Network security

1. (a) Give **one** example of multi-factor authentication. [1]

(b) Explain how a Denial of Service (DOS) attack can compromise an internal server and suggest **one** protection scheme to detect a DOS attack. [3]

(c) Give **two** purposes of using digital signature and describe how it works. [8]

Solution:

(a) multi-factor from the three categories:

- Something you know: password
- Something you have: OTP, token, access card, passport/NRIC card
- Something you are: fingerprint, retina, voice, face

(b) DOS attacks the network traffic by **sending massive request** to exhaust server resources and bandwidth, and hence **disables the server from responding to legitimate requests**.

It can be detected by firewall, intrusion detection/protection system.

(c) purpose:

- Authentication: the message was created by the known sender
- Non-repudiation: the sender cannot deny having sent the message
- Integrity: the message is not altered in transit

Process:

- The sender uses a **hash algorithm** to create a **hashed version of the message**
- The sender uses **its private key** to encrypt the hash to the **digital signature**
- Both the message (encrypted or not) and the digital signature are **sent to the receiver**
- The receiver uses the **sender's public key** to decrypt the digital signature back to the sender's version of hash
- The receiver uses the same hash algorithm to **create a new hash** from the received message
- If the two hashes **match**, it means the data is not altered and is sent by the known sender

Recursion

3 Consider the following pseudocode algorithm:

```
01    FUNCTION P(n: INTEGER) RETURNS INTEGER
02        IF n <= 0
03           THEN
04              RETURN 0
05        ELSE
06              m ← 10 * P(n DIV 2) + (n MOD 2)
07              RETURN m
08        ENDIF
09    ENDFUNCTION
```

Note that `X MOD Y` gives the remainder and `X DIV Y` gives the quotient when `X` is divided by `Y`.

(a) Draw a trace diagram for the recursive function call `P(12)` and give the final result returned. [4]

(b) State the purpose of the function `P`. [1]

(c) State **two** key characteristics of a recursive function, and when is it suitable to be used. [3]

(d) Give **one** advantages and **one** disadvantages of using recursive function. [2]

4. (a) State **two** key characteristics of a recursive function, and when is it suitable to be used. [3]

(b) Consider the following pseudocode algorithm:

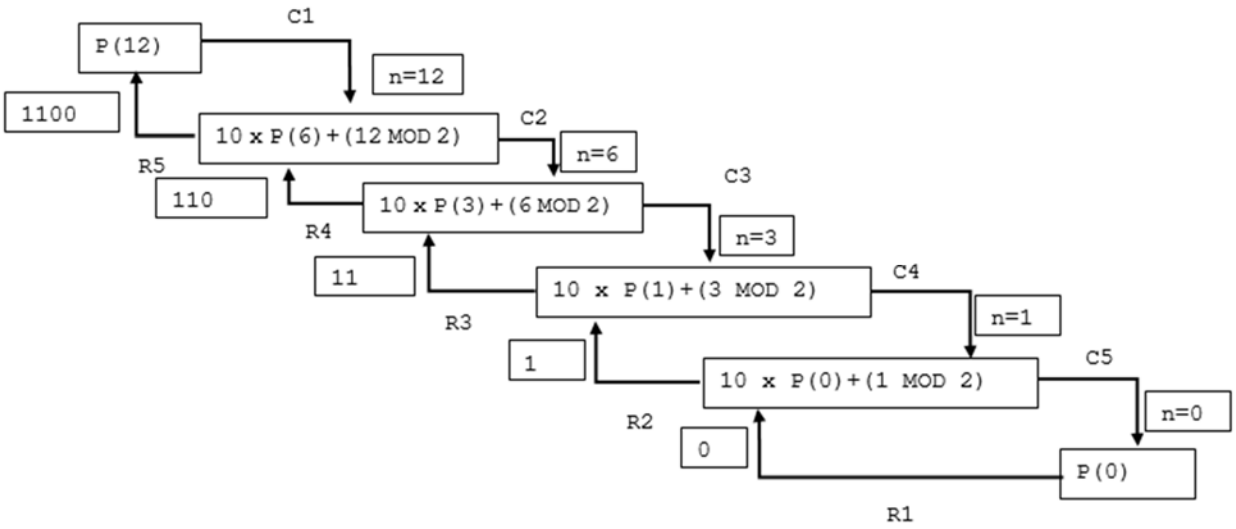
```
01   FUNCTION P(x: INTEGER, n: INTEGER) RETURNS INTEGER
02       IF n = 0
03          THEN
04             RETURN 1
05          ELSE
06             m ← x * P(x, n-1)
07             OUTPUT (x, n, m)
08             RETURN m
09       ENDIF
10   ENDFUNCTION
```

