makes use of the identifiers and functions from Task 4.1 to Task 4.4 and allows two players, X and O, to play a game of Tic-Tac-Toe-Tomek.

The `main` function should include the following:

- display the initial game board with the single 'T' displayed in it
- start with player X to make the first move
- ensure players X and O take turns to make their move

- display the game board after every move made by a player
- check for winner
- display message on which player has won the game or whether the game ends in a draw. [7]

Run your `main` function and produce outputs of **three** games where player X wins one game, player O wins another game, and a drawn game.

Copy and paste all outputs in a text file as

TASK4_5_<your name>_<centre number>_<index number>.txt [3]

Save your Jupyter Notebook for Task 4.

36 [JPJC/PRELIM/9569/2021/P2/Q1]

Your program code and output for Task 1 should be saved in a single .ipynb file.

Name your Jupyter Notebook as TASK1_<your name>_<class>_<index number>.ipynb

The file `marathon.CSV` contains the full list of athletes who took part in the 42.195km marathon race. The first line in the file is the heading for the records. Each subsequent line is a record of a runner in the form:

`<name of athlete>,<country code>,<timing in h:mm:ss>`

For example, `Abdi ABDIRAHMAN,USA,2:18:27`

Several athletes did not complete the race and their timing is recorded as 'DNF' to indicate 'did not finish'.

For example: `Alemu BEKELE,BRN,DNF`

Task 1.1

Write program code to find out the number of athletes who did not finish the race and output the following three statements:

Number of DNF: `x`

Total number of athletes: `y`

Percentage of athletes who finished race: `z`

`x` is the number of athletes who did not finish the race,

`y` is the total number of athletes who participated in the marathon race,

and `z` is the percentage (rounded to 1 decimal place) of athletes who finished the race.

[7]

Task 1.2

Write a function `insertionSort` that takes an unsorted list as a parameter, sorts the list using the insertion sort algorithm, and returns the sorted list.

[6]

Task 1.3

By making use of the `insertionSort` function from Task 1.2, or otherwise, find out the top 20 athletes and list them in order of rank under the heading (Rank, Country, Name, Timing).

[5]

Sample Output:

Rank	Country	Name	Timing
1	ABC	Harry TAN	2:08:38
2	XYZ	Andy LEE	2:09:58
3

37 [JPJC/PRELIM/9569/2021/P2/Q2]

Name your Python file as

TASK2_<your name>_<class>_<index number>.py

You will build a simplified, one-player version of the classic board game Battleship. Refer to Task2_Client_SampleOutput.JPG for the sample output.

In this version of the game:

- The **server program** initialises a grid measuring 4 metres by 5 metres.
- The grid is to be represented on the screen by a rectangular grid.
- Each square metre of the grid is represented by an x-coordinate and a y-coordinate.
- The top left square metre of the grid display has $x = 0$ and $y = 0$.
- Use "0" to represent an unoccupied space.
- There will be a single ship hidden in a random location.
- The ship only occupies one square metre of the grid.
- The **client program** allows the player to input guesses to sink the ship. After each guess, "x" represents the incorrect position guessed while "s" represents the sunken ship.
- The ship will not appear if it has not been sunk.
- The game is terminated by the server program when **three** guesses are used or the player has guessed the correct position.

Task 2.1

Write the code for the following functions and procedures for the **server program**.

[8]

Subroutine Header	Description
InitialiseGrid(): ARRAY[0:3, 0:4] OF CHAR	Initialises a 4 by 5 two-dimensional array and returns the array. Use "
DisplayGrid(arr: ARRAY[0:3, 0:4] OF CHAR)	Sends the encoded grid to the client.
ValidateRow(row: INTEGER): BOOLEAN	Returns True if the row is valid and False otherwise.
ValidateCol(col: INTEGER): BOOLEAN	Returns True if the column is valid and False otherwise.
CheckResult(row,col:INTEGER): BOOLEAN	Checks if the guess is correct. If the guess is correct, uses "s" to rep

Task 2.2

The following client program is given in Battleship_Client.py.

```
import socket

client_socket = socket.socket()

address = input('Enter IPv4 address of server: ')
port = int(input('Enter port number of server: '))
client_socket.connect((address, port))

while True:
    data = client_socket.recv(1024)
    if b"Enter" in data:
        choice = input(data.decode())
        client_socket.sendall(choice.encode())
        print()
    else:
        print(data.decode())
        if b"GAME OVER" in data or b"YOU WON" in data:
            break

client_socket.close()
```

Write the corresponding server program that:

[16]

- Instantiates the server socket.
- Binds the socket to localhost and port number 6789.
- Listens for incoming request, accepts incoming request and establishes connection with the client.
- Generates a random position for the hidden ship.
- Sends a "Welcome to Battleship!" message to the client.
- Uses the subroutines coded in Task 2.1 to play the game.
- Sends "YOU WON!" to the client for the correct guess and ends the game.
- Sends "GAME OVER. . ." to the client if three guesses are used.
- Closes the sockets.

38 [JPJC/PRELIM/9569/2021/P2/Q3]

Your program code and output for Task 3 should be saved in a single .ipynb file.

Name your Jupyter Notebook as TASK3_<your name>_<class>_<index number>.ipynb

JP Fitness Club is a gym that keeps details of its members. You are tasked to help the club manage the members' details and store them in a SQL database.

There are two types of membership: normal and annual. Each member has a unique membership number, first name, surname, contact number and last visit date recorded.

A normal member deposits a selected amount into their account. Each time the member visits the gym, the entrance fee is deducted from the amount held in his account. The member may top up the account any time.

An annual member pays a fixed fee per year, starting from the date of registration. He may then visit the gym any number of times for the whole year without paying the entrance fee.

Three classes have been identified: Member, NormalMember, AnnualMember.

The class Member has these attributes and methods defined on it.

Attribute	Data type	Description
memberID	String	8 digit membership number. First four digits represent the year of joining gym and last 4 digits represent the member's unique ID.
first_name	String	First name of member, at most 15 characters.
surname	String	Surname of member, at most 15 characters.
contact_number	String	8 digit contact number.
last_visit	String	The date when member last visited the gym, in the format YYYY-MM-DD.
memberType	String	Indicates type of membership, either "normal" or "annual".
Method	Return type	Description
showMember()	None	Outputs member's membership number, first name, surname, contact number, last visit date and membership type.
isActive()	Boolean	Indicates whether a member is active or not. Returns True if the last visit date is within the current year, False otherwise.

The class NormalMember inherits from Member and has these additional attributes and methods defined on it.

Attribute	Data type	Description
stored_value	Float	Amount stored in member's account. Display in 2 decimal places. Initialise to 0.00.
Method	Return type	Description
showMember()	None	Output memberType in addition to member's membership number, first name, surname, contact number, last visit date and membership type.

The class AnnualMember also inherits from Member and has these additional attributes and methods defined on it.

Attribute	Data type	Description
annual_fee	Integer	Annual fee paid by member. Initialises to \$500
date_register	String	Date when member joins annual membership, in format YYYY-MM-DD. Ini
Method	Return type	Description
showMember()	None	Outputs memberType in addition to member's membership number, first name, surnam

Task 3.1

Write program code using object-oriented programming for the classes `Member`, `NormalMember`, and `AnnualMember`.

Include all the identifiers stated and other appropriate methods to access and modify the attributes. [15]

Task 3.2

The text file, `members.TXT`, contains data items for a number of members. Each data item is separated by a comma, with each member's data on a new line as follows:

- membership number
- first name
- surname
- contact number
- member type

Write program code to read in the information from the text file, `member.txt`, creating an instance of the appropriate class for each member, storing each instance in the same list.

Run `showMember` method to display each of the members' details. [5]

Task 3.3

The members' details are to be stored in a SQL database.

The file `JPgym.SQL` contains the SQL code to create database `JPgym.db` with the single table, `Member`. The table will have the following fields:

- `MemberID` – primary key, text

- FirstName – first name of member, text
- Surname – surname of member, text
- ContactNo – contact number of member, text
- LastVisit – date that member last visited gym, text
- MemberType – indicates 'normal' or 'annual' membership, text

Copy and paste this SQL code into your Python program to create the database and table.

Also, write program code in Python to insert all the information from the file into the JPgym.db database.

Run your program and check that all information has been inserted using SQLite database software. [6]

Task 3.4

Write a SQL query code in Python to display all members with “normal” membership in ascending order of FirstName.

Display only the following fields from the query: FirstName, Surname, ContactNo. [3]

39 [JPJC/PRELIM/9569/2021/P2/Q4]

JP Mobile sells mobile phones and manages its inventory using a NoSQL database. Information about the mobile phones is stored in the JSON file `items.JSON`.

The following fields are recorded:

- brand of mobile phone,
- model,
- colour(s) available,
- price in dollar,
- quantity in stock.

Task 4.1

Write program code to import the information from the JSON file into a MongoDB database. Save the information under the `phone` collection in the `jp_mobile` database. Ensure that the collection only stores the information from the JSON file.

Save your program code as `TASK4_<your name>_<class>_<index number>.py`

[4]

Task 4.2

The shop decides to include one or more free gifts for new batches of mobile phones it sells.

Write program code for a user to insert information of a mobile phone by getting user input of the following: brand, model, colour, price, quantity, free gift(s).

Your code should allow user to input one or more free gifts.

If a phone's brand, model and colour already exists, add the new quantity to the existing quantity in the database, and replace the existing price with the new price.

