DUNMAN HIGH SCHOOL
Preliminary Examination
Year 6

COMPUTING

Paper 1 Written

9569 / 01

23 Sep 2022

3 hours

READ THESE INSTRUCTIONS FIRST

An answer booklet will be provided with this question paper. You should follow all 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 Robots are now used to pack the National Day Parade (NDP) funpacks after Major Chua Boon Kiat, head of NDP packs production, researched automation options in the industry amid the Covid-19 pandemic to support the business operations of the NDP.

The speed and maintenance schedule of the packing robots are described below.

If the robot

- has been packing heavy items, maintenance check needed after 5 hours of work
- has been packing non-heavy items, maintenance check needed after 10 hours of work
- has a motor temperature of above 60 degree celsius during packing, packing speed needs to be reduced by half
- had not been used for more than a total of 100 hours since the last maintenance check, no maintenance check should be conducted (regardless of item weight it has been packing)

(a) Create a decision table showing all the possible conditions and actions. [4]

Conditions								
Not used for > 100 hours	Y	Y	Y	Y	N	N	N	N
Packing heavy-weight items	Y	Y	N	N	Y	Y	N	N
> 60 degrees celsius	Y	N	Y	N	Y	N	Y	N
Actions								
No maintenance check	X	X	X	X				
maintenance check after 5h					X	X		
maintenance check after 10h							X	X
Reduce packing speed by half					X		X	

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

Conditions					
Not used for > 100 hours	Y	N	N	N	N
Packing heavy-weight items	-	Y	Y	N	N
> 60 degrees celsius	-	Y	N	Y	N
No maintenance check	X				
maintenance check after 5h		X	X		
maintenance check after 10h				X	X
Reduce packing speed by half		X		X	

2 The NDP committee analyses public sentiments through collecting adjectives posted on social media newsfeeds.

These adjectives will be placed in a binary search tree (BST) data structure for further analysis work.

Study the following pseudocode which contains an error.

```
TYPE node

    data: String
    left: node
    right: node

ENDTYPE


PROCEDURE inorder(root)

    IF root = NULL THEN
        inorder(root.left)
        OUTPUT root.data
        inorder(root.right)

ENDPROCEDURE

FUNCTION insert(root, data)

    IF root = NULL THEN

        return_node ← node
        return_node.left ← NULL
        return_node.right ← NULL
        return_node.data ← data

        RETURN return_node

    ENDIF

    IF data < root.data THEN

        root.left ← insert(root.left, data)

    ELSE

        root.right ← insert(root.right, data)

    ENDIF

    RETURN root

ENDFUNCTION

root ← NULL
root ← insert(root, "joyful")
root ← insert(root, "excited")
root ← insert(root, "proud")
```

```
50  root ← insert(root, "amazed")
51  root ← insert(root, "happy")
52  root ← insert(root, "excited")
53  root ← insert(root, "joyful")
54  root ← insert(root, "joyful")
55
56  OUTPUT "Inorder traversal of the given tree:"
57  inorder(root)
```

(a) State the type of error that occurs at line 12 and rewrite the line. [2]

(i) Logic error

(ii) `IF root <> NULL THEN`

(b) Draw the tree structure of the above BST. [2]

(i)
```
      joyful
     /      \
```
(ii)
```
  excited      proud
```
(iii)
```
    /   \       /
```
(iv)
```
amazed happy  joyful
```
(v)
```
          /          \
```
(vi)
```
    excited          joyful
```
(vii)

(c) Draw the tree structure of **(b)** after its root value is deleted. [1]

(i) inorder: amazed excited excited happy joyful joyful joyful proud

(ii)
```
       happy
     /       \
```
(iii)
```
  excited      proud
```
(iv)
```
    /   \        /
```
(v)
```
amazed excited joyful
```
(vi)
```
                    \
```
(vii)
```
                     joyful
```
(viii)

(ix) OR

(x)