(i) Trace the function call `P(5, 3)`, write down the output in the correct order and the final result returned. [4]

(ii) Describe the purpose of function `P`. [1]

Solution:

3(a)



Final result: P(12) = 1100

(b) Convert the integer n to binary.

(c) A function which contains a call to itself. Should include at least one terminal case – a case that contains no further calls to the recursive subprogram. Used when the original task can be reduced to a simpler version of itself.

(d)

Advantages:

- When solution to a problem is essentially recursive, enables programmer to write a program which mirrors the solution.
- Recursive solutions are often much shorter than non-recursive ones.

Disadvantage:

- If the recursion continues too long, the stack of return addresses may become full (ie no available memory is left) and the program will crash.
- Recursive routines can be difficult to follow and to debug.

Q3 Marker's Comments:

Few students managed to draw the trace diagram correctly. The first call should start with C1, n=12, the last call C5, n=0. Each call should be labeled as *C1, C2… with n=12, n=6* … The "Returns" should be labeled as *R1, R2… with a returned value.*

Some students did not indicate clearly the final result.

On suitability: Used when the original task can be reduced to a simpler version of **itself**.
Some students use the word **problem, code** instead of **itself**.

Recursive function **might not** be more efficient than an iterative one and it **might not** execute faster than an iterative one.

Need to state clearly which sentence is an advantage or disadvantage. Don't let the marker guess as sometimes it is not clear.

Some students mentioned that recursive functions require more resources. You will need to be clearer and explain why more resources are needed to complete the answer.

4.

(a)

- A function which contains a call to itself.
- Should include at least one terminal case – a case that contains no further calls to the recursive subprogram so that it will not continue indefinitely.
- Used when the original task can be reduced to a simpler version of itself

(b) (i)

x	n	m
5	1	5
5	2	25
5	3	125

Return 125

(ii) Calculates x to the power of n

Searching & Sorting

5. The arrays `PollData[1:10]` and `CardData[1:10]` store data.

PollData	12	85	52	57	25	11	33	59	56	91

CardData	11	12	25	52	33	56	57	59	91	85

(a) State why it will take lees time to complete an insertion sort on `CardData` than on `PollData`. [1]

(b) Write an algorithm, in **pseudocode**, that performs an insertion sort on the `CardData` array. [5]

(c) (i) A binary search algorithm is used to find a specific value in an array.

Explain why an array needs to be sorted before a binary search algorithm can be used. [2]

(ii) The current contents of `CardData` are shown.

11	12	25	33	52	56	57	59	85	91

Explain how a binary search will find the value `25` in `CardData`. [4]

6 Following an outbreak of catpox, the National Infectious Disease Center (NCID) keeps a record of all the infection cases. The records are stored in a list sorted by their date and hour of diagnosis by a doctor.

(a) For contact tracing purposes, NCID wants to identify a patient who was diagnosed on a certain date and time. Explain whether a linear search or a binary search would be more suitable. [2]

As the number of infections increase, it was determined that storing the records in a sorted list is inefficient for contact tracing as the search algorithm is still taking too long. It was suggested to use a hash table with the date and hour of diagnosis as the hash key. For example, someone who was diagnosed at 3:15pm on 13 September 2022 would be assigned the key 13092215 where 130922 is the date and 15 is the hour of diagnosis.

(b) Explain what a hash key is and state the worst-case scenario time complexity of a hash table search. [2]

(c) Give **one** advantage and **one** disadvantage to the method provided in finding the hash key. [2]

(d) State **two** methods to resolve collisions in a hash table. Explain which method would be better suited in the context of contact tracing. [3]

7. A supermarket has a list of items in its inventory. Use `items = ['beef', 'apple', 'eggs', 'carrots', 'detergent', 'fish']` as an example, to answer this question.