Run your program and insert the following 2 documents:

No.	Brand	Model	Colour	Price	Quantity	Free gift(s)
1	orange	22	black	900	11	power bank
2	solo	A33	red	1300	7	power bank, earbuds

Add your program code to `TASK4_<your name>_<class>_<index number>.py`

[7]

Task 4.3

Write a function `display_all` that will display all the information in the `phone` collection under these fields: `brand, model, colour, price, quantity, free_gift(s)`.

If no free gift comes with the phone, print a `None` statement.

Include a final statement that shows the total number of documents in the collection.

Run the `display_all` function to show your output.

Add your program code to `TASK4_<your name>_<class>_<index number>.py`

[6]

Task 4.4

The shop uses a web browser to display the database content. The manager wants to filter the mobile phones by `brand` and display the results in a web browser.

Write additional Python code and the necessary files to create a web application that

- receives a `brand` string from a HTML form, then
- creates and returns a HTML document that enables the web browser to display an ordered list of mobile phones sorted by `price`.

For each document, the web page should include the:

- `brand` as the heading
- `model`,
- `colour`,
- `price`

Save your program as

`TASK4_4_<your name>_<class>_<index number>.py`

with any additional files / sub-folders as needed in a folder named

`TASK4_4_<your name>_<class>_<index number>`

Run the web application and input `brand` as `'solo'` in the webpage.

Save the output of the program as

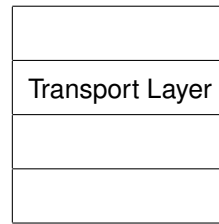
`TASK4_4_<your name>_<class>_<index number>.html`

[12]

40 [NYJC/PRELIM/9569/2021/P1/Q1]

(a) The TCP/IP networking model comprises 4 layers.

(i) Copy and complete the below diagram for the TCP/IP stack.



[3]

(ii) State the purpose of the Transport Layer in the TCP/IP network model.

[2]

(iii) State the reserved port range used in the Transport Layer.

[1]

(b) A certain port number in binary form is 00000100 00011000.

(i) Express this value in denary (decimal) form.

[3]

(ii) Explain how it may be checked if this binary port value is within the reserved port range, without converting it to decimal.

[2]

(c) (i) State which layer of the TCP/IP model the Domain Name System (DNS) protocol belongs to.

[1]

(ii) Describe the purpose of the DNS protocol.

[2]

41 [NYJC/PRELIM/9569/2021/P1/Q2]

(a) Validation and verification are used in data entry.

(i) State the purpose of validation. [1]

(ii) State the purpose of verification. [1]

A network program using TCP needs to check if a port number is within an acceptable range of values.

(b) Is this fulfilled by validation or verification? Explain your answer. [2]

(c) Provide three sets of suitable test values for the above check. [3]

42 [NYJC/PRELIM/9569/2021/P1/Q3]

A printing shop needs to set up a print queue system to serve its customers. This print queue will manage print tasks, by sending them one at a time to available printers.

For all print tasks, the data that will be stored include:

User

Printer address

Job name

Status

The print queue itself stores the following data:

Number of jobs

When a print task is added to the queue:

- The task is stored inside the queue, in FIFO order
- The jobs count is incremented by one

When a print task is sent to a printer:

- The print task is removed from the queue, in FIFO order
- The jobs count is decremented by one

(a) (i) Draw a class diagram that shows the following for the situation described above.

- The classes
- properties
- appropriate methods

[9]

(ii) Explain the meaning of the terms:

1. inheritance

[2]

2. polymorphism

[2]

The printing shop wishes to implement a circular queue to limit the maximum number of pending jobs and improve the performance of their system.

(b) (i) State two differences between a linear queue and a circular queue.

[2]

(ii) Suggest whether inheritance or polymorphism is a more suitable principle to apply in the implementation of both linear queue and circular queue in the same program. Explain your answer.

[4]

- (c) Using a suitable diagram, pseudocode, or other method, show how an item would be added to a circular queue implemented with a static array. [4]

43 [NYJC/PRELIM/9569/2021/P1/Q4]

An algorithm for sorting an array of elements is shown.

```

01 FOR i = 1 to Array.LENGTH - 1
02     FOR j = 1 to Array.LENGTH - 1
03         IF Array[j] > Array[j+1]
04             THEN
05                 t = Array[j]
06                 Array[j] = Array[j+1]
07                 Array[j+1] = t
08             ENDIF
09     ENDFOR
10 ENDFOR

```

- (a) (i) State the algorithm represented. [1]
- (ii) State the time complexity of this algorithm. [1]
- (iii) Copy and complete the trace table below with the value of Array at the end of each iteration of i in the algorithm. [5]

i	Array
Initial	[2, 3, 4, 5, 6, 1]
1	
2	
3	
4	
5	
6	

- (b) Describe **two** improvements that could be made to the above algorithm to improve its efficiency. [4]
- (c) Explain why insertion sort is usually used instead of bubble sort, although both have the same time efficiency. [2]

44 [NYJC/PRELIM/9569/2021/P1/Q5]

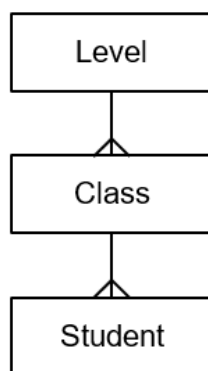
A school database has some data in the following table:

StudentID	Student Name	Class	Subjects
1	Wong Yong Ming	1917	H2MATH, H2PHY, H2CHEM, H2ECON
2	Vikram Singh	1911	H2MATH, H2CHEM, H2ECON, H1GEOG
3	Muhd Bashir bin Ramdan	1911	H2MATH, H2CHEM, H2ECON, H1ELIT

(a) (i) State and explain if the above table is in third normal form (3NF). [4]

(ii) Describe two advantages that normalised data has over redundant data. [2]

In an effort to improve the school database, the IT administrator came up with an ER diagram. Part of the full ER diagram is as shown.



Each of the three entities in the ER diagram has a name attribute.

(b) (i) Write table descriptions to implement the above ER diagram. [5]

(ii) Write an SQL query to retrieve only the student name and class name for all students in the level JC2. [3]

A fast-growing startup is writing code to provide a new service. The user needs have not yet been fully determined, and the data schema is likely to undergo further changes before being finalised.

(c) (i) Suggest if SQL or NoSQL is more suitable for the needs of this startup. Give **two** reasons to support your answer. [4]

(ii) Describe **two** challenges the startup will face in using NoSQL databases. [2]

(d) The startup is concerned that a hardware failure may wipe out critical data and leave them unable to continue operating.

Suggest what the startup should do to **ensure** that they are safe against data loss in such a scenario. [4]

45 [NYJC/PRELIM/9569/2021/P1/Q6]

Alice, a programmer, is implementing a DNS cache using a hash table.

(a) Explain the purpose of the hash function in a hash table.

[2]

The description for a particular hashing algorithm using rolling polynomials is as follows:

For each character in the data, do the following:

1 Let i represent the position of the character (1st char = 1, 2nd char = 2, ...)

2 Let $ascii$ represent the ASCII value of the character

3 Calculate the sum of $i \times (31^{ascii})$ for all characters

(b) Implement this algorithm in pseudocode.

You may assume that the function `Ord()` is available, which takes in a single character and returns the ASCII value of the character.

[4]

(c) Bob, another programmer, suggests that a Binary Search Tree would be a more appropriate data structure for the DNS cache.

(i) Describe one advantage of using a hash table for the DNS cache.

[2]

(ii) Describe one advantage of using a Binary Search Tree for the DNS cache.

[2]

(d) (i) State the algorithm used to retrieve the sorted contents of a cache from a Binary Search Tree.

[1]

(ii) Using any appropriate diagrams, pseudocode, or other appropriate method, show how this algorithm might be carried out.

[5]

(iii) Explain why the Binary Search Tree might need to be periodically recreated.

[3]

46 [NYJC/PRELIM/9569/2021/P2/Q1]

Name your Jupyter Notebook as

TASK1_<your name>_<centre number>_<index number>.ipynb

A text file, TIDES.TXT, contains the low and high tide information for a coastal location for each day of a month. Each line contains tab-delimited data that shows the date, the time, whether the tide is high or low and the tide height in metres.

Each line is in the format:

YYYY-MM-DD\tHH:mm\tTIDE\tHEIGHT\n

- The date is in the form YYYY-MM-DD, for example, 2019-08-03 is 3rd August, 2019
- The time is in the form HH:mm, for example, 13:47
- TIDE is either HIGH or LOW
- HEIGHT is a positive number shown to one decimal place
- \t represents the tab character
- \n represents the newline character

The text file is stored in ascending order of date and time.

For each of the sub-tasks, add a comment statement, at the beginning of the code using the hash symbol '#', to indicate the sub-task the program code belongs to, for example:

```
2*In[1]: # Task 1.1
        Program Code

Output:
```

Task 1.1

Write program code to:

- read the tide data from a text file
- find the highest high tide and print this value
- find the lowest low tide and print this value

Use TIDES.TXT to test your program code.

[7]

Save your Jupyter Notebook for Task 1.

Task 1.2

The tidal range is the difference between the heights of successive tides; from a high tide to the following low tide or from a low tide to the following high tide.

Amend your program code to:

- output the largest tidal range and the date on which the second tide occurs
- output the smallest tidal range and the date on which the second tide occurs

Use `TIDES.TXT` to test your program code.