(xi)
```
      joyful
     /      \
```
(xii)
```
  excited      proud
```
(xiii)
```
    /   \       /
```
(xiv)
```
amazed happy  joyful
```
(xv)
```
          /
```
(xvi)
```
    excited
```
(xvii)

(d) The above pseudocode is undergoing a redesign so that `inorder(root)` shows the count of the adjective as per the following output where the number in `( )` is the number of duplicates.

```
amazed ( 1 )
excited ( 2 )
happy ( 1 )
joyful ( 3 )
proud ( 1 )
```

A new variable `count` is used to store the number of adjectives if there are duplicates. Some of changes include:

- `count: Integer` added at line 6
- `return_node.count ← 1` added at line 27
- `OUTPUT root.data, "(", root.count, ")"` replaces the code at line 14

These changes however are insufficient.

Making use of `count`, write a segment of pseudocode to be inserted at line 31 in addition to the above changes to achieve the adjective count output shown above. [3]

(i) First three lines [3]

```
(1)          IF data = root.data THEN
(2)              root.count ← root.count + 1
(3)              RETURN root
(4)          ENDIF
```

3 The security team for the NDP protects their digital communication using the Rivest–Shamir–Adleman (RSA) cryptosystem so as to prevent leakage of details such as the route the president takes to the Marina Bay Floating Platform.

The security of RSA relies on the practical difficulty of factoring the product of two large prime numbers.

For example, to decode a message encrypted with RSA number

```
152260502792253336053561837813263742971806811496138068865790849458012296
325895289765400035069200613 9
```

the decoder needs to know that its two prime factors are

```
379752279369436739228088727554456278545655366381 99
```
and
```
400946909509208810306837352927614683892148997240 61
```

A code breaker develops the procedure `prime_factors_iter(RSA_number, candidate)` to print out the factors of the `RSA_number`, given a starting possible prime number as `candidate`.

```
01  PROCEDURE prime_factors_iter(RSA_number: INTEGER, candidate:
    INTEGER):
02
03      WHILE candidate * candidate <= RSA_number
04          IF RSA_number % candidate <> 0 THEN
05              candidate ← candidate + 1
06          ELSE
07              RSA_number ← RSA_number DIV candidate
08              OUTPUT candidate
09          ENDIF
10      ENDWHILE
11
12      IF RSA_number > 1 THEN
13          OUTPUT RSA_number
14      ENDIF
15
16  ENDPROCEDURE
```

(a) For testing purposes, we will be using 72857 and 40 as parameters to dry run the function. Complete the trace table for `prime_factors_iter(72857, 40)`. [5]

The table headers have been provided and the first two rows have been filled in for you. Simply complete the subsequent rows without copying the headers and current values.

candidate * candidate <= RSA_number	RSA_number% candidate <> 0	candidate	RSA_number	OUTPUT
		40	72857	
True	True	41	72857	
True	False	41	1777	41
True	True	42	1777	
True	True	43	1777	
False		43	1777	1777

(b) Write pseudocode for function `prime_factors_recur(RSA_number, candidate)`, a recursive version of `prime_factors_iter(RSA_number, candidate)`. [4]

```
01  FUNCTION prime_factors_recur (RSA_number: INTEGER, candidate:
    INTEGER)
02
03      IF candidate * candidate > RSA_number THEN
04          OUTPUT(RSA_number)
05          RETURN
06      ENDIF
07
08      IF RSA_number % candidate <> 0 THEN
09          RETURN prime_factors_recur(RSA_number, candidate + 1)
10      ENDIF
11
12      OUTPUT candidate
13
14      RETURN  prime_factors_recur (RSA_number DIV candidate,
    candidate)
15
16  ENDFUNCTION
17
```

(c) Describe how a stack is used in the recursion of **(b)**. [3]

(i) Each time when the recursive call is made, an activation record is created including all **parameters/local variables/return address and return value(s)** *of the currently evaluated RSA_number and candidate*. This activation record is **pushed** onto the run-time stack.

