NATIONAL JUNIOR COLLEGE
Mathematics Department

General Certificate of Education Advanced Level
Higher 2

COMPUTING

9569/01

Paper 1 Written

15 Sep 2025

3 hours

Additional Materials: Pre-printed A4 Answer Booklet

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.

There are 7 questions totalling 100 marks.

Answer **all** questions.
Approved calculators are allowed.

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

This document consists of 13 printed pages and 3 blank pages.

1 A Super Fibonacci Sequence (SFS) is a sequence of integers with the property that, from the third term onwards, every term is the **sum of all previous terms**.

For example, in the sequence `1,2,3,6,12,... :`

- The third term, `3`, is `1 + 2`
- The fourth term, `6`, is `1 + 2 + 3`
- The fifth term, `12`, is `1 + 2 + 3 + 6`

We assume that the first term is always 1. However, we are free to choose the second term. By setting different values for the second term, we generate different sequences.

For example:

- If the second term is `5,` the sequence is: `1,5,6,12,24,48,96,...`
- If the second term is `6,` the sequence is: `1,6,7,14,28,56,112,...`

a) A **recursive** function, `super_fib_R(term_2, n)` is to be written in pseudocode.

This function will:

- compute the `nth` term of the Super Fibonacci Sequence with the given second term, `term_2`. You can assume that `n > 0` and `term_2 > 0`.
- returns the `nth` term.

For example, `super_fib_R(5, 5)` should return `24`, since the fifth term of the sequence starting with `1,5,6,12,24, …` is `24`.

An incomplete algorithm for the function is given below:

```
01  FUNCTION super_fib_R(term_2:INTEGER, n:INTEGER)
    RETURNS  INTEGER
02    IF n = 1 THEN
03    :
 :    :
 :    :
    ENDFUNCTION
```

Complete the **recursive** algorithm in pseudocode with the following requirements:

- You **may not** use any iterative constructs (such as `FOR` or `WHILE` loops); the entire solution must rely solely on recursion.

- You **may not** make use of any external function (e.g. `SUM)` or any helper functions.
- You **may not** use any abstract data types (such as array or list).
- You are **not allowed** to change the function signature.

[4]

b) Write an algorithm in pseudocode for an **iterative function** named `super_fib_I(term_2, upper)` that prints all the terms of a Super Fibonacci Sequence (SFS) up to a given upper limit.

- The first term is always 1.
- The second term is `term_2`, provided as an input parameter, is greater than 0.
- From the third term onwards, each term is the sum of all previous terms in the sequence.
- You can assume that the upper limit, `upper`, provided as an input parameter, is greater or equal to `term_2`.
- The function should print each term in the sequence, if it is **less than or equal** to the given upper limit specified as the input parameter, `upper`.
- You must use an **iterative** construct (such as a `WHILE` loop) to generate the sequence.

For example, `super_fib_I(3,5)` should print `1,3,4`. [4]

c) Using the variables from `super_fib_I` in 1b), draw a trace table to trace the execution of `super_fib_I(3, 10)`. Include all variable states and printed output. For example, if the variables `a` and `b` are used, the trace table should at minimum look like this:

a	b	output

[3]

d) Test cases are used to validate the correctness and robustness of your code. Based on the algorithm you have written for the function `super_fib_I` in 1b). Copy and complete the table below by filling out the appropriate test cases.

Type of Test Case	Test Case	Expected Output	Reason of Test
Normal	`super_fib_I(2,7)`	`1,2,3,6`	Inputs should produce correct expected output
Normal			Inputs should produce correct expected output
Boundary			
Abnormal			

[4]

2 A engineer is writing code for a network server program that periodically receives data updates from Internet of Things (IoT) devices. The server is implemented in a programming language that supports dynamic memory allocation. The following algorithm describes what is to happen when the server recieves a signal indicating that an IoT device is about to transmit data.

```
1   allocate 64K bytes of memory to current buffer
2   receive byte stream and store into current buffer
3   IF sender terminates connection before current buffer is full THEN
4     decode data in current buffer and write to a text file
5     release the memory used by the current buffer
6     block and wait for the next transmit signal from IoT device
7   ELSE // buffer is full but there is still incoming data
8      decode data in current buffer and write to a text file
9      allocate another 64K bytes of memory for new buffer
10     assign the new buffer as the current buffer
11     GOTO line 2
12  ENDIF
```

The engineer tested the code by using a simulator to send data to the server program and found it to work correctly. The server is then deployed to run in a production environment. After several days of continuous operation, the server program crashes due to a runtime error. Upon rebooting the server, the program resumes normal operation. Without any further investigation, the engineer claims that it was a intermittent network problem and requests the user to just re-boot the server when it happens again.