[4]

47 [NYJC/PRELIM/9569/2021/P2/Q2]

Name your Jupyter Notebook as

TASK2_<your name>_<centre number>_<index number>.ipynb

The task is to implement a todo list using a linkedlist data structure. For each of the sub-tasks, add a comment statement, at the beginning of the code using the hash symbol '#', to indicate the sub-task the program code belongs to, for example:

2*In[1]:	# Task 2.1 Program Code
----------	----------------------------

Output:

Task 2.1

The class `ToDoList` represents a `LinkedList` and has the following attributes:

- `__head` – a pointer to the first node of the `LinkedList`; if empty, it has a value of `None`
- `__tail` – a pointer to the last node of the `LinkedList`; if empty, it has a value of `None`

`ToDoList` has the following methods defined on it:

- `add(item)` – wraps `item` in a `ToDoItem` instance, and adds it to the end of the `LinkedList`
- `remove(item)` – removes the first `ToDoItem` containing `item` from the `LinkedList`
- `list()` – returns a Python list containing each `item` in the `ToDoList`

The class `ToDoItem` represents a `Node` of the `LinkedList` and has the following attributes:

- `title` – a short description of the todo item
- `__next` – a pointer to the next node in the `LinkedList`; if this is the last node, it has a value of `None`

`ToDoItem` has the following methods defined on it:

- `link_to(todoitem)` – links this `ToDoItem` instance to `todoitem`, another instance of the `ToDoItem` class

Implement the above classes.

[13]

Task 2.2

Add the following items to a new `ToDoList`:

- “Buy milk”
- “Buy flour”
- “Buy eggs”
- “Bake cake”

Display the contents of the `ToDoList`.

[7]

Task 2.3

Remove the following items from the `ToDoList`:

- “Buy milk”
- “Buy eggs”

Display the contents of the `ToDoList`.

[3]

Save your Jupyter Notebook for Task 2.

48 [NYJC/PRELIM/9569/2021/P2/Q3]

Name your Jupyter Notebook as

TASK3_<your name>_<centre number>_<index number>.ipynb

The task is to write a function that takes a sequence of characters representing a colour, and translates the colour into a different number base.

8-bit colours are represented with three numbers, indicating the level of the colours red (R), green (G), and blue (B) respectively. Each number is an integer from 0 to 255. 255 represents the fully saturated colour, while 0 represents zero saturation (black).

In HTML, these colours may be represented using hex code as well. In hex code, the R, G, and B values are converted to hexadecimal. Hex codes begin with the symbol '#' followed by the three R, G, and B hexadecimal values.

For example, the hex code #0A0B0C represents a colour with RGB values 10, 11, and 12 respectively.

For each of the sub-tasks, add a comment statement, at the beginning of the code using the hash symbol '#', to indicate the sub-task the program code belongs to, for example:

```
2*In[1]: # Task 3.1
        Program Code

Output:
```

Task 3.1

Write a function called `task3_1(hex)` that:

- takes `hex`, a string representing a hex code, beginning with a '#' symbol followed by three valid hexadecimal values between 00 and FF
- returns and displays either:
 - a 3-integer tuple representing RGB values
 - or
 - the error message, "invalid data"

[5]

Test the function fully with suitable test data.

For example,

```
task3_1("#FFFFFF")
```

should return and display (255, 255, 255)

[3]

Task 3.2

Some image programs do not represent colours using 8-bit integers. Instead, they represent them as a normalised float value. In this representation, a value of 1.0 represents the fully saturated colour and a value of 0 represents zero saturation (black).

Write a second function `task3_2(rgb)` that:

- takes a 3-integer tuple `rgb` representing RGB values
- returns and displays either:
 - a 3-float tuple representing normalised RGB
 - or
 - the error message, "invalid data"

[5]

Test the function fully with suitable test data.

For example,

```
task3_2((128, 128, 128))
```

should return and display (0.50196, 0.50196, 0.50196)

```
task3_2((255, 255, 255))
```

should return and display (1.0, 1.0, 1.0)

[3]

Task 3.3

Image filters are functions that take in image data and change the RGB values of its colours according to an algorithm. The algorithm for converting an image to grayscale calculates the average of the RGB values and sets the R, G, and B values to this average.

Write a third function `task3_3(hex)` that:

- takes `hex`, a string representing a hex code
- returns and displays a 3-float tuple representing normalised RGB of the colour converted to grayscale [4]

Test the function fully with **two** suitable values.

For example,

```
task3_3("#FF8000")
```

should return and display (0.5, 0.5, 0.5)

[2]

Save your Jupyter Notebook for Task 3.

49 [NYJC/PRELIM/9569/2021/P2/Q4]

A bookstore uses a text file to store data about its inventory of books. The bookshop carries two kinds of books: printed books and virtual books. The bookshop wishes to transfer this information into a database.

The bookshop also wishes to create an online bookstore that allows users to add books to a shopping cart for purchase.

Task 4.1

Create an SQL file called `TASK4_1_<centre number>_<index number>.sql` to show the SQL code to create database `bookstore.db` with three tables: `Book`, `Printed`, and `Virtual`. The `Printed` and `Virtual` tables represent physical and virtual books respectively, and stores properties unique to each type of book.

The `Book` table will have the following fields:

- `BookID` – the primary key, an integer value
- `Title` – the title of the book
- `Price` – the price of the book, in cents
- `Type` – the type of book: "physical" or "virtual"

The `Printed` table will have the following additional field:

- `Weight` – the weight of the book

The `Virtual` table will have the following additional field:

- `DownloadLink` – the download link for the book

Save your SQL code as

`TASK4_1_<your name>_<centre number>_<index number>.sql`

[6]

Task 4.2

Python programming language and object-oriented programming will be used to implement the online bookstore and shopping cart on a web page.

The class `Book` will store the following data:

- `title` – stored as a string
- `price` – stored as an integer

The class `Cart` will store the following data:

- `items` – stored as a list of `Book` objects

The class `Cart` has a method defined on it:

- `total_price()` – returns an integer representing the total price of books in the cart

Save your program code as

`TASK4_2_<your name>_<centre number>_<index number>.py`

[6]

The `PrintedBook` class inherits from `Book`, and stores the following additional data:

- `weight` – stored as an integer

`VirtualBook` class inherits from `Book`, and stores the following additional data:

- `download_link` – stored as a string

Add your program code to

`TASK4_2_<your name>_<centre number>_<index number>.py` [3]

The text file, `bookstore.txt`, contains data items for books stocked by the bookstore. Each data item is separated by a comma, with each book's data on a new line as follows:

- book title
- price
- type
- weight
- download link

Write program code to read in the information from the text file, `bookstore.txt`, creating an instance of the appropriate class for each book (either `PrintedBook` or `VirtualBook`). [4]

Write program code to insert all information from the file into the `bookstore.db` database.

Run the program. Add your program code to

`TASK4_2_<your name>_<centre number>_<index number>.py`

[8]

Task 4.3

The data from the text file, `bookstore.txt`, is to be used to implement a shopping cart in a web browser.

Write a Python program and the necessary files to create a web application that:

- displays a list of books stocked by the bookstore
- enables the user to add books to a shopping cart using an ID
- displays the contents of the shopping cart
- shows the total price of items in the shopping cart

For each book displayed the web page should include the:

- book ID
- book title
- price

Save your program as

`TASK4_3_<your name>_<centre number>_<index number>.py`

with any additional files / sub-folders as needed in a folder named

`TASK4_3_<your name>_<centre number>_<index number>`

[7]

Run the web application and add the following books to the shopping cart:

- Title: "Northanger Abbey", Price: 13.99, Type: Physical, Weight: 178g
- Title: "War and Peace", Price: 17.49, Type: Physical, Weight: 432g
- Title: "Computer Programs", Price: 20.99, Type: Virtual, Link: <https://mybookstore.com/dJHtFy>
- Title: "Data Science", Price: 14.99, Type: Virtual, Link: <https://mybookstore.com/fJynJk>

Save the output of the program as

`TASK4_3_<your name>_<centre number>_<index number>.html`

[4]

50 [RVHS/PRELIM/9569/2021/P1/Q1]

Draw a reduced decision table based on the following conditions regarding how John should go to school. [5]

- If it is a Monday, John always takes his dad's car to school if he does not oversleep.
- If John oversleeps, he always takes Taxi to school.
- Otherwise, if it is a rainy day, John takes Taxi to school. If not, by MRT.

51 [RVHS/PRELIM/9569/2021/P1/Q2]

The recursive function below helps to check if a password string `pw` satisfies certain requirements. The meaning of the function parameters `pw`, `digits`, `upper_l`, `lower_l` and `length` are password string, minimum number of digits, minimum number of uppercase letters, minimum number of lowercase letters and minimum length of the password respectively.

```
def check_pw(pw, digits, upper_l, lower_l, length):
    if len(pw) == 0:
        return digits < 1 and upper_l < 1 and lower_l < 1 and length < 1
    else:
        char = pw[0]
        if char.isdigit():
            return check_pw(pw[1:], digits-1, upper_l, lower_l, length-1)
        elif char.isalpha():
            if char.isupper():
                return check_pw(pw[1:], digits, upper_l-1, lower_l, length-1)
            else:
                return check_pw(pw[1:], digits, upper_l, lower_l-1, length-1)
        else:
            return False
```

- (a) State the values of all arguments in each recursive function call when the following code is executed. Then, state the value that the function returns.

```
>>> check_pw("SP500", 3, 1, 1, 5)
```

[3]

The function in **2a** is rewritten in such a way that string slicing on the password string `pw` is removed. A new function parameter `i` is added to help the recursive function to keep track of the position in `pw` in which it is currently checking.

```

def check_pw(pw, i, digits, upper_l, lower_l, length):
    if A:
        return digits < 1 and upper_l < 1 and lower_l < 1 and length < 1
    else:
        B
        if char.isdigit():
            return check_pw(pw, C, digits-1, upper_l, lower_l, length-1)
        elif char.isalpha():
            if char.isupper():
                return check_pw(pw, C, digits, upper_l-1, lower_l, length-1)
            else:
                return check_pw(pw, C, digits, upper_l, lower_l-1, length-1)
        else:
            return False

```

(b) State the code in **A**, **B** and **C**.