(ii) The **top** activation record in the run-time stack is always the procedure **currently being executed**

(iii) When the procedure terminates/returns, its activation record is **popped** from the stack.

(d) Suggest why the recursive method may be less preferred to the iterative method for this scenario. [2]

(i) is slower because of the need to manage stack operations. Recursion uses more memory because of the need to store multiple stack frames. Memory may be scarce due to large RSA numbers. Also, speed is important in code breaking. (or any relevant situation).

(e) The factors of `RSA_number` are to be stored in a data structure for later use.

(i) Suggest why a static data structure may be more suitable than a dynamic data structure. [1]

(1) Since there are only two factors, there is no need for resizing of the data structure.

(ii) State one benefit a static data structure has over a dynamic data structure. [1]

(1) Faster access speeds [1]

(f) A check digit is added at the end of a 5-digit RSA number factor such that the last digit is a modulus-eleven check digit. Each digit is weighted, with the first digit having a weight of 6 and each subsequent digit decreases its weight by 1.

(i) Suggest why a check digit is added. [1]

(1) Validation check for the other digits ("accuracy" may not be a suitable a sufficient answer as the word may mean different things such as different accuracy levels/thresholds etc)

(ii) Calculate the check digit for a RSA public key `98389`. Show your working. [3]

(1) 6x9 = 54
(2) 5x8 = 40
(3) 4x3 = 12
(4) 3x8 = 24
(5) 2x9 = 18 [1]
(6) Sum: 148 [1]

Check digit: 11-(148 mod 11) = 6 [1]

4 A costume designer for NDP performers needs software to manage orders and engages a software engineer who uses object-oriented programming to model the ordering system to calculate the total price for a costume.

For each costume, the following items of data are recorded:

- an order number
- any special requirements (e.g. use of any environmentally-friendly materials)
- base price (in $)

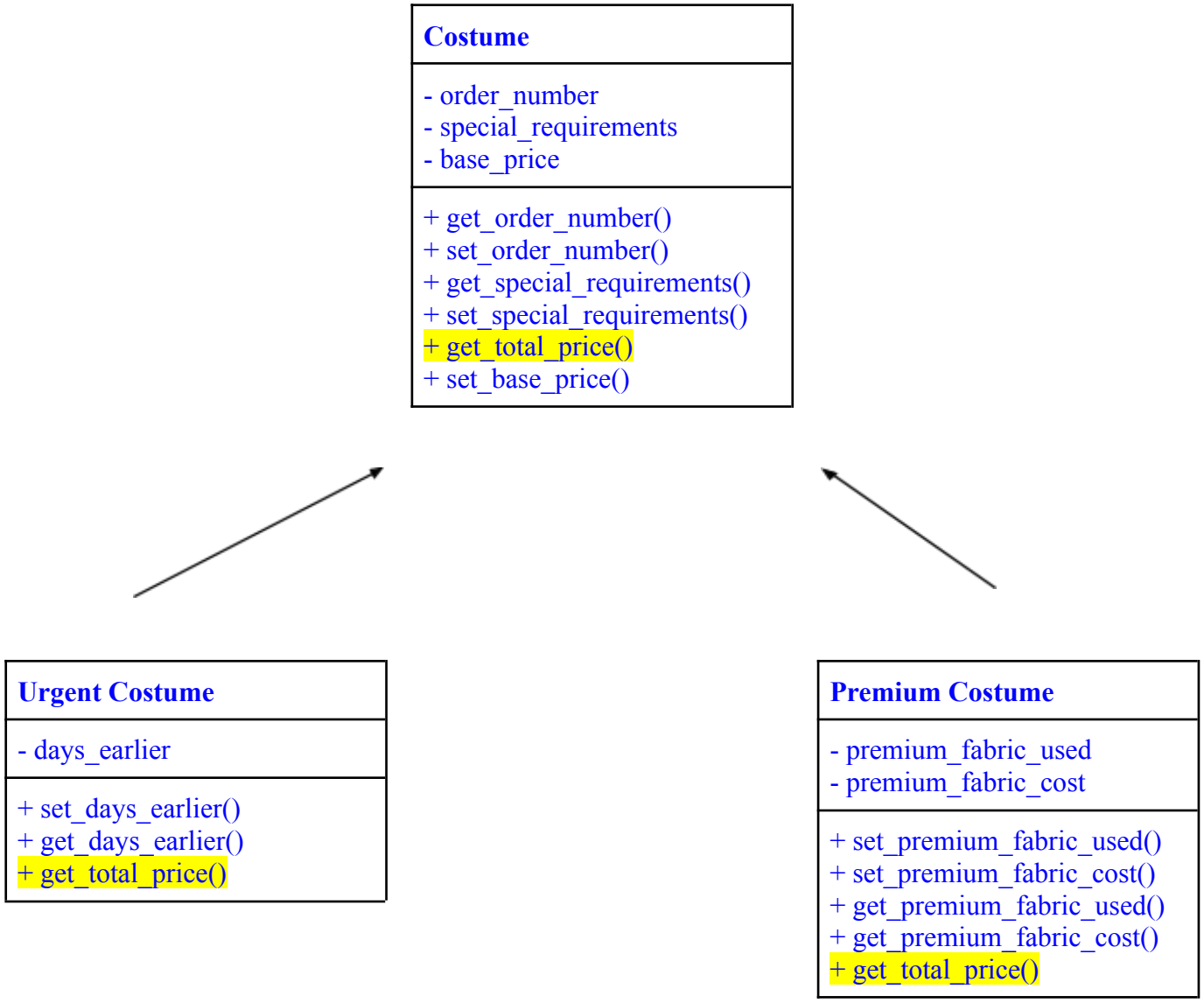
For urgent orders, the number of days the costume is needed earlier than the standard manufacturing time of two weeks is recorded. The total price includes an additional charge of 10% of the original price per day earlier.