(a) Explain how a merge sort algorithm can be used to sort `items` in alphabetical order. [4]

(b) State the worst-case time complexity of the above sort algorithm. [1]

The following sorting function was provided by the IT team in the supermarket.

```
1  def sort(array):
2      n = len(array)
3      for i in range(0,n-1):
4          for j in range(0,n-1):
5              if array[j] > array[j+1]:
6                  array[j+1], array[j] = array[j], array[j+1]
```

(c) Find the number of comparisons that are made when `items` is sorted using this algorithm. [2]

(d) Modify the `sort(array)` code to improve its efficiency. State the number of comparisons made if your algorithm is used on the list `items`. State the worst-case time complexity of your algorithm. [3]

Solution

5. The arrays `PollData[1:10]` and `CardData[1:10]` store data.

PollData	12	85	52	57	25	11	33	59	56	91

CardData	11	12	25	52	33	56	57	59	91	85

(a) State why it will take lees time to complete an insertion sort on `CardData` than on `PollData`. [1]

(b) Write an algorithm, in **pseudocode**, that performs an insertion sort on the `CardData` array. [5]

(c) (i) A binary search algorithm is used to find a specific value in an array.

Explain why an array needs to be sorted before a binary search algorithm can be used. [2]

(ii) The current contents of `CardData` are shown.

11	12	25	33	52	56	57	59	85	91

Explain how a binary search will find the value `25` in `CardData`. [4]

(d) Complete this procedure to carry out a binary search on the array show in **part (c)(ii)**. The missing parts are labelled A, B, C and D. [4]

```
PROCEDURE BinarySearch(CardData, SearchValue)
     First ← 1
     Last ← ARRAYLENGTH(. . . . A . . . .)
     FOUND ← FALSE

     WHILE (First <= Last) AND (NOT Found)
          Midpoint ← . . . . B . . . .
          IF CardData[Midpoint] = SearchValue
               Found ← TRUE
          ELSE IF SearchValue < CardData[Midpoint]
               Last ← . . . . C . . . .
          ELSE
               First ← . . . . D . . . .
          ENDIF
     ENDWHILE
ENDPROCEDURE
```

(e) Using the contents of `CardData` shown in **(c)(ii)**. Write an algorithm, in **pseudocode**, for a linear search that will read a value, X, and output the position of X or output a message stating that X is not in `CardData`. [5]

6.

(a) A binary search algorithm should be used instead as the list is sorted chronologically. It would be a more efficient method as the time complexity is O(log n) compared to O(n) if a linear search was used.

(b) A hash key is used to find a location in a Hash Table to search for or store data. The worst case scenario time complexity for hash table search is O(n).

(c) This method of finding a hash key is simple. (advantage) However, as most diagnosis would happen in the day time, during the working hours of clinics and hospitals, most of the data would

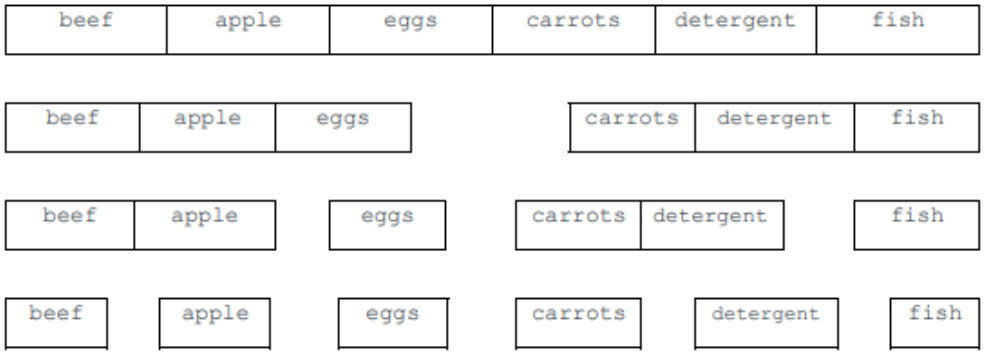
be clustered within a certain range of the hash table, leading to higher chances of collision. (disadvantage)

(d) 2 possible methods are chaining and linear probing. Chaining is the better strategy.
Chaining is a better option as it would allow better tracking of trends. Linear Probing would also have a huge disadvantage of having a record stored at a time that is not related to the person who was diagnosed. For example, if 24 people were diagnosed in the same hour, the 24^{th} person would be stored in a record that is 1 full day away.
Linear Probing is also not as good as linear probing may result in skipping certain hash keys, leading to a breakage in the "links" or breakage of the "contact tracing", which may be better tracked if chaining was used.