[3]

(c) Describe clearly or write in Python the modification needed for the function `check_pw()` to also display a suggestion of a new password if the password requirements are not met. For example, if `pw` does not have enough digits, it will append digits to `pw` so that it can satisfy the requirement.

For example,

```
>>> check_pw("WoBeiShiGeDa", 0, 2, 6, 7, 15)
```

```
Suggested password: WoBeiShiGeDa33Uxxx
```

```
False
```

"33Uxxx" is added to "WoBeiShiGeDa" so that the password passes the requirements.

[4]

52 [RVHS/PRELIM/9569/2021/P1/Q3]

The following recursive procedure is created to encode a character `char` based on a `shift` value. For example, if letter "a" is shifted by 3, it will become letter "d"; and if shifted by -3, it will become letter "x".

The `ORD` and `CHR` function will help to convert the character to its ASCII value and vice versa.

```

01 PROCEDURE ENCODE_CHAR(char:  STRING, shift:  INT):
02     IF shift == 0:
03         RETURN char
04     ELSE IF shift > 0:
05         DECLARE new_char:  STRING
06         new_char = CHR((ORD(char) + 1) % 26)
07         shift -= 1
08         RETURN ENCODE_CHAR(new_char, shift)
09     ELSE:
10         Shift += 26
11         RETURN ENCODE_CHAR(char, shift)
12     END IF
13 END PROCEDURE

```

(a) Identify one error from the above code, state the type of the error, including its definition and explain how the errors can be fixed. [2]

(b) Assume the above error has been fixed. Copy the following trace table to your answer booklet. State the line number each time one of the return statements is called and complete it based on the following function call.

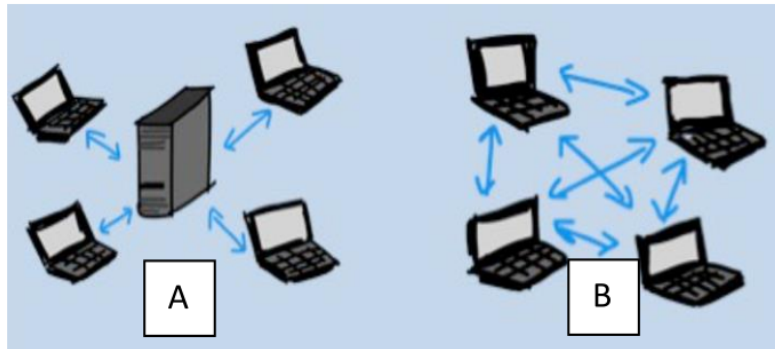
`ENCODE_CHAR("y", -24)`

[3]

Line No.	char	shift	new_char

53 [RVHS/PRELIM/9569/2021/P1/Q4]

Answer all questions.



- (a) State the network architecture model of A and B as shown in the diagram below. [1]
- (b) State an advantage of model A over B in the diagram above. [1]
- (c) State and explain if each of the following statements are correct. [6]
- “One of the functionalities of the DNS is that different users can simultaneously receive different IP translations for the same domain name.”
 - “The 4 top layers of TCP/IP model are application, internet, data link and physical.”
 - “The internet layer is not responsible for reliable transmission. It makes no guarantees about the proper arrival of packets.”
 - “2C:54:91:G8:F9:E3 is a valid MAC address.”
 - “2001:0db8:0001:0ab9:C0A8:0102 is a valid IPv6 address.”
 - “The internet and the World Wide Web are the same thing.”
- (d) State the purpose of HTTP and explain how the protocol works. [3]
- (e) Explain packet switching. [1]
- (f) State an ethical issue related to artificial intelligence. [1]
- (g) State the purpose of defining the code of conduct for computer use. [2]
- (h) State a difference between data validation and data verification. [1]

54 [RVHS/PRELIM/9569/2021/P1/Q5]

The implementation of a Binary Search Tree (BST) using three 1D arrays is shown below.

Each unused node that are not in the logical BST is initially connected in a singly linked list manner using the `leftPtr` array. The first position of this linked list is indicated by the variable `nextFree`.

When a piece of data is inserted into the BST, a node will be disconnected from the linked list and added to the logical BST. The root of this logical BST is indicated by the variable `root`. The logical structure of the BST is managed by `leftPtr` and `rightPtr` which are the positions of the left and right child of the node respectively.

Below is an illustration for such BST with a 0-based index array.

root	7
nextFree	0

Array Index	0	1	2	3	4	5	6	7
Data	-	7	10	6	1	4	2	9
leftPtr	-1	-1	-1	5	-1	6	4	3
rightPtr	-1	-1	-1	1	-1	-1	-1	2

(a) Draw the logical BST at this point of time. [1]

(b) State the post order traversal of the BST [1]

(c) State the values of `root`, `nextFree` and the values in the arrays `data`, `leftPtr` and `rightPtr` after the **each** of the following BST operations are executed sequentially. [3]

- Add 8
- Recursive Delete 6

(d) State an advantage of BST over Hash table. [1]

(e) Explain what can make a Hash Table Search inefficient besides a bad hash function and how to overcome it. [2]

(f) State 2 characteristics of a good hash function. [2]

55 [RVHS/PRELIM/9569/2021/P1/Q6]

Study the following sorting code carefully.

```
def sort(lst):
    if len(lst) <= 1:
        return lst
    else:
        pivot = lst[0]
        smaller = []
        larger = []
        for i in range(1, len(lst)):
            if lst[i] < pivot:
                smaller.append(lst[i])
            else:
                larger.append(lst[i])
        return sort(smaller) + [pivot,] + sort(larger)
```

- (a) State the name of the above sorting algorithm. [1]
- (b) Explain why the above sorting algorithm is inefficient when it is used on a nearly sorted array. [1]
- (c) Explain how you can modify the code to improve the efficiency. [2]

Bubble sort and insertion sort are both used to sort a nearly sorted integer array of size 1000 and there are only 5 integers in the array that are not in the correct position.

- (a) State and explain why insertion sort generally perform better than bubble sort for nearly sorted array. [2]
- (b) Draw a flow chart for the algorithm described below. [4]

Given an integer k and an array `arr[]` representing the destination floors for n people waiting currently at the ground floor and k is the capacity of the elevator. It takes 1 unit time for the elevator to reach any consecutive floor from the current floor. The algorithm finds the minimal time taken to get all the people to their destination floor and then return to the ground floor.

```
def minTime(n, k, arr) :  
    # Sort in descending order  
    arr.sort(reverse = True)  
    minTime = 0  
    # Iterate through the groups  
    for i in range(0, n, k) :  
        # Update the time taken for  
        # each group  
        minTime += (2 * arr[i])  
    # Return the total time taken  
    return minTime
```

56 [RVHS/PRELIM/9569/2021/P1/Q7]

A relational database is created to store data about contractors engaging workers to perform renovation jobs.

The database designers are told that:

- each contractor can recruit different workers to perform various jobs.
- each worker can have skills to perform different jobs.
- each job can have different levels of skills "A", "B" or "C" and their hourly rate is calculated based on their skill level for the job.

A first attempt is represented by the following table:

Contractor Name	Contractor Contact	Worker Name	Job	Skill Level	Hourly Rate	Date	Starting Time	Ending Time
Su Ming De	94190731	Tan Yong Quan	Carpentry	A	40	20200801	0900	1300
		Goh Yi Xi	Carpentry	B	35	20200801	0800	1700
		Goh Yi Xi	Electrical	A	42	20200801	0900	1600
Qin Kai Hui	95967027	Chong Jun Jie	Ceiling	B	23	20200801	1000	1200
		Tung De Ming	Waterproofing	A	30	20200802	0900	1300
		Tan Yong Quan	Carpentry	A	40	20200802	1200	1800
		Goh Siew Ming	Waterproofing	A	30	20200802	0900	1300
Shen Rui Min	98022248	Tung De Ming	Waterproofing	A	30	20200803	0900	1300
		Tan Yong Quan	Carpentry	A	40	20200803	1000	1500
		Goh Siew Ming	Ceiling	B	23	20200803	0900	1300

(a) Explain why this table is not in first normalized form.