For premium orders, the area of premium fabric used (in addition to normal fabric) and the per-square-metre cost of premium fabric is also recorded. The total price includes an additional charge of the amount of premium fabric used multiplied by the per-square-metre cost of premium fabric.

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

(1) the superclass
(2) any subclasses
(3) inheritance
(4) polymorphism
(5) properties
(6) appropriate methods

[9]



(b) Apart from doing a presence check, state and elaborate on two validation techniques that might be applied to the number of days the costume is needed earlier. [2]

(i) Range check - between 0 to 14 days [1].

(ii) Type check - integer data type [1]

(c) Define polymorphism and describe, with examples, how it occurs in this situation. [2]

(i) Allow methods in the subclass to have the same name to allow for different behaviour. [1]

(1) (i.e. a method with the same name to have many different implementations. when a method, inherited from a parent (super-) class, has its implementation changed to better suit the child (sub-) class. Overriding is a form of polymorphism)

(ii) Subclass Urgent Costume or Premium Costume has same method get_total_price() but different implementation - calculated price takes into account additional charges instead of just base_price. [1]

(d) Explain why polymorphism is important in general and how it will benefit this software engineer. [2]

(i) Enables code generalisation where any code that can be generalised and used for 'similar but different' problems is valuable [1],

(ii) and will save time in developing and testing solutions. [1]

5 NDP funpacks contain a collection of items, including snacks, drinks, a fan and a flag. In previous years, these were packed by inmates and full-time national servicemen (NSF).

The database below is used to store data on packing activity.