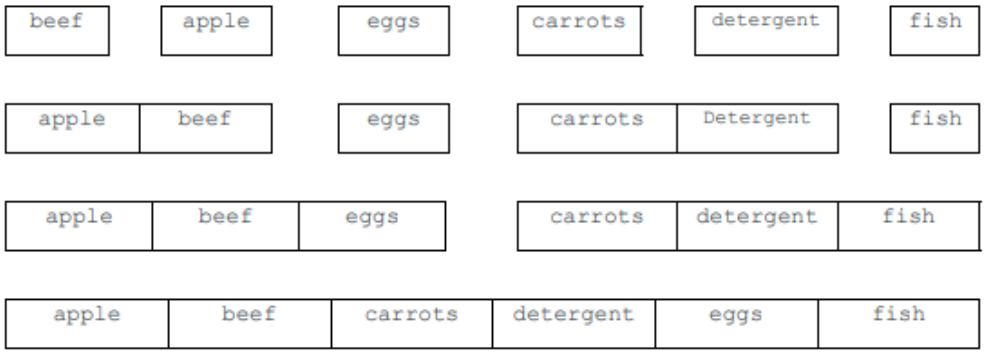
Marker's Comments:
(b) Most students understood that a hash key is derived from a hash function, but many then failed to give an explanation as to how it could be used.
(c) Some students incorrectly interpreted this as to give one advantage of using the hash key, rather than the method of generating the hash key.

7.

(a) Merge sort is a recursive algorithm which continuously **splits the array in half until it cannot be further divided**. If the array has multiple elements, split the array into halves and **recursively**. If the array becomes empty or has only one element left, the dividing will stop and the empty array or one-element array is considered sorted. (i.e. it is the base case to stop the recursion)



Finally, when both halves are sorted, the merge operation is applied. Merge operation is the process of taking two smaller sorted arrays and combining them to eventually make a larger one. The two halves from before are merged into a larger one and this merge continues until all the pairs of halves are merged together.



When all the halves are merged back together, the merge sort is complete.

(b) $O(n \log n)$

(c) 25 times.

n = len(items) = 6

	In line 3, the for-loop happens for 0 to 4 inclusive and repeats lines 4 to 6. (5 times) In line 4, the for-loop happens for 0 to 4 inclusive and repeats lines 5 to 6. (5 times) Hence, the algorithm repeats line 5 (5X5 times)
(d)	Step 4 change to `for j in range(0,n-1-i):` This reduces the number of checks to $5+4+3+2+1=15$. Time complexity = O(n^2)

	`def sort(array):`
	`  max = len(array)`
	`  swapped = True`
	`  while swapped:`
	`  swapped = False`
	`  max -= 1`
	`  for j in range(0,max):`
	`  if array[j] > array[j+1]:`
	`  array[j+1], array[j] =`
	`  array[j], array[j+1]`
	`  swapped = True`
	
	9 comparisons (or 10 if max is fixed as len(array)-1)
	O(n^2)

```
def sort(array):
    n = len(array)
    last_check = n-1 # initialise flag
    while last_check > 0:
        last_swap = 0
        for j in range(0,last_check):
            if array[j] > array[j+1]:
                array[j+1], array[j] =
```

```
                        array[j], array[j+1]
                last_swap = j # move flag
        last_check = last_swap
```

8 comparisons

O(n^2)

Q7 Marker's Comments

In part (a), many students did not indicate that the algorithm is recursive. (or divide-and-conquer). Some students also explained the wrong order of merge/split. As the split is by halving, the merge is reversing the halving. This means that there will not be a point where we have:

`['apple', 'beef'], ['carrots', 'eggs'], ['detergent', 'fish']`

Some students also did not use `items` when it is required in the question.

For part (b) and (d), complexity should be written as `O(f(n))`, `where O()` stands for order of magnitude. Many students did not write complexity in a proper manner.

For part (d), it is difficult to explain the indentation or replacement with words. (eg. for answer where a `swap` flag is added, many students said "add `swap=True` after line 6", it is unsure where the indentation should be.) Hence, it is strongly advisable that if modifications are made, the code is re-written.