[2]

The following is an attempt to reduce data redundancy:

Contractor	Contractor ID	Contractor Name	Contractor Contact
	1	Su Ming De	94190731
	2	Qin Kai Hui	95967027
	3	Shen Rui Min	98022248

Worker	Worker ID	Job	Worker Name	Skill Level	Hourly Rate
	1	Ceiling	Chong Jun Jie	B	23
	2	Ceiling	Goh Siew Ming	B	23
	2	Waterproofing	Goh Siew Ming	A	30
	3	Carpentry	Goh Yi Xi	B	35
	3	Electrical	Goh Yi Xi	A	42
	4	Carpentry	Tan Yong Quan	A	40
	5	Waterproofing	Tung De Ming	A	30

Contract Record	Contractor ID	Worker ID	Job	Date	Starting Time	Ending Time
	1	4	Carpentry	20200801	0900	1300
	1	3	Carpentry	20200801	0800	1200
	1	3	Electrical	20200801	1300	1600
	1	1	Ceiling	20200801	1000	1200
	2	5	Waterproofing	20200802	0900	1300
	2	4	Carpentry	20200802	1200	1800
	2	2	Waterproofing	20200802	0900	1300
	3	5	Waterproofing	20200803	0900	1300
	3	4	Carpentry	20200803	1000	1500
	3	2	Ceiling	20200803	0900	1300

- (b) Explain what a composite key is. [1]
- (c) State a suitable primary key for Worker table and explain why the table is not in second normal form. [3]
- (d) A table description can be expressed as:
- TableName (Attribute1, Attribute2*, Attribute3, ...)
- The primary keys are indicated using a solid underline and foreign keys are indicated by using a dashed underline/asterisk. Write table descriptions for the required tables in the database so that they are in third normal form (3NF). [6]
- (e) Create an entity-relationship (ER) diagram showing the degree of all relations. [3]
- (f) Using the above example, elaborate why a relational database model has advantage in maintaining data integrity over a flat file system. [3]
- (g) The homeowner would like to know a schedule of the renovation jobs performed to their house. They are **NOT** interested in knowing the exact worker's name. Write an SQL query to output the **contractor's name**, **worker's job**, **worker's skill level** and **date** based on the contractor's name "Su Ming De". The output is to be in the ascending order based on the date of job performed. [5]

57 [RVHS/PRELIM/9569/2021/P1/Q8]

You are also to design an Object-Oriented solution for the above-mentioned project. Both contractor and workers are to create a `User` account on the platform, with details such as `user_id`, `password` and `gender`.

The contractors will have to register their company details such as `company_name` and `address`, while the workers need to register their `bank_account_number`.

(a) Draw a class diagram, with base class **User**, showing:

- appropriate sub-classes,
- inheritance,
- the properties required,
- appropriate methods, including but not limited to the **constructor** methods, and at least **one** pair of **'get'** and **'set'** methods for each class,
- circle the polymorphed methods. [6]

(b) Using the above example, state the definition of inheritance and explain its purpose/advantage in object-oriented programming. [3]

The platform hopes to expand its function to allow register of homeowner accounts. The homeowners can view which are the workers came to their home address for renovation work on the date/time specified by the contractors.

(c) State how this would affect the class, properties and methods in the current example. [3]

(d) State how this would affect the tables, attributes and relationships of the relational database stated in **7(d)** and **(e)**. [3]

(e) Explain how NoSQL addresses shortcomings of relational databases. [4]

Some homeowners request to have access to the hourly rate and personal contact of renovation workers.

(f) From the perspective of the company, explain to the homeowners how such a feature is against the data protection obligations stated in the Personal Data Protection Act (PDPA). [2]

58 [RVHS/PRELIM/9569/2021/P2/Q1]

Task 1.1

Write a function `task1_1(name_A, name_B)` where `name_A` and `name_B` are strings which consists of only alphabet letters and spaces. The function will return `True` if

- the alphabet letters combination used in string `name_A` and `name_B` are the same and
- the spaces in string `name_A` and `name_B` are at the same locations.

[7]

For example,

```
>>> match_names("Abcde", "Deabc")
True
>>> match_names("Abcde Fgh I", "Ihgfe Dcb A")
True
>>> match_names("Abcd Efgh I", "Ihgfe Dcb A")
False
>>> match_names("Abcde Fzh I", "Ihgfe Dcb A")
False
```

Test your program with the following test data:

```
print(task1_1("Abcde", "Deabc"))
print(task1_1("Abcde Fgh I", "Ihgfe Dcb A"))
print(task1_1("Abcd Efgh I", "Ihgfe Dcb A"))
print(task1_1("Abcde Fzh I", "Ihgfe Dcb A"))
```

Task 1.2

Write the function `task1_2()` to read the names in file `"namelist_A.txt"` and find a matching name in `"namelist_B.txt"` that satisfied the conditions stated in **Task 1.1**. If a matching name cannot be found in `"namelist_B.txt"`, it will just display `"*****No match*****"`. Your results should be displayed in the following manners.

The time-complexity of your program code matters.

[13]

46 task1_2()	
From namelist_A.txt	From namelist_B.txt
Aaidg Znhel Lladunne	Ldnhe Lunaz Alnigde
Aanzo Tah Yjeynun	Njuyn Anz Oyaheta
Aeah Oih Cpl Onsk	Haoh Ilp Ank Esco
Agt Knn Ayso	*****No match*****
Ahm Gi Lpiin	Ani Pm Igilh
Aih Icw Nganh	Chn Iia Ghwan
Alijg In	Niigj Al
Anhaay Htn Ala Knwn	Aktyha Ann Awa Nlhn
Arrae Vaidtnrasnatem	Dnarv Etmsatairaaern
Attt Nne Kooj Cngh	Nojt Nen Takg Toch
Aycd Ssaorhza	Rsaz Asdohcya
Beo Lywe Oone	Yoo Wnee Bloe
Bniimg Eos Mdne Jnl	Msnemn Nje Doii Blg
Bno Atige Ngng	Agg Ngnit Oebn
Chnyinlo Iag Apne Aact	Cnpiaagl Cei Ahnt Yaon
Cianra Akl Ahtn Nem	Hatmaa Nnc Kare Lin
Cnhh Ones Ewic	Nhwi Nhsc Oece
Co Eehw Ehe	He Eeoc Ehw
Dai Imra Flv Fonlt	Dar Lgea Ilm Ynite

59 [RVHS/PRELIM/9569/2021/P2/Q2]**Task 2.1**

Complete the doubly linked list class `Doubly_LL` by implementing both the `insert` and `delete` class functions. [10]

- The class function `insert(node)` takes a `Node` instance `node` as input and inserts it at the tail of the linked list. Take note that the attributes `prev` and `next` of `Node` instance `node` are both `None` before the insertion.
- The class function `delete(node)` takes a `Node` instance `node` which exists in the linked list as input. The function removes/detaches `node` from the linked list and returns it. The `node` returned has both its attributes `prev` and `next` set to `None`, but data remains unchanged.

Task 2.2

The class `LRUQ` uses the doubly linked list class `Doubly_LL` to implement its least recently used queue (`lruq`). [8]

- The attribute `hashmap` is a dictionary object that takes the node's data as key and the node instance itself in `lruq` as value.
- The attribute `size` is the max number of nodes that `lruq` can have.
- The attribute `count` is the number of nodes that `lruq` currently have.

Complete the least recently used queue class `LRUQ` by implementing the `use` function. The class function `use(value)` takes an integer `value` as input.

- If `value` is in `lruq` (referenced by `hashmap`), it removes the node in the `lruq` and re-insert it to the end of the `lruq`.
- If `value` is not in `lruq` (not referenced by `hashmap`), it references `value` in `hashmap` and inserts a new `Node` instance with `value` as its data to the end of `lruq`. If `count > size`, it removes the least recently used node in `lruq` and de-references it in `hashmap`.

Hint: To de-reference a key in `hashmap`, you can call the following. `self.hashmap.pop(key, None)` where `self.hashmap` is a dictionary object.

The test function `test2_2()` is provided for you in `task2.ipynb`. The expected outcome of this test function is shown on the next page. Take note that the size of the least recently used queue is 6 in this test function.

Latest item used: 3

From least recently used to most recently used: Print from head:

3

Latest item used: 8

From least recently used to most recently used: Print from head: 3 8

Latest item used: 2

From least recently used to most recently used: Print from head:

3 8 2

Latest item used: 45

From least recently used to most recently used: Print from head:

3 8 2 45

Latest item used: 3

From least recently used to most recently used: Print from head:

8 2 45 3

Latest item used: 45

From least recently used to most recently used: Print from head:

8 2 3 45

Latest item used: 45

From least recently used to most recently used: Print from head:

8 2 3 45

Latest item used: 12

From least recently used to most recently used:

Print from head: 8 2 3 45 12

Latest item used: 31

From least recently used to most recently used: Print from head:

Generated with Python Script by BRW
8 2 3 45 12 31 //Queue is full at this point

Latest item used: 42

From least recently used to most recently used: Print from head:

[Turn Over

60 [RVHS/PRELIM/9569/2021/P2/Q3]

Task 3.1

Write program code to read the csv file "health_facilities.csv" and insert all information in the file as documents into a NoSQL MongoDB database called "Health" with one collection called "facilities". The "_id" of the documents in the database should start from 1, 2, 3 and 4 etc. The correct data type of each field is expected to be inserted into the database. [10]

Task 3.2

(a) Write a MongoDB Pymongo query to retrieve all public acute hospital documents with their corresponding number of beds more than 7200. [4]

(b) Write program code to bubble sort the results retrieved in **Task 3.2 a)** according to the average number of beds per facility. Then, display the top 3 years which has the highest average number of beds per facility using the format below.

The three years that have the highest average number of beds per facility are: _____, _____ and _____.

[7]

Task 3.3

(a) Write a MongoDB Pymongo query and program code to display all "_id"s of Not-for-Profit health facilities documents that have no facility. [3]

(b) Write MongoDB Pymongo code to update the fields "no_of_facilities" and "no_beds" of only 3 documents retrieved in **Task 3.3 a)** to 1 and a random number from 10 to 20 respectively. [6]

61 [RVHS/PRELIM/9569/2021/P2/Q4]**Car Loaning System**

CaRent is a company providing electronic car rental services. The company engages you to design a web application using Flask microframework to aid in the car rental process.

The following information of each Customer is stored:

`CustomerID` – auto increment integer value to keep track of the ID of customer.

`Name` – name of customer.

`Gender` – gender of customer, to be stored as a single character, using either 'M' or 'F'.

`Contact` – contact number of customer.

The following information of each Car is stored:

`VIN` – vehicle identification number (VIN) of the car.

`Brand` – brand of the car.

