NANYANG JUNIOR COLLEGE
TEMASEK JUNIOR COLLEGE
VICTORIA JUNIOR COLLEGE

JC2 PRELIMINARY EXAMINATIONS

Higher 2

COMPUTING 9569/01

Paper 1 Written **14th September 2022**

3 hours

READ THESE INSTRUCTIONS FIRST

An answer booklet will be provided with this 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 and **1** blank page.

1 The following recursive function returns the highest common factor of two given positive integers.

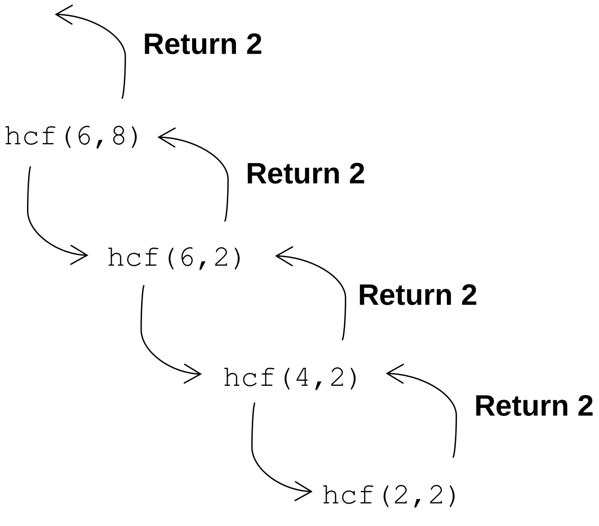
```
01 FUNCTION hcf(a : INTEGER, b : INTEGER) RETURNS INTEGER
02     IF a = b
03       THEN
04         RETURN a
05     ENDIF
06     IF a > b
07       THEN
08         RETURN hcf(a-b, b)
09       ELSE
10         RETURN hcf(a, b-a)
11     ENDIF
12 ENDFUNCTION
```

(a) **(i)** State what is meant by **recursion**. [1]

(ii) State the line numbers that indicate that the function is recursive. [1]

(iii) State the significance of line `02`. [1]

An example of a trace tree diagram showing the recursive function call `hcf(6,8)` is shown below:



(b) Use the above example to create a trace tree diagram for the recursive function call `hcf(15,36)`. [5]

(c) Explain the use of a stack when the recursive function `hcf` executes. [3]

(d) Describe an error that might be returned by this recursive function when the values of `a` and `b` are such that the function `hcf` calls itself many times. [1]

2 A 1-dimensional array containing 8 numbers is shown.

Index	1	2	3	4	5	6	7	8
Data element	11	14	17	23	28	35	36	49

(a) Linear search and binary search are two different algorithms which can be used for searching arrays. Assume every data element in the above array is equally likely to be searched for.

(i) State and explain whether a binary search or a linear search would be the most appropriate method in searching for a specific data element in the array given in Figure 2. [2]

(ii) List the data elements in the order they are accessed, during a binary search for the data element numbered 28. [2]

(iii) Describe a situation where a linear search would complete in a shorter time than a binary search. [1]

(b) The quicksort algorithm was used to produce the sorted data in the table. This implementation of the quicksort utilised the last data element of the unsorted array as the pivot.

(i) State the worst-case time complexity of a quicksort algorithm. [1]

(ii) Describe one improvement that could be made to the quicksort algorithm used in this implementation, to improve its efficiency. [1]

Another sorting algorithm that could be used to sort an array is a bubble sort.

(c) State the time complexity of the bubble sort algorithm. [1]

Two programmers carried out an experiment to test the execution time of the sorting algorithms, using the sorted data shown in the table. One of them used a quicksort algorithm, while the other used a bubble sort. The one who used the bubble sort discovered his execution time was shorter as compared to the one who used quicksort.

(d) Comment on this result. [2]

3 The nodes of a binary search tree holding integers in ascending order can be stored in the elements of an array, `bst_arr`.

Each element of the array `bst_arr` comprises three parts: a left pointer, the data and a right pointer.

L_ptr	**Data**	**R_ptr**

The pointers contain the array index of a node either to the left or right of the current node. A value of −1 indicates there are no further nodes in a particular direction.

An integer variable, `Root_ptr`, holds the index of the root node.