Task

TaskID	TaskDescription
1	Packing of fans and flags
2	Packing of snacks
3	Cartoning of fun packs

TaskPacker

TaskID	PackerID	HoursWorked
3	4	32
1	2	22
3	1	13
3	3	14
1	4	19
1	6	30
2	7	41
3	8	13
1	5	35
2	2	50
2	1	31

Packer

PackerID	PackerName	PackerTypeID	PackerTypeName	ItemsPackedPerHour
1	Lenny Chee	1	NSF	312
2	Gilbert Poh	2	Inmate	232
3	Robert Singh	1	NSF	122
4	Adi Othman	1	NSF	133
5	Fu Su Hui Caroline	2	Inmate	423
6	Tony Chua	2	Inmate	233
7	Ng Xue Qian Garth	1	NSF	281
8	Kok Han	1	NSF	296

(a) Identify a foreign key and the corresponding table in which the foreign key lies. [1]

(i) TaskID or PackerID in the TaskPacker table

(b) State whether table TaskPacker is in first normal form (1NF). Explain why. [2]

(i) Yes

(ii) Any of the following

(1) data is atomic:
(2) No columns with similar or repeated data
(3) Each data item cannot be broken up any further - no commas in the data

(c) Explain why the table Packer is not in third normal form (3NF). [2]

(i) There is transitive dependency[1]

(ii) PackerTypeName is dependent on non-key attribute PackerTypeID [1]

(d) Decompose the Packer table into 3NF and write table descriptions for each table. [2]

(i) Packer(PackerID, PackerName, PackerTypeID*, ItemsPackedPerHour)

(ii) PackerType(PackerTypeID, PackerTypeName)

(e) More information such as address and bank account details is to be collected from the packers. Suggest an action that must be taken regarding the collection, disclosure and use of this data under the Personal Data Protection Act. [1]

(i) Any response in the categories of

(1) ensure data is only used for the purpose it was stated to be collected for
(2) ensure data and data access history is able to be retrieved by the individual (or Data Protection Officer) according to guidelines stated in Access and Correction Obligation
(3) do not transfer data to another country unless according to regulations prescribed
(4) cease retention of personal data or remove the means by which the personal data can be associated with particular individuals when it is no longer necessary for any business or legal purpose
(5) ensure there are reasonable security arrangements to protect the personal data possessed or controlled to prevent unauthorised access, collection, use, disclosure or similar risks.
(6) Accuracy Obligation: Make reasonable effort to ensure that the personal data collected is accurate and complete, especially if it is likely to be used to make a decision that affects the individual or to be disclosed to another organisation.

(f) After the NDP packing completes, NDP organisers decide to create an archive and not a backup of some of the database data. Explain why an archive and not a backup is created for this. [2]

(i) Why not backup

(1) Not current working files
(2) No need for quick restoration in event of system failure

(ii) Why archival

(1) It is data that's no longer in day-to-day use but still needs to be retained for historical purpose, future audit or legal requirements or ad-hoc reference eg past year students records (due to their their enduring cultural, historical, or evidentiary value)

6 Quicksort is a divide-and-conquer algorithm that is commonly used for sorting.

(a) State the time complexity (worst case) for quicksort. [1]

(i) O(n2)

Study `modified_quicksort`. You may assume `InsertionSort(arr, start, end-1)` and `mergeSort(arr, start, end-1)` will successfully sort array `arr` using in-place insertion sort and merge sort respectively.

```
01  PROCEDURE modified_quicksort(arr : ARRAY)
02
03      maxdepth ← log2(LENGTH(arr)) * 2 // logarithm with base two
04      modified_quicksort_helper(arr, 0, LENGTH(arr), maxdepth)
```

```
05
06  ENDPROCEDURE
07
08
09  FUNCTION modified_quicksort_helper(arr: ARRAY, start: INTEGER,
    end:INTEGER, maxdepth: INTEGER)
10
11      IF end - start < 10 THEN // array is small when less than 10
12          InsertionSort(arr,start,end-1)
13          RETURN
14
15      IF maxdepth <= 0 THEN
16          mergeSort(arr,start,end-1)
17
18      ELSE
19          pos ← partition(arr, start, end)
20          modified_quicksort_helper(arr,start,pos + 1,maxdepth - 1)
21          modified_quicksort_helper(arr,pos + 1,end,maxdepth - 1)
22      ENDIF
23
24  ENDFUNCTION
25
26
27  FUNCTION partition(arr : ARRAY, start: INTEGER, end:INTEGER)
28
29      pivot ← arr[start]
30      low ← start - 1
31      high ← end
32
33      WHILE TRUE
34
35        low ← low + 1
36
37        WHILE arr[low] < pivot
38          low ← low + 1
39        ENDWHILE
40
41        high ← high - 1
42
43        WHILE arr[high] > pivot
44          high ← high - 1
45        ENDWHILE
46
47        IF low >= high THEN
48          RETURN high
49        ENDIF
50
51        temp ← arr[low]
52        arr[low] ← arr[high]
53        arr[high] ← temp
54
55      ENDWHILE
56
57  ENDFUNCTION
```

