**1** In a real time monitoring application, an unsorted list of one million numbers are being collected and stored as an array every one minute. The application requires a unique value to be found in this sequence of one million numbers in order to continue running otherwise the application will terminate. Therefore, a search operation needs to be done on the array every minute.

**(a)** Describe the main differences between a linear search and a binary search algorithm. [2]

**(b)** Which of the two search algorithms described in **(a)** is more appropriate to be use by the above application? You have to justify your answer by analysing the worst case run time efficiency of the two solutions. [2]

**(c)** Using pseudo code, describe a linear search function to return the index of the smallest value in an array `A`, **larger** than a value `k`. If there is no such value a value of `-1` is returned.

```
FUNCTION LinearSearch(A: Array[1:N] OF INTEGER, k:INTEGER) RETURNS INTEGER
```

For example,

`LinearSearch([17, 39, 37, 47, 15], 35)` should return `3`. [4]

**(d)** If a binary search algorithm is to be implemented, there are two ways to implement it in a programming language, either using a recursive function or an iterative function. What issues do you need to consider when deciding whether to use the iterative or recursive version of the function. Which version will you choose for the application described above and why? [2]

**(e)** Given an array `A` with `N` elements, write the pseudo code for a recursive binary search function that will return the index of any occurrence of a value k in the array. A value of `-1` is returned if not found. You may describe additional helper functions that you deemed necessary.

```
FUNCTION BinSearch_1(A: ARRAY[1:N] OF INTEGER, k:  INTEGER )
RETURNS INTEGER
```

[4]

**(f)** Using pseudo code write an iterative version of the binary search to return the largest value in the array **smaller** than `k` in the array. If no value is found `-1` is returned.

```
FUNCTION BinSearch_2(A: ARRAY[1:N] OF INTEGER, k:  INTEGER )
RETURNS INTEGER
```

For example, the return value for `BinSearch_2([2,4,6,8],6)` should return `4`. [4]

**(g)** Describe **four** test cases that you will use to test the correctness of your algorithm in **(f)**. Your test cases should cover all the test categories. [4]

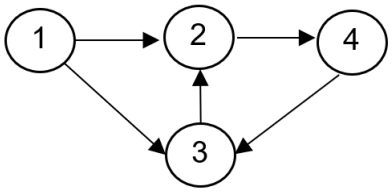
**(h)** Modify the recursive algorithm in **(e)** to return the lowest and highest indexes in the array of all values that matches the key value `k`.

The algorithm must have a run time of $O(\log N)$ where $N$ is the size of the array. You may describe additional helper functions that you deemed necessary.

```
FUNCTION BinSearch_3(A:ARRAY[1:N] OF INTEGER, key:  INTEGER)
RETURNS ARRAY[1:2] OF INTEGER
```

For example, `BinSearch_3([2,4,4,4,6,8],4)` return value is `[2,4]`. [8]

**2** A directed graph can be used to show relationships between objects. It consists of a set of nodes used to represent the objects and a set of edges used to represent the relationship between objects. For example, a visual representation of a directed graph looks like this:




The directed graph above can be used to represent the possible direct connectivity between a pair of nodes. For example, Node 1 have a direct path to Node 2 and Node 3. The edges are directional, meaning Node 1 can reach Node 2 but Node 2 cannot reach Node 1.

We can use a Node class to represent a Node object as follows:

| Node |
|---|
| -ID: INTEGER |
| +constructor(ID: INTEGER)<br>+getID() : INTEGER |

A List data structure can be used to represent the set of nodes that are directly reachable from a particular node. For example, the following incomplete table shows how Lists can be used to represent the directly connected neighbouring nodes from a source node.

| **Source** | **Directly Connected Neighbours** |
|---|---|
| Node(1) | [ Node(2), Node(3) ] |
| Node(2) | |
| Node(3) | |
| Node(4) | [ Node(3) ] |

**(a)** Complete the table above. [2]

**(b)** Using a UML class diagram or otherwise describe the methods that needs to be implemented for such a List data structure. Your methods must be able to support the requirements of part **(d)**. [3]

**(c)** Another data structure is needed to implement the directed graph as described above.

Describe this data structure and its operations and explain **why** you choose it. Your solution must support the requirements of part (d). [3]

