VICTORIA JUNIOR COLLEGE
JC 2 PRELIMINARY EXAMINATION
Higher 2

COMPUTING [MARKING SCHEME] 9569/01

Paper 1 Written **17 September 2025**

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.

1 A programmer is writing a program to manage and search for records. The records include phone numbers, which comprise 8 decimal digits.

(a) Convert the phone number `62842281` to:

(i) Hexadecimal representation [2]

62842281 = 3*16^6 + 11*16^5 + 14*16^4 + 14*16^3 + 5*16^2 + 10*16^1 + 9*16^0 [1]
Or repeated division by 16

3BEE5A9 [1]

(ii) ASCII value for each digit.
(`'0'` has an ASCII value of 48 and `'9'` has an ASCII value of 57.) [2]

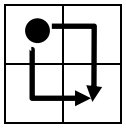
Map each ASCII decimal character to ASCII value
54, 50, 56, 52, 50, 50, 56, 49 [1 mark for every 4 correct values]

(b) The programmer's supervisor suggests a binary search tree instead of a sorted array for managing phone numbers.

Suggest **two** reasons for the supervisor's advice. [2]

BST has lower time complexity for adding items [1]
BST maintains items in sorted order, whereas array may require re-sorting [1]

2 A recursive function `countPaths` is designed to find the number of different possible paths from the top-left corner to the bottom-right corner of a grid. The function can only move right or down at each step.



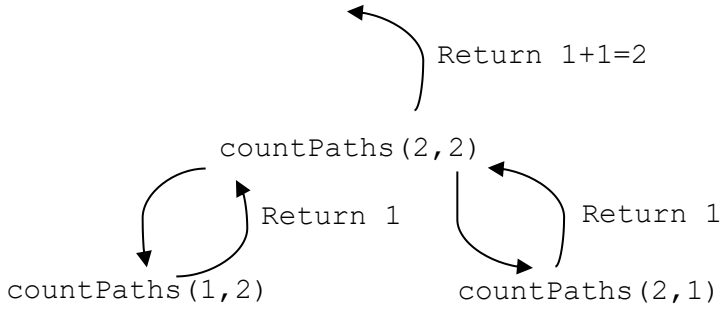
For example, in a 2×2 grid, where `row = 2` and `col = 2`, there are two possible paths from the top-left corner to the bottom-right corner of the grid:

- Right → Down
- Down → Right

The function is defined as follows:

```
01 FUNCTION countPaths(row : INTEGER, col : INTEGER) RETURNS INTEGER
02   IF row = 1 OR col = 1 THEN
03     RETURN 1
04   ENDIF
05   RETURN countPaths(row - 1, col) + countPaths(row, col - 1)
06 ENDFUNCTION
```

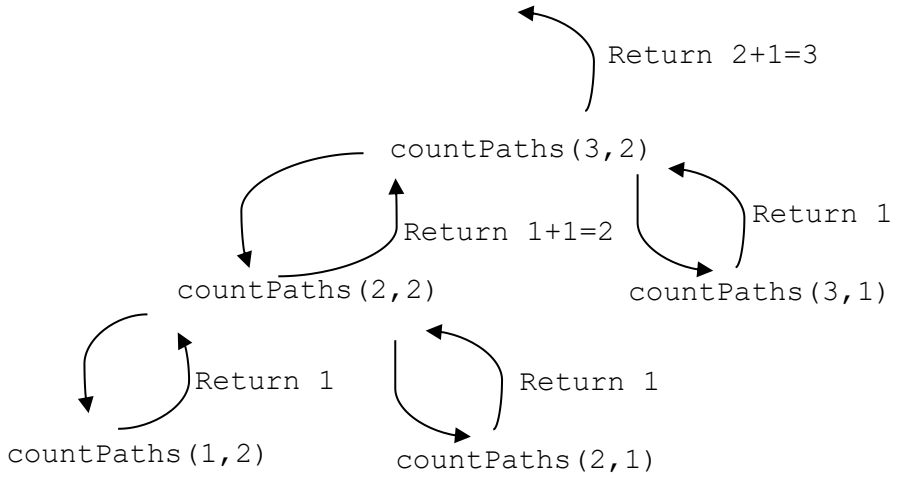
(a) An example of a trace tree diagram showing `countPaths(2,2)` is shown as follows:



