**ANGLO-CHINESE JUNIOR COLLEGE**

**JC2 PRELIMINARY EXAMINATION**

Higher 2

**COMPUTING** **9569/01**

Paper 1 Written **31 August 2021**

**3 hours**

**READ THESE INSTRUCTIONS FIRST**

An answer booklet will be provided with the question paper. You should follow the instructions on the front cover of the answer booklet. If you need additional answer paper ask the invigilator for a continuation booklet.

Answer **all** questions.

Approved calculators are allowed.

The number of marks is given in brackets [ ] at the end of each question or part question.

The total number of marks for this paper is 100.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

This document consists of **10** printed pages.

**O:\ACJCDepts\acjc\acjc 2\ACJC Crest for Printing Purposes\Archived files DO NOT USE\ACS Crest B&W.tif**



**[Turn Over**

**1** 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** 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'

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

**(ii)** **(a)** Suppose collisions in the hash table are to be resolved using open hashing.

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

**(b)** 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]

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

**3** 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.

PUSH Root of Tree into Stack

Node ← POP from Stack

YES

NO

START

STOP

INPUT Tree, Searchfor

CREATE Stack

Is Stack Empty?

Is Node’s data equal to Searchfor?

Does Node have a left child?

Does Node have a right child?

PUSH left child of Node into Stack

PUSH right child of Node into Stack

NO

NO

NO

YES

YES

OUTPUT

Not Found

OUTPUT

Found

YES

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

Tree

**(ii)** State whether this is a depth-first search or a breadth-first search. [1]

**(iii)** Draw a flowchart to illustrate how the other kind of search in **(ii)** can be carried out using the same input parameters. [3]

**(iv)** Given the same input Tree and the same Searchfor value of 5, draw a trace table to illustrate the algorithm in **(iii)**. [5]

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

**(a)** Thetable 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 atthedatabase 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.

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** 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.

VEHICLE

Type : STRING

Max\_Speed : INTEGER

Current\_Speed : INTEGER

Position : Array[1:2] OF INTEGERS

Petrol\_level : FLOAT

Petrol\_use\_rate : FLOAT

get\_type()

get\_max\_speed()

get\_current\_speed()

get\_position()

get\_petrol\_level()

increase\_speed(INTEGER)

decrease\_speed(INTEGER)

turn\_left()

turn\_right()

add\_petrol(FLOAT)

CAR

Passengers : INTEGER

get\_passengers()

add\_passenger()

remove\_passenger()

TRUCK

Max\_Load : FLOAT

Current\_Load: FLOAT

get\_load()

add\_load(FLOAT)

remove\_load(FLOAT)

**(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** 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: | hayley123 | | |  |
|  |  |  | | |  |
|  | Password: | • • • • • • • • • • • • | | |  |
|  |  |  | | |  |
|  |  | | Login |  |  |
|  |  | |  |  |  |

**(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** 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]