**(d)** An algorithm is to be implanted using only the Node object and the two data structures that you described in **(b)** and **(c)**. All operations on the data structures that you use must be fully described in **(b)** and **(c)**. The algorithm is to return a List object containing all the Node objects' ID values that are reachable from a source Node. Describe the algorithm as a function in pseudo code. The function header is as follow:

```
FUNCTION FindPath(graph: OBJECT, source: INTEGER) RETURNS List
```

where `graph` is the data structure described in **(c)**, `source` is the ID value of the source Node,

and List is a list of integer values representing the Nodes that are reachable from the source Node.

For example, based on the graph represented above, the following function call,

`FindPath(graph,2)` should return a List consisting of the Nodes' IDs `[2,4,3]`.

Node 2 is the source; Node 4 is a directly connected from Node 2 and Node 3 is reachable from Node 4.

You may describe other helper functions in your algorithm if necessary. The order in the List does not matter. [6]

**(e)** Using pseudo code, describe an algorithm that can determine which nodes in the graph is able to reach all other nodes. [3]

**3** In a Hash table implementation, a hash function is used to generate a key to perform a lookup on the Hash table. They key will determine where the data is stored on the Hash table.

**(a)** Give two main properties of a good hash function. [2]

**(b)** When a hash function generates the same key when given two different input data, this is known as a collision. One method of resolving a collision is to use separate chaining. Explain how the separate chaining algorithm works. [2]

**(c)** The other method for resolving collision is called linear probing. Give one disadvantage of the linear probing method [1]

**(d)** The following algorithm uses a probing method to resolve collision:

```
1 FUNCTION Insert(table:ARRAY[0:99] OF STRING, name:STRING) RETURNS BOOLEAN
2 //returns True if name is inserted into table else False
3     key ← HASH(name) // returns an integer from 0 to 99
4     IF table[key] = NONE //NONE is used to denote an empty slot
5     THEN
6         table[key] ← name
7         RETURN True
8     ENDIF
9     step ← 1
10     WHILE True
11     DO
12         i ← (key + step * step ) MOD LEN(table) //LEN returns size of array
13         IF table[i] = NONE
14         THEN
15             <A>
16         ELSE IF i = key
17         THEN
18             <B>
19         ELSE
20             <C>
21         ENDIF
22     ENDWHILE
23 ENDFUNCTION
```

**(i)** Fill in the pseudo code `<A>` at line 15. [1]

**(ii)** Fill in the pseudo code `<B>` at line 18. [1]

**(iii)** Fill in the pseudo code `<C>` at line 20 [1]

**(iv)** Explain how this algorithm is different from the linear probing method and one advantage of this algorithm over linear probing. [2]

**(v)** Identify and explain one disadvantage of this algorithm. [2]

**4** On the Internet, there are many network protocols used by different server applications to communicate with a client.

**(a)** Explain what is a network protocol? [1]

**(b)** One example of a protocol used by a server on the Internet is the DNS protocol. Describe the purpose of the DNS protocol. [1]

**(c)** Is the DNS system used on the Internet a centralised or de-centralised system? Explain. [2]

**(d)** For the following URL:

`http://nationaljc.moe.edu.sg:8088/20SH07`,

describe the different parts in the URL and their significance. [2]

**(e)** Explain the processes involved after you type the URL, described in **(d)**, on a web browser address bar. [2]

**(f)** A native email client, example Microsoft Outlook needs to be installed on your desktop in order to send and receive emails to and from an email server.

What is the difference between using a native email client as described above and using a web browser to access a Web application like Gmail to send and receive emails? [2]

In web scraping, program code is written to read the html content from a web server and extract data from the html content.

You have just discovered that a number of hidden pages in a school's web site that display personal information of its staff and students.