Use the above example to create a trace tree diagram for the recursive function call `countPaths(3,2)`. [4]

Correct root node showing countPaths(3,2) [1] Correct first level of recursive calls [1] Correct first level of return values [1] rect final return calculation and value (2+1=3) [1]

Note: Second level of recursive calls given in example



(b) Explain how the base case in lines 02 to 03 prevents infinite recursion. [2]

Explanation that base case is reached when either dimension becomes 1 [1]
Explanation that this ensures termination by returning a value [1]

When the function is called, it checks if either row or col is equal to 1. If this condition is met, the function returns 1, indicating that there is exactly one way to reach the destination when either dimension is reduced to 1. This means that the function will no longer call itself recursively, effectively terminating the recursion when it reaches this base case, thus preventing the function from running indefinitely

(c) Write an iterative version of the original `countPaths` function. [5]

Correct array/table initialization [1]
Proper handling of edge cases [1]
Correct nested loop structure [1]
Correct cell value calculation [1]
Return correct final value [1]

```
FUNCTION countPathsIterative(row : INTEGER, col : INTEGER) RETURNS
INTEGER

    DECLARE grid : ARRAY[1:row, 1:col] OF INTEGER

    // Initialize the first row and first column
    FOR i ← 1 TO row DO
        grid[i,1] ← 1
  NEXT i

    FOR j ← 1 TO col DO
        grid[1,j] ← 1
    NEXT j

    // Fill the rest of the paths matrix
    FOR i ← 2 TO row DO
        FOR j ← 2 TO col DO
            grid[i,j] ← grid[i – 1,j] + grid[i,j – 1]
        NEXT j
    NEXT i

  RETURN gird[row,col]

ENDFUNCTION
```

3 A software company is developing a print server that handles print jobs from multiple users. Each print job has a job ID and number of pages. The system needs to handle both regular and priority print jobs.

(a) The regular print jobs use a linear queue data structure.

Explain why a linear queue would be appropriate for regular print jobs. [2]

Definition of queue as FIFO structure [1]
Explanation of appropriateness for print jobs (order preservation) [1]

A queue is a First-In-First-Out (FIFO) data structure where elements are added at one end (rear) and removed from the other end (front).
This is appropriate for print jobs as it ensures jobs are processed in the order they were submitted (effectively manage multiple print jobs without the risk of mixing them up), maintaining fairness in the printing system.

(b) The company decides to implement the queue as a circular queue using a static array of size 5.

State **one** difference between a linear queue and a circular queue. [2]

- Once a linear queue is full, no more print jobs can be added until all the queued jobs have been processed. [1]
- With a circular queue, print jobs can be added in positions from which the print job has been processed. [1]

A linear queue stores print jobs sequentially and can become inefficient when it reaches capacity, as it cannot accept new print job until all the print jobs have been processed, potentially wasting space. In contrast, a circular queue wraps around and efficiently utilizes space by allowing a new print job to be added at the front if there is space due to dequeued job, thereby preventing wasted positions.

(c) Draw a diagram showing the state of a circular queue after the following operations, showing the positions of front and rear pointers:

1. Enqueue jobs: 101, 102, 103, 104
2. Dequeue two jobs
3. Enqueue jobs: 105, 106 [3]

Correct element placement [1]
Correct front pointer position [1]
Correct rear pointer position [1]

[106] [ ] [103] [104] [105]
^Front (at index 2)

[106] [ ] [103] [104] [105]
^Rear (at index 0 or index 1 if rear is pointing to next available slot)

(d) Explain **one** disadvantage of using a circular queue instead of a linear queue for this implementation. [2]

Valid disadvantage [1]
Sufficient explanation in the context [1]

A circular queue has a more complex implementation logic that requires additional calculations to handle the wraparound of the rear pointer and to determine when the queue is full.
For example, when adding a new print job, the program needs to check if (rear + 1) mod size equals front to determine if the queue is full, rather than simply checking if rear has reached the end of the array as in a linear queue. This increased complexity makes the code more difficult to maintain and debug if issues arise with the print job system.