`Vehicle Type` – type of the car, can be 'Sedan', 'Hatchback', 'SUV' or 'MPV'.

`Energy Source` – type of energy source the engine is running on, can be 'Diesel', 'Gasoline', 'Hybrid' or 'Electricity'.

`DailyPrice` – daily price for renting the car.

`Availability` – availability of the car, can be 'Available' or 'Unavailable'.

The following information of each RentalPoint is stored:

`PointID` – auto increment integer value to keep track of the ID of rental service point.

`Address` – address of the rental point.

`OpWeekDay` – weekdays that the rental point is open, stored as a 7-digits string, starting from Sunday to Saturday, with '1' indicating open and '0' indicating closed. E.g. '0111110' means it is open on weekdays and closed on weekend.

`OpStartHr` – starting time of daily operation, stored as a 4 digits string, using 24hour time format.

`OpEndHr` – ending time of daily operation, stored as a 4 digits string, using 24hour time format.

The following information of each RentalRecord is stored:

`CustomerID` – ID of customer.

VIN – VIN of car.

StartDate – start date for the rental service.

CollectionPointID – ID of the collection point.

ReturnDate – return date for the rental service.

ReturnPointID – ID of the return point.

The information is to be stored in four tables:

Customer

Car

RentalPoint

RentalRecord

Task 4.1

Create an SQL file called `Task4_1.sql` to show the SQL code to create the database `car_rental.db` with the three tables.

The table `Customer` must use `CustomerID` as its primary key, the table `Car` must use `VIN` as its primary key, and the table `RentalPoint` must use `PointID` as its primary key.

The table `RentalRecord` should use `CustomerID`, `VIN` and `StartDate` as a composite key, while `CustomerID`, `VIN` and `CollectionPointID/ReturnPointID` must refer to `CustomerID` in `Customer`, `VIN` in `Car` and `PointID` in `RentalPoint` as foreign keys.

Save your SQL code as `Task4_1.sql`

[6]

Task 4.2

The files `customers.csv`, `cars.csv`, `rental_points.csv` and `rental_records.csv` contains information about the customers, cars, rental points and the past rental records. The first row of each file contains the header of the respective columns. Each row in the files is a comma- separated list of information.

Write a Python program to insert all information from the three files into the database `car_rental.db`. Run the program.

Save your program code as `Task4_2.py`

[6]

Task 4.3

You are tasked to implement a function to search and display all past rental records of a customer. Using the customer's name 'Goh Yi Xi', query and display a list of data with the following fields as shown in the table, sorted in the ascending order according to the start date.

Name	Contact	VehicleType	StartDate	ReturnDate	DailyPrice
...

Write the SQL code required.

Save this code as `Task4_3.sql`

[5]

Task 4.4

The company wants to implement a function to register new cars for rental into the database. Office staff can register new cars by adding the values of the attributes in the `Car` table.

Write a Python program and the necessary files to create a web application that:

- Receive the following information:
 - VIN, Brand, VehicleType, EnergySource, and DailyPrice of a car through a HTML form.
 - Availability should be set to the default value of 'Available'.
 - Note that VehicleType and EnergySource should be in **dropdown** list format to improve data validity.
- Check if the VIN is valid based on the following algorithm:
 - Step 1: Translate all letters to integer values using the following table (I, O, and Q are not allowed in a valid VIN):

A: 1	B: 2	C: 3	D: 4	E: 5	F: 6	G: 7	H: 8	N/A
J: 1	K: 2	L: 3	M: 4	N: 5	N/A	P: 7	N/A	R: 9
N/A	S: 2	T: 3	U: 4	V: 5	W: 6	X: 7	Y: 8	Z: 9

- Step 2: Use the following weight factor for each position in the VIN. **The 9th position is that of the check digit.** Its weight factor has been substituted with a 0, which will cancel it out in the multiplication step.

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Weight	8	7	6	5	4	3	2	10	0	9	8	7	6	5	4	3	2

- The sum of product of the letter/digit with their corresponding weight factor is then divided by 11.

- The remainder is the check digit. If the remainder is 10, the check digit will use X instead.
- E.g. the VIN with values 1M8GDM9A_KP042788 will produce a check digit of X and hence 1M8GDM9AXKP042788 is a valid VIN.
- If VIN is valid, create a new car record in the Car table, and display the record in the confirmation page.
- Otherwise, inform the user that the VIN is invalid.

<p>Input Page</p> <h3>New Car Record</h3> <p>VIN: <input type="text" value="1M8GDM9AXKP042788"/></p> <p>Brand: <input type="text" value="Toyota"/></p> <p>Vehicle Type: <input type="text" value="Sedan"/></p> <p>EnergySource: <input type="text" value="Gasoline"/></p> <p>Daily Price: <input type="text" value="305.2"/></p> <p><input type="button" value="Submit"/></p>	<p>Valid VIN and register successful:</p> <h3>New Car Registered Successfully!</h3> <p>A new car record have been registered:</p> <table border="1"> <tr><td>VIN</td><td>1M8GDM9AXKP042788</td></tr> <tr><td>Brand</td><td>Toyota</td></tr> <tr><td>VehicleType</td><td>Sedan</td></tr> <tr><td>EnergySource</td><td>Gasoline</td></tr> <tr><td>DailyPrice</td><td>305.2</td></tr> <tr><td>Availability</td><td>Available</td></tr> </table> <p>Invalid VIN:</p> <h3>Invalid VIN Number</h3> <p>The following VIN number that you entered is invalid.</p> <p>VIN: 1M8GDM9A2KP042788</p>	VIN	1M8GDM9AXKP042788	Brand	Toyota	VehicleType	Sedan	EnergySource	Gasoline	DailyPrice	305.2	Availability	Available
VIN	1M8GDM9AXKP042788												
Brand	Toyota												
VehicleType	Sedan												
EnergySource	Gasoline												
DailyPrice	305.2												
Availability	Available												

You may assume:

- All inputs are in valid format.
- VIN: 1M8GDM9AXKP042788 is a new record to the database

Save your program as Task4_4.py

With additional files or sub-folders as needed in a folder named Task4_4

Run the web application. Enter the values based on the sample input above.

Then save the output of the program as Task4_4.html.

[15]

62 [YIJC/PRELIM/9569/2021/P1/Q1]

An iterative function Fn has two parameters, Arr and Object, and returns an integer. The pseudocode is as follows:

```

01 FUNCTION Fn(Arr: ARRAY, Object: STRING) RETURNS INTEGER
02     Current ← 1
03     REPEAT
04         IF Object = Arr[Current]
05             THEN
06                 RETURN Current
07             ENDIF
08         IF Object > Arr[Current]
09             THEN
10                 Next ← 1 + Current * 2
11             ELSE
12                 Next ← Current * 2
13             ENDIF
14         Current ← Next
15     UNTIL Current > LENGTH(Arr)
16     RETURN -1
17 ENDFUNCTION

```

A

binary search tree is used to store the names of the 12 Chinese Zodiac animals. The order in which these names were added into the tree follows the order in the array X.

Array X			
X[1]	'Rat'	X[7]	'Tiger'
X[2]	'Monkey'	X[8]	'Dog'
X[3]	'Snake'	X[9]	'Horse'
X[4]	'Dragon'	X[10]	'Ox'
X[5]	'Pig'	X[11]	'Rabbit'
X[6]	'Sheep'	X[12]	'Rooster'

(a) Draw the binary tree using the array X.

[3]

(b) Complete the trace table templates provided for the following function calls and state the RETURN value

after each function call:

Current	Object = Arr[Current]	Object > Arr[Current]	Next
Output:			

(i) Function call: Fn(X, 'Sheep')

[2]

(ii) Function call: Fn(X, 'Duck')

[3]

(c) Describe the purpose of function Fn.

[2]

(d) Explain why this function Fn is more efficient than linear search.

[2]

(e) It is found that one of the Zodiac animals should be named as 'Goat' instead of 'Sheep'. Design another array Y such that the function Fn can be used with the correct list of the Chinese Zodiac animals.

Complete the template provided for the array Y.

[2]

Template for Question 1(b)

(i) Function call: Fn(X, 'Sheep')

Current	Object = Arr[Current]	Object > Arr[Current]	Next
Output:			

(ii) Function call: Fn(X, 'Duck')

Current	Object = Arr[Current]	Object > Arr[Current]	Next
Output:			

Template for Question 1(e)

Array Y			
Y[1]		Y[7]	
Y[2]		Y[8]	
Y[3]		Y[9]	
Y[4]		Y[10]	
Y[5]		Y[11]	
Y[6]		Y[12]	

63 [YIJC/PRELIM/9569/2021/P1/Q2]

A merge sort algorithm consists of the function `merge_sort(seq)` that takes in an unsorted list `seq` as an input and returns a sorted list. The function uses the helper function `merge(left, right)` which merges the two sorted lists, `left` and `right`, and returns a sorted list.

The following is the pseudocode for the function `merge(left, right)`:

```

01 FUNCTION merge(left: LIST, right: LIST) RETURNS LIST
02     IF LENGTH(left) = 0
03         THEN
04             RETURN right
05         ELSE
06             IF LENGTH(right) = 0
07                 THEN
08                     RETURN left
09             ENDIF
10         ENDIF
11     IF left[0] < right[0]
12         THEN
13             RETURN [left[0]] + merge(left[1:], right)
14         ELSE
15             RETURN [right[0]] + merge(left, right[1:])
16     ENDIF

```

- (a) Explain why this is a recursive function. [2]
- (b) State whether this `merge` implementation is stable or unstable and explain with an example. [2]
- (c) Complete the merge sort algorithm by writing the pseudocode for the function `merge_sort(seq)`. [4]
- (d) State **one** advantage and **one** disadvantage of a merge sort algorithm over a bubble sort algorithm. [2]
- (e) Explain whether using a merge sort algorithm or a bubble sort algorithm will be more efficient in arranging the data list `[2, 1, 3, 4, 6, 5, 8, 7, 9, 10, 12, 11]` in an ascending order. [3]

64 [YIJC/PRELIM/9569/2021/P1/Q3]

A stack is a last-in-first-out (LIFO) abstract data type (ADT) in which all the elements are inserted and removed from one end.