**(g)** Describe using structured English or pseudocode the steps that you need to perform in order to write a socket-based web scraping program to extract the personal data of the staff and students from the school's web site. [3]

**(h)** What are the ethical issues in writing the web scraping program? [2]

**(i)** The hidden pages' URL described above is known only to the staff and students of the school to facilitate communication among the staff and for students to contact the school staff for urgent matters during the full home-based learning period. Explain why this is not a good solution in the context of the Personal Data Protection Act in Singapore. [2]

**(j)** Suggest an alternate solution for the staff and students to share personal information as described in **(i)**. [1]

When writing the web scraping program, you realised that the string data cannot be displayed properly when you try to print them on a terminal console, but the content looks fine when displayed on a web browser. When you try to debug your program, you discovered the following html snippet on the school's web page:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta http-equiv="Content-Type"
      content="text/html;
      charset=utf-16"
/>
```

**(k)** What could be a possible cause of the problem described above. [2]

**5** A bike sharing company, Toys2Share, has bicycles available for rental all over Singapore. These bicycles are deployed in self-service Toys2Share booths in different locations.

Toys2Share bikes can only be checked out and checked in from these automated booths. Customers will use a native app installed in their smartphone to checked out a bike from any Toys2Share booth and return their bike to any Toys2Share booth. Toys2Share keeps an inventory of all the bicycles for rental in a relational database. There are two types of bicycles for rental, the normal mechanical bicycle and the electrical bicycle (e-bike) which uses a battery-operated electrical system to power the bicycle.

The data kept for all bicycles (mechanical and electrical) are as follows:

- Serial number
- Model number
- Year of manufacture
- Date of last service
- Mileage (distance in km accumulated)

For the e-bike, additional data needs to be stored:

- Power of electrical motor (in watts)
- Battery capacity (in Amp-hours)
- Current charge of battery (percentage, of the charge left on the battery)

The mileage and current charge of the battery are updated when the customer return and check in the bike in the Toys2Share booth. The e-bike's battery will also be charged.

A customer needs to register as a member with Toys2Share and download the app before he/she can check out any bike from the booths. Toys2Share requires the following data from the customer:

- Name
- Contact Number
- Credit Card Number
- Membership Tier.

The membership tier consists of three levels: Ordinary, Premier and VIP. The monthly fees for the three tiers are $1.00, $5.00 and $10.00 respectively. The customer may change his membership tier after each month.

When a customer successfully checked out a rental transaction record has to be stored on the database consisting of the following information.

- Customer that checks out the bike
- The serial number of the bike
- Check out date and time
- Check out booth
- Check in date and time
- Check in booth
- Rental fee

The check in date/time and check in booth data are updated automatically upon check in at a Toys2Share booth. The rental fee is also calculated and updated based on the rounded-up number of hours that the customer has check out the bike and the rental rate.



**[Turn Over**

The rental rate is based on the membership tier that the customer belongs to. The rental rates for the 3 tiers are as follows:

| **Membership Tier** | **Rate** |
|---|---|
| Ordinary | $0.50 |
| Premier | $0.30 |
| VIP | $0.10 |

In their daily operations, Toys2Share needs to send technicians to service those bikes whose last serviced date has exceeded six months.

Toys2Share plans to include other types of rentals for their customers in the near future. This includes e-scooters, inline skates, drones and other new gadgets that the company may acquire.

**(a)** Draw a normalised ERD of the data model for the relational database to be implemented for Toys2Share. The design of the data model must cater for the daily operational needs and future expansion plans of Toys2Share as described above. [6]

**(b)** Describe the relations in the data model. [4]

**(c)** When a customer wants to check out a bike from a Toys2Share booth, a rental record as described above, needs to be generated in the database. In order to maintain data accuracy and integrity, data validation needs to be performed on the data input by the user. Describe two validation techniques that can be use. [2]

**(d)** When a customer wants to check out a bike, the self-service kiosk will use the following rules to determine whether the checkout is successful :

- If a bike last serviced date is more than twelve months, the bike cannot be checked out.
- If a bike's mileage is more than 1000 km, a warning message will be sent to advice the customer not to ride more than 20 km.
- If it is an e-bike and the current battery charge is less than 50% , the bike cannot be checked out.
- Otherwise, the bike can be checked out.

**(i)** Draw a decision table to describe all the possible conditions and outcomes. [4]

**(ii)** Simplify the decision table in **(i)**. [2]

**(iii)** Describe the simplified table in **(ii)** using a flow chart. You may use OUTPUT statement to describe the action to be taken. [3]



**END**