(e) The print server keeps a history of completed print jobs in a stack data structure.

(i) Explain how a stack data structure would be appropriate for implementing an "undo" feature that allows the system administrator to restore recently completed print jobs back to the queue. [2]

Explanation of stack's LIFO nature being suitable for undo/history [1]
Clear connection to how this benefits system administrator's restore operations [1]

A stack is appropriate for implementing an undo feature because its Last-In-First-Out (LIFO) structure naturally matches the way undo operations work. When a print job is completed, it is pushed onto the stack, allowing the most recently completed job to be easily accessed and restored first. This means that if the system administrator decides to undo the last completed job, they can simply pop it from the stack and restore it back to the print queue. This matches the typical user expectation that undo operations will reverse the most recent actions.

(ii) Write pseudocode for a procedure `restoreJobs` that takes three parameters: the stack object, `historyStack`, the queue object, `printQueue` and an integer `N`. The pseudocode should restore the `N` most recently completed jobs from the `historyStack` back to the `printQueue`, maintaining their original print order. [6]

Correct procedure header with parameters [1]
Appropriate use of temporary storage to maintain order [1]
Correct loop to handle N items [1]
Proper stack operation (pop from history) [1]
Proper stack operation (push to temp) [1]
Proper queue operations to preserve original print order (pop from temp and enqueue to queue) [1]

```
PROCEDURE restoreJobs(historyStack : STACK, printQueue : QUEUE, N :
INTEGER)
    // Declare temporary storage to maintain order
    DECLARE tempStack : STACK
    DECLARE count : INTEGER

    count ← 0

    // Check if N is valid
    IF N <= 0 THEN
        RETURN
    ENDIF

    // Move up to N jobs to temporary stack to reverse order
    WHILE NOT historyStack.isEmpty() AND count < N
        tempStack.push(historyStack.pop())
        count ← count + 1
    ENDWHILE

    // Move jobs from temporary stack to queue to restore original
order
    WHILE NOT tempStack.isEmpty()
        printQueue.enqueue(tempStack.pop())
    ENDWHILE
ENDPROCEDURE
```

4 Most applications require data entry as an essential step for processing and output.

(a) During data entry, validation and verification are commonly applied.

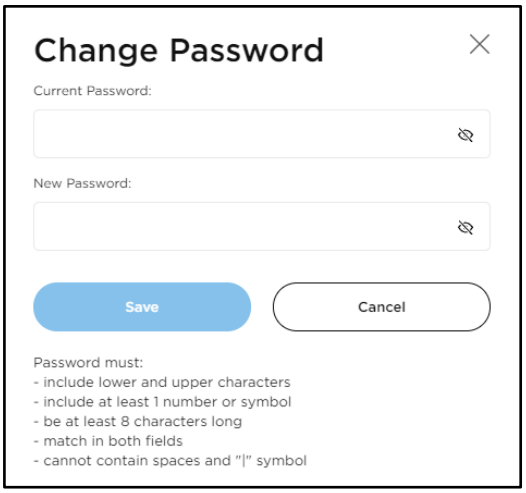
(i) State the purpose of validation. [1]

ensure that input data meets a set of requirements
*Reject "check"