The array `bst_arr` is shown:

Root_ptr
−1

Index	**L_ptr**	**Data**	**R_ptr**
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

(a) A binary search tree is constructed with the values

```
99, 97, 95, 125, 121, 109, 135
```

inserted in this given order.

(i) Copy and complete `bst_arr` to show how the nodes are stored. [4]

A procedure `P` is written to display the values of the binary search tree in the same order that they were inserted.

```
PROCEDURE P(Index : INTEGER)
    IF bst_arr[Index].L_ptr <> -1
      THEN
        P(bst_arr[Index].L_ptr)
    ENDIF
    OUTPUT(bst_arr[Index].Data)
    IF bst_arr[Index].R_ptr <> -1
      THEN
        P(bst_arr[Index].R_ptr)
    ENDIF
ENDPROCEDURE
```

(ii) By identifying the type of tree traversal performed by procedure `P`, explain why the procedure does not give the desired sequence of values. [2]

(iii) Rewrite procedure `P` so that it will give the desired sequence of values. [2]

The values 95 and 99 are deleted from `bst_arr` in this given order.

(iv) Rewrite `bst_arr` from **(i)** to show how the nodes are now stored after the values are deleted. [4]

A queue `Queue` and a stack `Stack` are implemented using the data structures and variables given in the table below:

Identifier	**Data Type**	**Description**
`Stack`	`ARRAY [1:15] OF INTEGER`	Array used for containing data stored in Stack.
`StackPtr`	`INTEGER`	Position of array where data item currently at the top of Stack is stored.
`Queue`	`ARRAY [1:15] OF INTEGER`	Array used for containing data stored in Queue.
`QueueFront`	`INTEGER`	Position of array where data item at the front of Queue is stored.
`QueueLast`	`INTEGER`	Position of array where data item at the end of Queue is stored.

Both `Queue` and `Stack` are initially empty.

(b) The values

`99, 97, 95, 125, 121, 109, 135`

are enqueued into `Queue` in this given order.

(i) State the values of `QueueFront` and `QueueLast` after the values are enqueued. [2]

Each value in `Queue` is now dequeued and pushed immediately onto `Stack`. The value of `StackPtr` starts from 1 as items are pushed.

The following table represents the array used for implementing `Stack`.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value															

(ii) Copy and complete the table to show its state after all the values from `Queue` have been moved to `Stack`. [1]

Each value in `Stack` is now popped and enqueued immediately back to the same `Queue`.

(iii) Comment on the state of the values in `Queue` now as compared to its state in **(i)**. [1]

Both `Queue` and `Stack` have been implemented as static data structures.

(iv) Suggest how `Queue` and `Stack` could be implemented as dynamic data structures.[1]

(v) From the perspective of memory management, give **one** advantage and **one** disadvantage of implementing `Queue` and `Stack` as dynamic data structures. [2]

4 You are a lead programmer in a software solutions company. Your team is tasked to create a booking module for a Singapore-based event booking portal.

(a) When designing a software module that manages data inputs, it is important to implement data validation and verification.

Explain the difference between data validation and verification. [2]

(b) All bookings require the user to enter their mobile number and email address for contact purposes.

(i) For each of the following booking information required by the client, describe an appropriate validation check. You must describe a different validation check for each booking information:

1. Mobile number [1]
2. Email address [1]

(ii) For each validation check in part **(b)(i)**, suggest a suitable test value, and explain why you chose the test value. [4]

The payment sub-module of the booking module requires storing 16-digit credit card numbers. The credit card number is validated using the Luhn formula, also known as Mod 10 algorithm:

1. Starting from the right, the first digit is location number 1.
2. Double all digits in the even-numbered positions.
3. Sum all the digits, including both the unchanged digits (i.e. those in the odd-numbered positions) as well as those doubled (e.g. 18 contributes 1 + 8).
4. If the sum is divisible by 10, the credit card number is valid. Otherwise, it is invalid.