It is common to either use a linked list or an array to implement a stack

- In the linked list implementation, a root pointer points to the top of a stack and a data structure. The data structure contains the value of the data and a pointer pointing to the next node in the stack.
- In the array implementation, a fixed size array is used to store the elements.

The basic stack operations of `push()`, `pop()` and `peek()` are provided in both implementations.

- `push()` is used to insert an element into the stack.
- `pop()` removes an element from the stack and returns the value of the element.
- `peek()` returns the value of the element at the top of the stack without removing it.

- (a) Describe what an abstract data type is and how it benefits the user. [3]
- (b) State one advantage and one disadvantage of implementing the stack ADT using a linked list. [2]
- (c) State one advantage and one disadvantage of implementing the stack ADT using an array. [2]
- (d) Describe how a `push()` operation is done in a stack ADT which is implemented using a linked list. [3]
- (e) Describe how the number of elements within a stack can be counted using only the basic stack operations provided. [3]

The following program code uses a Python list as an array with the built-in functions `<list>.insert()` and `<list>.pop()`.

```

01 def add(seq):
02     stk = []
03     def InsertOne(item):
04         if stk==[] or item < stk[0]:
05             stk.insert(0,item)
06     else:
07         temp = stk.pop(0)
08         InsertOne(item)
09         stk.insert(0,temp)
10     for ele in seq:
11         InsertOne(ele)
12     return stk

```

- (f) (i) The above is an example of implementing a stack using an array. State and explain how the program code adds all the elements in the sequence `seq` into the stack. [2]
- (ii) Modify the above program code such that it uses only the basic stack ADT operations provided. [4]

65 [YIJC/PRELIM/9569/2021/P1/Q4]

The following shows a sample of a student's result slip for the Preliminary Examination.

AUDE EXAURARE			
JUNIOR COLLEGE			
2021 JC2 Preliminary Examination			
<hr/>			
Name:	TAN XIAO MING		
Identification No.:	T0312321E		
Civics Group:	20S23		
Index No.:	18		
Subject	Code	Mark	Grade
H1 General Paper	8807	55	C
H1 Economics	8823	51	D
H2 Mathematics	9758	66	B
H2 Physics	9749	62	B
H2 Computing	9569	78	A

The college wishes to manage this result information using a relational database. The normalised database design requires to have a number of tables.

(a) Draw an Entity-Relationship (E-R) diagram showing these tables and the relationships between them. [4]

(b) A table description can be expressed as:

TableName (Attribute1, Attribute2*, Attribute3, ...)

The primary key is indicated by underlining one or more attributes. Foreign keys are indicated by using a dashed underline/asterisk. Write table descriptions for the tables you identified in **part (a)**. [5]

(c) Under the Personal Data Protection Act (PDPA), the NRIC/FIN can no longer be used as a unique identifier for each student. Suggest and justify a suitable alternative unique identifier for each student. [2]

(d) Write an SQL query to output the names, civics groups and grades of students who have obtained at least 60 marks for H2 Computing. [4]

66 [YIJC/PRELIM/9569/2021/P1/Q5]

The Food Services Industry Digital Plan (IDP) was recently launched by the Minister of State for Trade and Industry to help F&B businesses adopt more digital technologies in their daily operations. The manager of a local restaurant engaged a consultant to propose a digital solution for his restaurant operations.

After conducting a comprehensive study, the consultant proposed a web-based solution using a client-server model. The solution requires the following hardware to access the web server wirelessly in the local area network (LAN):

- A tablet device on each table for customers to browse the menu and order their food items.
- Multiple large monitors for the chefs in the kitchen to read the ordered food items.
- A computer station for the service staff to check the table number before serving the food to the customers.
- A computer in the manager's office to update the menu in the web server and print the daily sales report.

When a customer decides to pay the bill, a QR code will be generated on the tablet device for him to scan and make online payment using his personal mobile device.

- (a) Explain the meaning of the term client-server model. [1]
- (b) Describe the main software components to be developed for the web server to host this service for the restaurant. [2]
- (c) Describe how the customer would use the client tablet device to browse and order the food items. [3]
- (d) Suggest **one** feature on the digital form that will provide a positive experience for the customers when using the tablet device to order their food. Describe the usability principle applied in this feature. [2]
- (e) The manager recommends the proposed solution to the shareholders, but a number of social issues associated with the solution have been raised. Describe **two** possible issues that could have been raised. [2]

An alternative to this web-based solution would be to develop a native application programme for the customers to download and install on their mobile devices.

- (f) Describe **one** feature that is only available in the native application solution and how it is relevant to the solution proposed for the restaurant. [2]

The restaurant's manager is also keen to expand his business to accept online ordering for takeaways.

- (g) Explain **one** benefit of the web-based solution in this situation. [1]
- (h) Draw the network diagram for the proposed web-based solution and include all the required hardware for the restaurant to accept online ordering. [5]
- (i) Describe **two** benefits for the restaurant in implementing this solution. [2]
- (j) For online ordering, the restaurant needs to collect the customer's name, address and contact number. State and describe **two** data protection obligations that the manager needs to comply under the Personal Data Protection Act. [4]

67 [YIJC/PRELIM/9569/2021/P1/Q6]

A fantasy card game was developed using object-oriented programming (OOP) to store its cards' data. A card can either be a minion or a weapon. Each card has a name, mana cost, health or durability and attack power.

In order to play a card, a player must spend a certain amount of mana as specified in the card's mana cost. When the card has been played, the player may decide whether to use it to attack another minion or not.

A minion may belong to one of the following races: beast, demon, dragon or elemental. When a minion is attacked, its health would decrease according to the attacking card's attack power. Once the health of a minion decreases to zero, it is destroyed and removed from the game.

Instead of health, a weapon has durability and it cannot be attacked. When a weapon is used for an attack, its durability would decrease by one. Once the durability of a weapon decreases to zero, it is destroyed and removed from the game.

(a) Draw a class diagram, showing:

- the base class CARD,
- any sub-classes and inheritance from the base class,
- the properties for the base class and sub-classes,
- appropriate methods with at least one getter and one setter method.

[7]

(b) In relation to your diagram in part **(a)**, explain the terms:

- (i)** encapsulation,
- (ii)** inheritance,
- (iii)** polymorphism.

[6]

(c) Explain why OOP is a preferred programming paradigm in the development of this game.

[2]

68 [YIJC/PRELIM/9569/2021/P2/Q1]

During a contact tracing exercise, the TraceTogether token's tag number, the user's name and the user's mobile number are stored in an Abstract Data Type (ADT) `Person`. The information are stored as a three element tuple as follows:

(tag: INTEGER, name: STRING, hp: STRING)

Function	Return Type	Description
<code>make_person(tag,name, hp)</code>	Person ADT	A Constructor to create the Person ADT.
<code>get_tag(Person)</code>	INTEGER	Returns the tag number of the user.
<code>get_name(Person)</code>	STRING	Returns the name of the user.
<code>get_hp(Person)</code>	STRING	Returns the mobile number of the user.

Task 1.1

Write program code to implement the ADT `Person` with the above constructor and accessors. [4]

Task 1.2

Write program code for the function `read_file()` to read all the 21 users' information from the file `PEOPLE.csv` and return a list containing all the `Person` ADTs. [5]

Task 1.3

Write program code for the function `insertion_sort(lst)` that takes the list `lst` obtained from **Task 1.2** and sort them according to their tag numbers in an ascending order using the insertion sort algorithm. [5]

Searching for a tag number in a large list of users may be slow and tedious. However, if the list is sorted, performing a binary search would be more efficient.

Task 1.4

Write program code for the function `search(lst,num)` that searches for a tag number `num` in the sorted list `lst` obtained from **Task 1.3**. The code should

- perform a binary search on the sorted list.
- return the `Person` ADT if the tag number `num` exists. Otherwise, return `None`.
- print the number of comparisons made in the searching process.

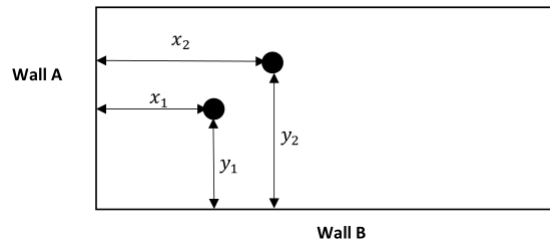
[8]

69 [YIJC/PRELIM/9569/2021/P2/Q2]

Every user is required to carry a tracing token inside an indoor sports facility so that the sensing device system can detect the token to read the position of the users and their temperature. The data is stored as records in the file `POSRECORDS.txt`, with the following entries separated by commas:

- the tag number (INTEGER) of the tracing token which could be used to identify the user
- the temperature (FLOAT) of the user measured in degree Celsius
- the location (x, y) (FLOAT, FLOAT) of the user which consists of the perpendicular x - and y -distances, measured in metres, from the walls A and B respectively

The diagram below shows two locations (x_1, y_1) and (x_2, y_2) :



The distance between two locations can be calculated using the formula:

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Task 2.1

Create an empty array `all_records` of size 20 and paste the data from the text file `POSRECORDS.txt` into your program code.