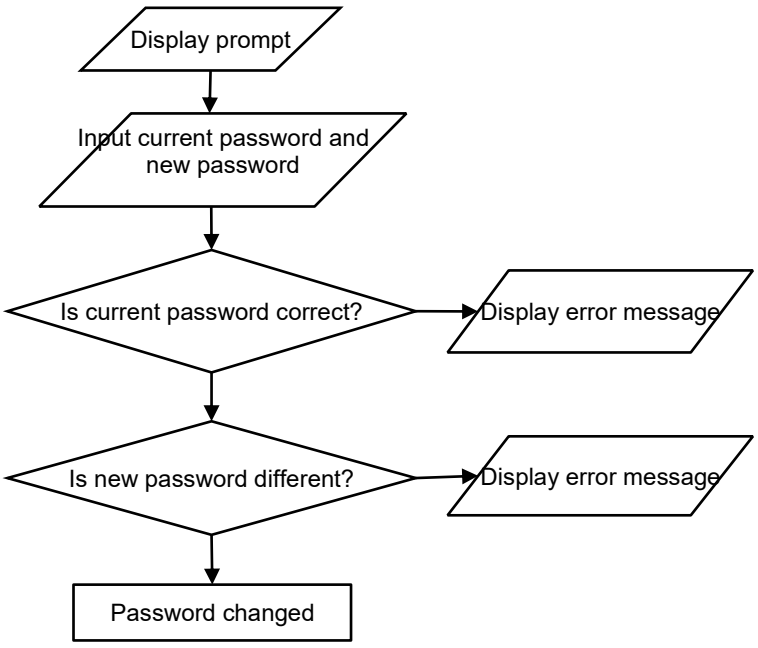
(ii) State the purpose of verification. [1]

ensure that data entered is what the user intended to enter
*Reject "check"

The image below shows the password change dialog box for a web application.



The flowchart below shows the process for changing a user's password.



Yes

Yes

No

No

(b) Redraw the flowchart to include, for the new password:

- **one** suitable validation technique
- **one** suitable verification technique [6]

appropriate symbol for validation, verification: decision diamond [1] appropriate validation check: format [1] OR appropriate validation check: length appropriate verification technique: double entry [1] "Yes" -> next step, [1] "No" -> error message [1] validation check is carried out before verification check [1]

(c) State **two** ways the password change interface can be improved using suitable usability principles. [2]

Help users recognise, diagnose, and recover from errors: the reason for invalid password should be displayed below the appropriate textbox [1] **Error Prevention:** disable the Save button (grayed out) until new password passes all checks [1] Other possible answers: consistency and standards and visibility of system status etc

5 A company, ImageMaps, wishes to develop an application that scans a user's device for photographs, detect the associated location, and display the images on a map using the associated locations.

(a) State the actions ImageMaps must take to secure critical business data such as user data against catastrophic hardware failure. [4]

Backups of critical data must be made [1]
Backups should be made regularly [1]
Multiple copies should be made and stored offsite / in different locations, to minimise risk of complete data loss [1]
The backups should be tested to ensure that data recovery can succeed [1]

Any other sensible answers related to backup.

(b) State the consequences, should ImageMaps suffer such a hardware failure. [2]

Catastrophic failure resulting in data loss can mean loss of business continuity (business is unable to operate), loss of revenue, loss of customer / stakeholder trust, etc [1]
Failure to protect critical data, especially financial records or customer personal data, may be a violation of data protection laws and other legal requirements [1]

(c) ImageMaps needs to store a large quantity of data that will not be frequently accessed. State how they can reduce costs while doing so. [2]

Data not frequently accessed can be archived [1]
by moving it to cheaper storage [1]

(d) State **two** actions the company must take regarding the collection and use of personal user data to comply with Singapore's Personal Data Protection Act (PDPA). [4]

Proper explanation of PDPA obligation [1]
Description of a concrete action to fulfill the obligation [1]

Notification: ImageMaps must inform customers of the purposes of data use through a privacy policy
Consent: ImageMaps must obtain customers' informed consent (e.g. through app permissions or webapp dialog box)
Purpose Limitation: ImageMaps must ensure employees do not use customer personal data for purposes other than the stated (e.g. cannot be used to train AI)
Access and Correction: ImageMaps must allow customers to see their stored personal data, and correct it if necessary.