(a) Discuss whether the engineer is abiding by the code of ethics of a computer professional. [2]

(b) Explain how a mistake in the algorithm is causing the problem and how to fix it. [3]

(c) Suggest two reasons why the mistake was not identified during testing. [2]

3 A company wants to motivate its staff to be punctual, limit unapproved absences, and reward extra effort while ensuring discipline when necessary.

At the end of each quarter (3 months), the HR department reviews each employee's attendance record. The review process works as follows:

- An employee's punctuality score is assessed — if the employee arrived late to work on more than 3 separate days in a quarter, it is flagged as a punctuality concern.
- The absence policy allows for up to 2 unapproved leave days in a quarter. If an employee takes more than that without valid reasons, it is flagged as an absence violation.
- The company encourages overtime, so if an employee logs any approved overtime hours in a quarter, it is noted for potential reward - but only if they have maintained satisfactory punctuality and leave records.
- Based on the quarterly assessment:
  - Any punctuality concern or absence violation will result in a salary deduction.
  - If an employee has both a punctuality concern and an absence violation in the same quarter, they will also receive a formal warning letter for poor attendance discipline.
  - If an employee shows no punctuality concerns and no absence violations, but has logged overtime, they will receive a performance bonus for their extra effort.
  - If an employee has any punctuality concerns or absence violations, they are not eligible for the performance bonus, even if they worked overtime.

a) Complete the decision table below showing all the possible conditions and their respective actions/outcomes.

PunctualityConcern								
AbsenceViolation								
OvertimeClocked								
DeductSalary								
IssueWarning								
ReceiveBonus								

[4]

b) Simplify your decision table by removing redundancies. [4]

c) Based on the simplified decision table in 3b), complete the pseudocode below to perform a quarterly assessment of an employee. The outcome of the assessment will be one or more of the of the following outputs:

- "Deduct Salary"
- "Issue Warning"
- "Receive Bonus"
- "None"

Use the following variable names in your pseudocode:

Variable Name	Value
punct	True if employee has been flagged with a Punctuality Concern, False otherwise
abs	True if employee has been flagged with an Absence Violation, False otherwise
over_t	True if employee has logged overtime ours, False otherwise

```
FUNCTION Quarterly_Assessment(punct, abs, over_t)
  :
  :
ENDFUNCTION
```

[4]

4 Silver Screen is a company that operates a cinema offering continuous movie screenings 24 hours a day. Each movie may be scheduled for screenings on various dates, with multiple showtimes available on each date. The cinema currently uses a computer system which requires their patrons to visit the cinema's booking office in person to purchase tickets – either for immediate viewing or to book in advance. The computer system used at the booking office relies on a database table that allows patrons to view the seat availability for specific movie screenings before processing ticket purchases. A sample of the records in the database table is shown below:

movie name	movie genre	duration	date	time	price	row	number	type	status
Bleach	Animation	100	2025-08-01	1900	10	A	5	Normal	Booked
Bleach	Animation	100	2025-08-01	1900	15	B	1	Premium	Available
Bleach	Animation	100	2025-08-01	1900	20	C	2	Couple	Booked
Dune	Sci-Fi	165	2025-08-01	2130	10	A	3	Normal	Available
Dune	Sci-Fi	165	2025-08-01	2130	15	B	2	Premium	Booked
Dune	Sci-Fi	165	2025-08-01	2130	20	C	1	Couple	Available

Each row in the table represents a seat for a specific movie screening and includes the movie title, genre, duration, screening date and time, seat row and number, seat type, ticket price, and booking status.

(a) Identify the attribute/s that can be used as the primary key for the table. [1]

(b) Identify the highest normal form this table satisfies (UNF, 1NF, 2NF, or 3NF) and briefly explain your answer. [3]

(c) Based on the table above, write the SQL query that retrieves all the available seats for the movie `Dune` to be screen on `2025-08-02` at `1400`. [2]

(d) Give an example of an insert, update, or delete anomaly that could occur with the table above. Explain why it is considered an anomaly. [2]

Perform normalisation on the table above up to Third Normal Form (3NF). Your database design should not only reduce redundancy but also support efficient execution of database operations.

(e) Draw an entity-relationship diagram (ERD) to describe the additional tables and their relationship in the normalised database. [4]

(f) Write the table descriptions for the tables identified in **(e)**. [4]

(g) Based on the normalised database, describe the sequence of database operations that must be performed in the following scenarios:

i. When a new movie becomes available for screening.

ii. When a customer purchases a ticket for a movie screening.

You do not need to provide the actual SQL statements. [4]

The existing system does not allow Silver Screen's customers to purchase tickets online or enable the company to keep track of customer information. The database model needs to be re-design to allow Silver Screen to:

- capture and store their customers' personal information including name, phone contact and email address.
- keep track of all the movie tickets that a customer has purchased.
- Implement a three-tier customer reward programme that offers discounts and privileges based on the customer's reward tier:
  - Gold Tier: Customers enjoy a 15% discount along with additional privileges.
  - Silver Tier: Customers receive a 10% discount.
  - Standard Tier: Customers do not receive a discount but can still participate in the basic reward programme.