For example, given the 16-digit number 4417 1234 5678 9113:

Step 1:	4	4	1	7	1	2	3	4	5	6	7	8	9	1	1	3
Step 2:	8	4	2	7	2	2	6	4	10	6	14	8	18	1	2	3

Step 3: 8 + 4 + 2 + 7 + 2 + 2 + 6 + 4 + (1+0) + 6 + (1+4) + 8 + (1+8) + 1 + 2 + 3 = 70

Step 4: The credit card number 4417 1234 5678 9113 passes the Luhn test since 70 is divisible by 10.

(c) Copy and complete the following pseudocode for the Luhn formula. [6]

```
FUNCTION isValid(CreditCard: STRING) RETURNS BOOLEAN
    // Complete the function
ENDFUNCTION

OUTPUT "Credit card valid? ", isValid("4417123456789113")
```

5 MOE launched the Personalised Digital Learning Programme for all secondary school students as part of the National Digital Literacy Programme in March 2020. Under this programme, all students will be issued personal learning devices for use on school networks.

(a) Describe **two** measures to improve the security of the school network and student learning devices. [4]

(b) Explain how each of the following might be used to compromise student learning devices:

(i) Trojan

(ii) Virus [4]

6 A technology company decides to implement a payments application using object-oriented programming (OOP).

Each employee will have the following data recorded:

- a unique employee ID
- the employee’s name
- the employee’s address
- the employee’ date of birth

There are two types of employees: Salaried and Apprenticeship.

Salaried employees are paid at the end of the month, at a base monthly rate. If the employee has worked more than 160 hours in a month, they will receive a bonus payment of $500. If the salaried employee pays into a pension, the company deducts 4% from the employee’s salary for the pension payment.

Apprenticeship employees are paid weekly, at an hourly rate. Apprenticeship employees do not receive bonus payments or pay into a pension.

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

- the superclass
- any subclasses
- inheritance
- properties
- appropriate methods. [12]

(b) Explain **inheritance** using examples from this situation. [2]

A student describes an example of a principle of object-oriented programming (OOP). She says:

“One method exists in the parent class but is overridden in the child class, to behave differently.”

(c) Identify the principle the student has described. [1]

BLANK PAGE

7 A database containing bus route information contains information about bus routes and bus stops. The Route table contains information about the stops on a route:

- ServiceNo – The bus service of the route
- Operator – The company operating the route
- Category – The type of route
- Direction – The direction of the route (1 or 2 only)
- StopSequence – The numerical sequence of the stop (First stop = 1, second stop = 2, ...)
- BusStopCode – The unique code of the bus stop

The Stop table contains information about each bus stop:

- RoadName – The road that the bus stop is on
- Description – A short description of the bus stop location
- BusStopCode – The unique code of the bus stop

Route

ServiceNo	**Operator**	**Category**	**Direction**	**StopSequence**	**BusStopCode**
10	SBST	TRUNK	1	1	75009
10	SBST	TRUNK	1	2	76059
10	SBST	TRUNK	1	3	76069
. . .	. . .	. . .	. . .	. . .	. . .
10	SBST	TRUNK	2	1	16009
10	SBST	TRUNK	2	2	16089
. . .	. . .	. . .	. . .	. . .	. . .
118	GAS	TRUNK	1	1	65009
. . .	. . .	. . .	. . .	. . .	. . .

Stop

RoadName	**Description**	**BusStopCode**
Tampines Ctrl 1	Tampines Int	75009
Tampines Concourse	Tampines Concourse Int	75019
Tampines Street 86	BLK 879B	75021
Tampines Ave 8	Opp Springfield Sec Sch	75031
Tampines Ave 8	Springfield Sec Sch	75039
. . .	. . .	. . .

(a) Explain why the table **Route** is not in third normal form (3NF). [2]

(b) State the primary key for the table **Stop**. [1]

(c) Draw an entity-relationship (ER) diagram showing the necessary tables and the relationships between them, such that all tables are in third normal form (3NF). [3]

(d) 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.

Write table descriptions for two tables to hold the data from the **Route** table each of which are in third normal form (3NF). [4]

(e) Explain why the use of composite keys is necessary in representing the bus route information. [2]

(f) Write an SQL query to output the number of bus services of each bus operator along `"Tampines Ave 8"`. [6]

A company is making a web application that allows users to look up bus route information for travel between any two locations in Singapore.

(g) State two relevant usability principles for this web application, and describe how they can be applied to the design of the web application. [4]

-- END OF PAPER --