(other appropriate obligations relating to collection and use, with appropriate action specified)

(e) State whether a native or web application is more suitable for ImageMaps's application. Explain your answer. [3]

Native application [1]
The application needs to access device storage [1]
Web apps do not have the same capabilities as native apps, such as the capability to access device storage [1]

Native app is more suitable due to deeper OS integration, better performance for image processing, and more reliable file system access across all devices.

6 A commercial bank is strengthening its cybersecurity framework after experiencing multiple security breaches.

(a) Compare between a Trojan horse and a worm in terms of how they spread and how they are executed in the targeted network or device. [2]

Trojan: Masquerades as legitimate software | Worm: Self-replicating [1] Trojan: requires user execution | Worm: spreads autonomously through networks [1]

(b) Describe **two** characteristics that make ransomware particularly dangerous to financial institutions. [2]

(Any 2 points, 1 mark each) Encrypts critical financial data [1] Causes operational disruption during attacks [1] High ransom demands target valuable data [1]

(c) The bank has deployed a firewall and Intruder Protection System (IPS).

Explain why implementing both a firewall and IPS provides better security than using either system alone. [2]

(Any 2 points, 1 mark each) Firewall filters basic malicious traffic [1] IPS detects sophisticated attacks in payloads [1] Defense-in-depth approach [1]

(d) Describe the complete process of generating and verifying a digital signature for an electronic funds transfer request. [3]

Hash document [1] Encrypt hash with sender's private key [1] Recipient verifies with public key [1]

(e) The bank must choose between biometric authentication and security tokens for employee access to sensitive systems.

Evaluate which method would provide better security, giving **two** reasons for your choice.[3]

(Accept either position with valid reasoning, 1 mark per reason) **Biometric Authentication** Preferred Because: Unique physiological traits [1] Cannot be lost/stolen like tokens [1] Harder to replicate than token codes [1] OR **Security Tokens** Preferred Because: Not affected by physical changes [1] Easier to revoke/reissue [1] No biometric data storage risks [1]

7 A packet switching network breaks data into packets, which are sent independently across the network and reassembled at the destination.

(a) Describe how networking protocols address the following problems:

(i) Data packets may arrive out of order and require reassembly [2]

Each packet is tagged with a sequence number [1]
The sequence number is included in the header to allow sequential reassembly [1]

A sequence number is assigned for each packet to ensure correct ordering and eventual reassembly of the original message.

#No need to differentiate between TCP and UDP.

(ii) Data packets may be corrupted in transit. [2]

The packet data is hashed to produce a checksum [1]
The recipient verifies the checksum against its hashed data [1]

OR

The receiver checks the packets as it reassembles [1]
If any are corrupted in transit, they are retransmitted [1]

Each packet has a sequence number. The receiver checks the packets, and if any are corrupted in transit, they are retransmitted; this supports correct reassembly of the original message.

#No need to differentiate between TCP and UDP.

(b) State and explain the time complexity of the following operations:

(i) Adding an item to a sorted array [2]

O(n) [1]
Need to traverse the whole array if the item is to be inserted at the end [1]

(ii) Adding an item to a balanced binary search tree (BST). [2]

O(log n) [1]
At every level, traverse to either left sub tree or right sub tree. If need to traverse to the leaf node, we would have traversed log n of the tree [1]

(c) A data structure is used to hold data packets that are being reassembled.

Describe **one** advantage and **one** disadvantage of each of the following data structures for packet reassembly.

(i) Statically allocated array [2]

Advantage: Any one of the following [1]
- O(1) random read or write.
- Enables incoming data packets to be written in sequence quickly
- Avoids memory fragmentation because memory blocks are contiguous and aligned.
- Enables faster reassembly because memory blocks are contiguous and aligned.