(b) Describe and explain the disadvantage(s) of quicksort which `modified_quicksort` tries to mitigate at each of the following segment(s) in **(i)** and **(ii)**. Also, explain how `modified_quicksort` (and its helper functions) does the mitigation.

(i) Lines 3, 15, 16 [7]

(1) Quicksort disadvantage

(a) - A poorly chosen pivot (i.e. always the smallest or largest element chosen as pivot) results in a most unbalanced partition

(b) - always divides the array such that one part has 0 elements and other part has n-1 elements. (i.e. when one of the sublists returned by the partitioning routine is of size n − 1 each recursive call processes a list of size one less than the previous list)

(c) - so max number of times the recursion needs to be called (because need to make n − 1 nested calls before we reach a list of size 1. This means that the call tree is a linear chain of n − 1 nested calls. The ith call does O(n − i) work to do the partition, and so in that case quicksort takes $O(n^2)$ time.)

(2) modified_quicksort advantage

(a) - if Quicksort was not able to get the solution after 2 * log(n) recursions for a branch, probably it hit its worst case and it is degrading to complexity O(n^2)

(b) - and so modified_quicksort switches to use mergesort instead

(c) - mergesort always divides evenly, doesn't depend on any chosen pivot

(d) - Therefore, splitting and merging together involve at most nlog2n operations, which is less than $O(n^2)$, more efficient than quicksort.

(e) See effect in array [20, 19, 18, 17, **15, 16,** 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

(ii) Lines 11 - 12 [3]

(1) quicksort disadvantage

(a) - quicksort doesn't do so well with small lists; insertion sort is more efficient with small data size than quicksort.

(b) - because unnecessary time spent on creating multiple recursive stack operations for a small list.

(2) modified_quicksort advantage

(a) - when remaining unsorted sublist is smaller than 10, it switches to use insertion sort

(3) See effect in array [20, 19, 18, 17, **15, 16,** 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

7 The final colour displayed by each pixel of the NDP's huge LCD screen is a mix of 3 colours, each of a different intensity. The LCD screen's display processor receives the 3-colour data in binary from an imaging computer attached to it. This imaging computer converts the colours from a hexadecimal `#RRGGBB` (`R` = red, `G` = green, `B` = blue) representation generated by imaging software into binary.

(a) Suggest the number of variations of final colours each pixel can display, and show your working. [1]

(i) (#00) and (#01 to #FF) -> 1+ 255 -> 256^3 = 16,777,216 final colours. [1]

When the Singapore national flag is displayed (red and white colours only), a few pixels on the screen wrongly display a shade of blue. A log file reveals that the microprocessor onboard the LCD screen receives the `#RRGGBB` as `0010 1110 1000 1001 1011 1001` in binary from the imaging computer for these pixels.

(b) Explain why the fault lies with the imaging computer and not the LCD screen's hardware. To support your answer, convert the binary values into denary and show your working. [2]

(i) All basic binary to denary conversion correct [1]

(1) Red intensity: $(00101110)_2 = (0 \times 2^8) + (0 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (-16 \times 2^0) = (46)_{10}$

(2) Blue intensity: $(10001001)_2 = (1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = (137)_{10}$

(3) Green intensity $(10111001)_2 = (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = (185)_{10}$

(ii) Correct display should show less blue/green intensity than red intensity [1]

(iii) So, fault lies with imaging computer ; it transmitted wrong colour values.