(h) Create a new ERD diagram to reflect the new database design. [4]

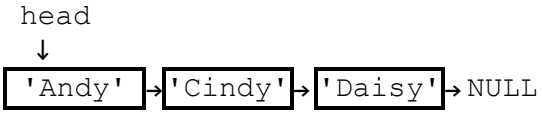
(i) Write the table descriptions for the additional or modified tables. [4]

(j) What measures must Silver Screen take to ensure that the collection, use, and storage of customer information comply with the requirements of the Personal Data Protection Act (PDPA)? [3]

(k) Silver Screen is deciding between developing a native application or a web application to allow customers to book tickets online via the Internet.

State one advantage and one disadvantage to the company of choosing a native application over a web application [2]

5 A linked list can be represented as a number of nodes each containing an item of data and a pointer to the next node. A pointer `head` indicates the location of the first node. A value of `NULL` in the `head` represent an empty list; a value of `NULL` in a pointer indicates the last node. The logical structure of a linked list looks like this:



The linked list is implemented using a variable, `head`, and two 1-dimensional arrays, `data` and `pointer` as shown below:

head: 2

index	**data**	**pointer**
0	'Cindy'	5
1	NULL	NULL
2	'Andy'	0
3	NULL	NULL
4	NULL	NULL
5	'Daisy'	NULL

The pseudocode for initialising the arrays and variable is as follows:

```
DECLARE data: ARRAY[0:5] OF STRING
DECLARE pointer: ARRAY[0:5] OF INTEGER
DECLARE head: INTEGER
FOR i = 0 TO 5
  data[i] = NULL
  pointer[i] = NULL
ENDFOR
head = NULL
```

The helper function `get_free_index()`, which returns the index of a free space in the array, is provided below:

```
FUNCTION get_free_index()
  FOR i = 0 to 5
    IF data[i] == NULL THEN
      RETURN i
    ENDIF
  ENDFOR
  RETURN NULL // No empty slots in array
ENDFUNCTION
```

(a) Complete the pseudocode below to insert a new item `"Bob"` into the linked list in lexicographical order. You can assume that the linked list is not empty.

```
index = get_free_index()// assume that there is space
data[index] = "Bob"
 :
 :
 :
```

[7]

(b) Update the diagram below to reflect the changes after 'Bob' is inserted into the linked list.

head: 2

index	**data**	**pointer**
0	'Cindy'	5
1	NULL	NULL
2	'Andy'	0
3	NULL	NULL
4	NULL	NULL
5	'Daisy'	NULL

[2]

(c) After 'Bob' is inserted, 'Cindy' is removed. Update the diagram to reflect these changes. [2]

(d) After an item is removed from the linked list, its memory space can be made available for re-use. Explain the necessary steps to mark this space as free so that the `get_free_index()` helper function described above can identify and utilise this space for subsequent insertions. [2]

(e) Instead of using the `get_free_index()` helper function described above to find an index to insert new item, propose a more efficient mechanism to keep track of the free spaces in the arrays for inserting new items. Explain why your proposed mechanism is more efficient in terms of time complexity. [3]

6. The global Internet is a network that uses packet switching and is unified by the TCP/IP protocol stack.

(a) Explain the term packet-switching. [2]

(b) Explain what a protocol means in the context of the Internet and describe how protocols provide unification across different systems and networks. [2]

(c) Explain how the packet-switching architecture of the Internet contributes to network resilience. [2]

(d) The Internet can be described as a **network of networks**. Explain what this means, using examples if necessary. [2]

(e) Explain what the Domain Name System (DNS) is and describe how it facilitates user access to websites on the Internet. [2]

(f) When a user accesses a website using a web browser, state two cryptographic techniques that can be used to ensure confidentiality of the transmitted data. [2]

7 A hash table is implemented using a fixed-size array, `table`, with 100 slots, capable of storing up to 100 objects. The array is initialised with `NULL` values. It uses linear probing as the collision resolution technique. A hash function `hashie(k)` is provided, which returns the index at which the object with key `k` should be placed. The hash table allows insertion and deletion of objects. You can assume that keys are always unique in the hash table.

Complete the pseudocode in the function below to search the hash table for an object with a given key `k`. The function should return `True` if found, or `False` if the key does not exist in the table.

```
FUNCTION search(k)
  index = hashie(k)
  IF table[index] IS NOT NULL and table[index].key == k
    //object.key gives the key of the object
    RETURN True
  :
  :
  :
  :
ENDFUNCTION
```

[5]

END OF PAPER