Disadvantage: Any one of the following [1]
- Number of packets must be known beforehand
- Inflexible/fixed size
- Static allocation is used for arrays, which are not easily resized.

*Can accept memory wastage.

(ii) Dynamically allocated BST. [2]

Advantage: Any one of the following [1]
- Number of packets need not be known upfront.
- O(log n) addition of data packets in order
- Can grow as packets arrive
- BST supports insert operations, allowing scalable reassembly for varying packet counts

*Can accept minimise memory wastage.

Disadvantage: Any one of the following [1]
- Packets arriving mostly in sequence might result in worst case performance due to unbalanced tree
- More complex implementation
- Dynamic memory brings overhead (slower and more complex memory management)
- Can cause memory fragmentation.

8 A software company is writing a program for a vehicle hire business. Both cars and vans are available for hire.

For all vehicles, the data that will be stored include:

Vehicle Registration Number (VRN)
Total distance travelled (km)
Date hired
Date of return
Cost per day
Available for hire

For cars, the additional data stored include:

Fuel type (petrol, diesel, electric, hybrid)

For vans, the additional data stored include:

Maximum load (kg)

The odometer in the vehicle displays the total distance the vehicle has travelled since manufacture.

When a vehicle is hired:

- total distance travelled is set to the odometer's value
- date hired is set to the current date
- return date is set to the date the vehicle is expected to be returned
- available for hire is set to FALSE.

When a vehicle is returned:

- hire cost is returned as the cost per day multiplied by the number of days the vehicle was hired
- total distance travelled is set to the odometer value
- date returned is set to the current date
- available for hire is set to TRUE.

The programmer uses a database to store vehicle hire data. Each row in the table represents a vehicle hire record. A vehicle must be hired for a minimum of one day.

Vehicle hire data is stored in the following table:

VRN	TotalDistance	DateHired	DateReturned	CostPerDay	Available	FuelType	MaxLoad

(a) Identify a suitable primary key for the above table. [1]

VRN AND DateHired [1]

(b) Identify suitable SQL data types for the following columns:

(i) `DateHired` [1]

DATE / TEXT [1]

(ii) `Available` [1]

BOOLEAN / INTEGER [1]

(c) State whether the above table is in Third Normal Form (3NF). Explain your answer. [4]

Not in 3NF [1]
3NF require 2NF [1]
non-key attributes CostPerDay, FuelType, MaxLoad depend only on VRN (name at least one appropriate non-key attribute) [1]
and not on the entire PK (partial dependence) [1]

A database consultant rewrites the above table into the following table descriptions:

```
Vehicle (VRN, Type, TotalDistance, CostPerDay, Available)
Hire (VRN, DateHired, DateReturned)
Car (VRN, FuelType)
Van (VRN, MaxLoad)
```

The vehicle type is stored as either `'Car'` or `'Van'`.

(d) Draw an entity-relationship (ER) diagram representing the four tables. [4]

All entities represented: Vehicle, Hire, Car, Van [1]
Vehicle <--1--1--> Car [1]
Vehicle <--1--1--> Van [1]
Vehicle <--1--n--> Hire [1]

(e) Write an SQL query to retrieve from the above tables the VRNs of hybrid cars and their latest date of hire. [6]

```
SELECT Vehicle.VRN, MAX(DateHired) [1]
FROM Vehicle [1]
  INNER JOIN Hire ON Vehicle.VRN = Hire.VRN [1]
  INNER JOIN Car ON Vehicle.VRN = Car.VRN [1]
WHERE Type = 'Car'
  AND FuelType = 'hybrid' [1]
GROUP BY Vehicle.VRN [1]

OR

SELECT Car.VRN, MAX(Hire.DateHired) [1]
FROM Car [1]
INNER JOIN Hire ON Car.VRN = Hire.VRN [2]
WHERE Car.FuelType = 'hybrid' [1]
GROUP BY Car.VRN [1]
```