(iv) For ref: `0010 1110 1000 1001 1011 1001  #2E89B9`

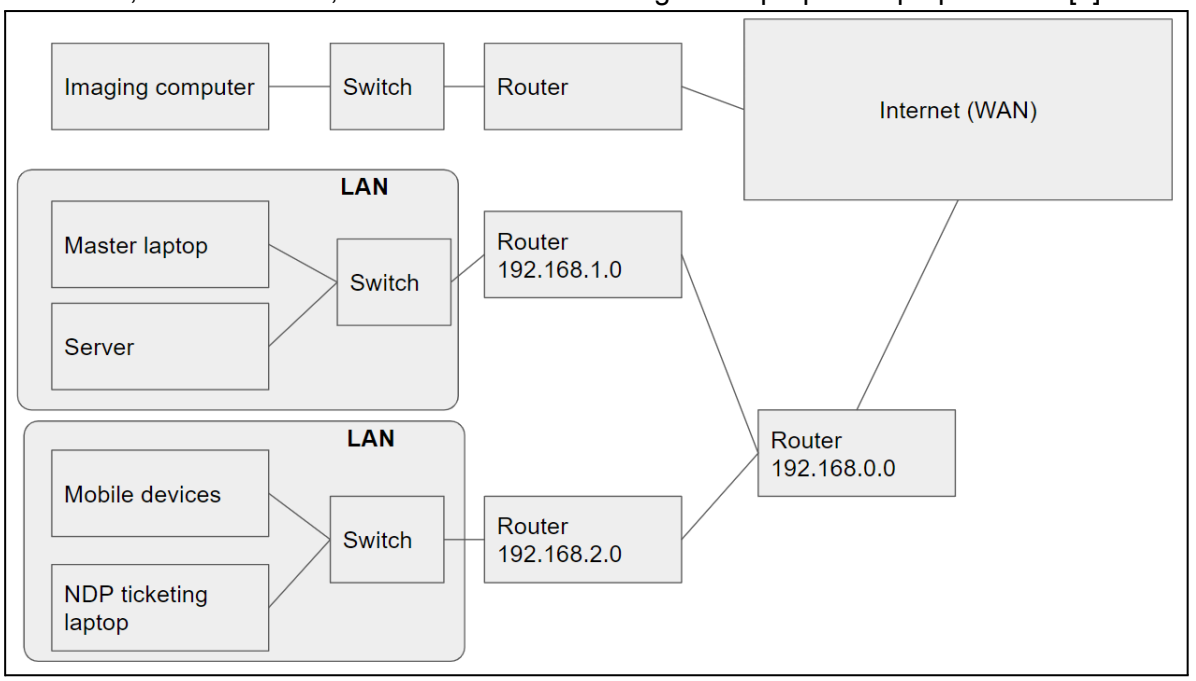
The computer's imaging software connects to the Internet to receive display information sent from a master laptop located at the NDP control room some distance away.

The NDP control room's network layout includes the master laptop which obtains display media from a web server. The master laptop and the server are placed on a different subnet from the main subnet which connects the rest of the control room devices: a general purpose laptop and mobile devices of the NDP committee.

All devices are connected to the Internet.

(c) Draw a network diagram of the above configuration and label the LAN, Internet, router(s), WAN, web server, switches/hubs, mobile devices and the general purpose laptop. [5]

(i)



(d) Suggest a reason for the above subnetting. [1]

(i) Makes it difficult for attackers to intercept sensitive data since network traffic doesn't pass into shared bandwidth / reduce congestion / efficient (travel shorter distance cos pass through less routers)

The server also provides email, web hosting and file transfer services. The requests to this server are sent from clients using an internet protocol.