Write program code for the function `distance(user1, user2)` that takes in two records and returns the distance between the two users in metres, correct to 2 decimal places. [3]

For safety reasons during the Covid-19 pandemic, two users are considered to be in close proximity if the distance between them is less than 1.5 m apart.

Task 2.2

Write program code for the function `analyse(user1)` that iterates through all the records in the array `all_records`, calculates the distances between `user1` and all other users in the array, and returns a list of tag numbers of the users who are in close proximity to `user1`. [5]

Under the Safe Management Measures, people with a temperature of more than 37.5 degree Celsius will be flagged out as RED cases and the people who are in close proximity to these RED cases will be flagged out as YELLOW cases. The facility manager needs to submit both lists to the authority daily for follow-up actions.

Task 2.3

Write program code for the function `red_list(all_records)` that iterates through all the records in the array `all_records` and returns a list of tag numbers belonging to the RED cases. [2]

Task 2.4

Write program code for the function `yellow_list(red_cases, all_records)` that iterates through all the records in the array `all_records` to check for people who were in close proximity to any of the RED cases. The function returns a list of tag numbers belonging to the YELLOW cases.

Those people flagged out as RED cases in **Task 2.3** should not appear in the list of YELLOW cases even though they may be in close proximity to another RED case.

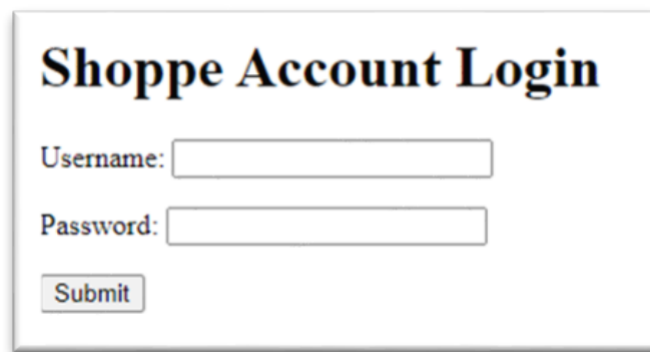
You may use the functions written in Task 2.1 and Task 2.2 for the program code in this task. [5]

70 [YIJC/PRELIM/9569/2021/P2/Q3]

Shoppe e-Commerce has a mobile application for customers to make purchases through its online platform. All the customers' details, product data and ordering records are kept in the database `shoppe.db`.

Task 3.1

Write program code for the webpage `index.html` for the customer to log in to their account. The `/login` route in the server code should verify the customer's username and password using the data in the `Account` table in the database.


A screenshot of a web form titled "Shoppe Account Login". The form has a white background with a thin grey border. It contains two text input fields: "Username:" and "Password:". Below the password field is a "Submit" button. The text is in a serif font.

If the log-in details are valid, the customer will receive the webpage `display.html`. Otherwise, the customer will be redirected back to the log-in page. [7]

Task 3.2

Write program code for the webpage `display.html` to display the customer's details, with the profile picture, and a menu for the customer to choose the option to update the profile picture or to check the shopping cart.

Customer Details



UserID: c4322
Username: Mary
Email: marylim@yahoo.com

Menu Options

☐ Click here to update your profile picture.
☐ Click here to check your shopping cart.

[Return to Login Page](#)

[6]

A customer, John, does not have a profile picture and he now wishes to upload the file `mypic.png`. This picture file will be renamed as `John.png` before storing in the web server.

Task 3.3

Write program code for the customer to select and upload a picture file, and it should include the following:

- `/menu` route in the server code to provide the customer with a webpage `profile.html` when the option to update the profile picture is chosen
- `profile.html` to allow the customer to upload a profile picture in the `.png` format
- `/update` route in the server code to receive the uploaded picture file and rename it as `<username>.png` before storing in the server's `\static\photo\` directory
- `success.html` to display a webpage informing the customer that the profile picture has been successfully uploaded

[7]


The Shoppe customers usually browse through the available products on the platform and add them to their shopping carts. When they have decided on their purchase, they will select some or all the items in the shopping cart before checking out to make payment.

Userid : c4322

Username : Mary

Select the items in your shopping cart to checkout:

Product Name	Unit Price	Quantity	Select
Toilet Rolls	5.3	3	<input type="checkbox"/>
3-in-1 Coffee	5.2	1	<input type="checkbox"/>
Marker	1.5	2	<input type="checkbox"/>
Canon Ink Cartridge	45.0	3	<input type="checkbox"/>
Pencil	0.3	4	<input type="checkbox"/>
Water Colour	3.0	1	<input type="checkbox"/>

Please select a delivery date: 

Please provide the delivery address:

[Return to Login Page](#)

Task 3.4

Write program code for the customer to select the items to check out for payment, and it should include the following:

- /menu route in the server code to query the Cart table in the shoppe.db when the customer chooses the option to check the shopping cart
- cart.html to display the list of items in the shopping cart and let the customer select them for checking out; the customer will also be required to indicate the preferred date and address for delivery
- /checkout route in the server code to receive the customer's inputs and insert a record into the Orders table in the shoppe.db
- success.html to display the **total cost** and inform the customer that the purchase has been successfully recorded.

[14]

71 [YIJC/PRELIM/9569/2021/P2/Q4]

Lessonology is a learning management system that utilises gamification elements to motivate students to complete their assignments. The Linked List data structure is used to store the students' names and their total experience points. Each node contains a student's name, the student's total experience points, and a pointer to the next node. The nodes are linked together according to the order provided in the `DATA.txt` file.

A program is to be written to implement nodes as an instance of the class `Node`. The class `Node` has the following properties and method:

Class: <code>Node</code>	
Properties	
Identifier	Description
<code>Name</code>	The node's value for a student's name.
<code>Exp</code>	The node's value for the student's total experience points.
<code>Pointer</code>	The pointer to the next node.
Method	
Identifier	Description
<code>SetPointer()</code>	Set the pointer to point at the next node or point to <code>None</code> when it is the last node.

A linked list is implemented as an instance of the class `StudentList`. The class `StudentList` has the following property and methods:

Class: <code>StudentList</code>	
Properties	
Identifier	Description
<code>Start</code>	The pointer at the start of the linked list.
Methods	
Identifier	Description
<code>Constructor</code>	Initialise the linked list with the pointer <code>Start</code> assigned to <code>None</code> .
<code>Add()</code>	Add a new node into the linked list.
<code>Update()</code>	Update the value for the total experience points of a student's node in the linked list.
<code>Delete()</code>	Delete a node in the linked list.
<code>Display()</code>	Display the current content of the linked list in table form.

Task 4.1

Write program code for the classes `Node` and `StudentList`, including the `Constructor`, `Add()` and `Display()` methods. The code should follow the specification given. Do not write the `Update()` and `Delete()` methods yet.

The `Add(node)` method for the `StudentList` class should add the `node` containing a student's name and the student's total experience points to the linked list, according to the order given in the `DATA.txt` file.

Test your code by reading the data from the file `DATA.txt` and adding them as nodes into the linked list. The diagram below shows a portion of the expected output when using the `Display()` method on the populated linked list:

Name	Experience Points
ANDREW	17616
ANGIE	16001
AU YONG	15589
AZMAN	775
BENG CHOO	15411
BOB	6244
BRIAN	20404

[9]

Task 4.2

Each time a student completes an assignment, points will be awarded and the student's total experience points will be updated.

Write program code for the `Update(name,points)` method for the `StudentList` class that takes a student's `name` and the awarded `points` as inputs to update the student's total experience points in the node. (You may assume that the node containing the student exists in the linked list.)

For example, `Update('BRIAN',100)` will update the total experience points of a student whose name is 'BRIAN' from 20404 to 20504.

[3]

Task 4.3

Write program code to implement the `Delete(name)` method for the `StudentList` class to search and remove a node, containing a particular student's `name`, in the linked list. Return `True` if the node is found and removed; otherwise return `False`. (You may assume that the students' names are unique in the linked list.)

[4]

Task 4.4

Another linked list which has pointers linking the nodes in decreasing order of the experience points is implemented as an instance of the class `Leaderboard`.

The class `Leaderboard` has the following properties and methods:

Class: <code>Leaderboard</code>	
Properties	
Identifier	Description
<code>Start</code>	The pointer at the start of the linked list.
Methods	
Identifier	Description
<code>Constructor</code>	Inherit the property and all the methods from the class <code>StudentList</code> . Initialise the linked list with the po
<code>Add()</code>	Modify the <code>Add()</code> method in the parent class to add a new node in decreasing order of total experience
<code>Update()</code>	Modify the <code>Update()</code> method in the parent class such that the linked list is still in decreasing order of ex
<code>DisplayTop()</code>	Display the content of the nodes in the linked list for the top students based on their total experience po

Write program code for the class `Leaderboard` to inherit the properties and methods from the class `StudentList` with the modified `Add()` and `Update()` methods. The additional `DisplayTop(n)` method should display the top `n` number of students in the linked list, based on their total experience points. (You may assume that no two students have the same total experience points.)

Test your code by reading the data from the file `DATA.txt` and adding them as nodes into this linked list. The diagram below shows the expected output when using the `DisplayTop(5)` method on the linked list:

```

Displaying Top 5 students
Name          | Total Experience Points
-----
HENDERSON    |          21653
YOCK TIM     |          20740
HUI FANG     |          20563
BRIAN        |          20404
DESMOND      |          20033
-----End of Display-----

```