(e) With reference to TCP/IP layers and protocols, explain how different requests arriving at this server are handled. Support your answer with examples. [3]

(i) Data is handled at transport and application layer (since request has arrived at the server (device on network)) [1]

(ii) The server decides which service/server application is required based on the **port** number used to pass the data.[1]

(iii) Message is decoded using the appropriate communication protocol (SMTP for email, HTTP for web, FTP for file transfer, etc) [1]

(iv) a response may be sent back to the client

A security analyst suspects a malicious hacker is behind the wrongly coloured pixels and begins her investigation.

She watches the duty technician set up the LCD display screen: He types in the correct URL into the imaging software's web browser authentication page to establish a data transfer connection with the master laptop. She then conducted some tests and suspected a DNS spoofing attack as she

discovered that the authentication page used by the technician was from a malicious IP address despite it looking authentic.

.

To confirm her discovery, she navigated to the authentication page by typing in its server IP address (instead of the URL) into the browser and this led to the correct pixels displayed on the NDP screen.

(f) Suggest why it may not be a good long-term solution to require the duty technician to do what she did (i.e. type the IP address instead of the URL). [1]

(i) difficult to remember, ip address can change [1]

(g) Explain what a hacker could have done at one of the Internet Service Providers (ISPs) to alter the DNS lookup process to cause the above DNS spoofing attack. [3]

(i) 1) hacker hacked one of the ISP DNS servers,

(ii) 2) change DNS records

(iii) 3) redirect DNS requests to malicious sites

(1) The DNS server sends your computer the relevant IP address.

(a) Note: DNS does not return the IP address to your browser.

A log file reveals that despite the DNS spoofing attack there are times the browser still successfully connects to the correct IP address using the URL.

(h) Explain how packet switching could have caused these successful connections. (Assume the setup technician always types the URL into the browser authentication page.) [3]

(i) Packet switching diverts DNS packets through different routes [1]

(ii) to different a DNS server which has not been hacked [1]

(iii) possibly due to usually lowest network traffic or node/router/DNS availability [1]

A digital signature can be implemented to ensure the correct authentication page will always get loaded in the browser of the imaging computer.

(i) Describe how a digital signature works in the above scenario. [4]

(i) Each missing point deduct 1m

(1) 1. Master computer runs a hash function against the page data, to produce a hash total or digest.

(2) 2. master computer encrypts the digest with private key. This is the encrypted digest.

(3) 3. When imaging computer receives the page data, it uses master computer's public key to decrypt the digest.

(4) 4. imaging computer runs the hash function on the page data and checks whether the digests match.

(5) 5. if doesn't match, request a different DNS server to resolve the IP address of the master computer.

(j) Explain why a digital signature can prove that the correct authentication page is loaded. [2]

(i) Encrypted digest (the digital signature for the page data) **only** can be produced by master computer…

(ii) …because when the digests match, imaging computer knows that the page data was not changed in transit and that the page data is really from the master computer, proving the integrity of the data.

During the NDP performance, monitoring software running on the server warns the system administrator of a denial of service (DOS) attack.

(k) Suggest why the warning was generated by the monitoring software. [1]

(i) Any related to

(1) Black listed botnet
(2) Overload of web server, performance of web server slows down
(3) Higher than normal traffic.

(l) State an authentication technique to limit access to the imaging computer. [1]

(i) Any related to

(1) biometrics, for example: fingerprints, facial recognition, iris scans
(2) token values, such as from a physical device, a mobile phone or a software application
(3) use two-factor authentication (2FA), which uses two different ways of authentication for better security.

(m) Suggest how installing a firewall on the imaging computer will further protect this network.[1]

(i) Any related to

(1) prevents unauthorised access to and from the network.
(2) Filters network traffic by blocking access by unauthorised users.
(3) Prevents programs from communicating via unauthorised port numbers.
(4) Note: does not protects computer from running unauthorised software