Name:		Index Number:		Class:	

DUNMAN HIGH SCHOOL
Mid-Year Examination
Year 6

COMPUTING (Higher 2)

Paper 1 Written

9569/01

7 July 2023
3 hours

READ THESE INSTRUCTIONS FIRST

Write your name, index number and class on the work you hand in.
Write in dark blue or black pen on both sides of the paper.
You may use an HB pencil for any diagrams or graphs.
Do not use staples, paper clips, glue or correction fluid.

Answer **all** the questions.

Approved calculators are allowed.

You are reminded of the need for good English and clear presentation in your answers.
Please ask the invigilator if you require additional paper.

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 **20** printed pages.

1 An insurance company calculates the cost of car insurance from a basic price.

The driver may:

- get a discount on the basic price of the insurance
- have to pay an extra charge

The decision is arrived at as follows:

- for a driver aged 25 or over:
    - 5% discount if no previous accident
    - no discount if a previous accident
- for a driver under the age of 25:
    - 5% discount if no previous accident and licence held for 3 or more years
    - no discount if a previous accident but licence held for 3 or more years
    - no discount if no previous accident but licence held for less than 3 years
    - 10% extra charge if a previous accident and licence held for less than 3 years

(a) Complete the decision table. [6]

Conditions	Age under 25	Y	Y	Y	Y	N	N	N	N
	Previous accident	Y	Y	N	N	Y	Y	N	N
	Licence held for 3 or more years	Y	N	Y	N	Y	N	Y	N
Actions	10% extra discount								
	No discount								
	5% discount								

(b) Simplify your solution by removing redundancies. [3]

Conditions	Age under 25								
	Previous accident								
	Licence held for 3 or more years								
Actions	10% extra discount								
	No discount								
	5% discount								

(c) The simplified table produced in **part (b)** is used as a design for pseudo-code. The following identifier table shows the parameters to be passed to the function `CostPercentageChange`. This function returns the percentage change from the basic price as an integer. A discount should be shown as a negative integer. An extra charge should be shown as a positive integer.

Identifier	**Data type**	**Comment**
`DriverAge`	`INTEGER`	Age of driver in years
`HadAccident`	`BOOLEAN`	Whether driver has had a previous accident
`YearsLicenceHeld`	`INTEGER`	Number of years the driver has held licence

Write **pseudo-code** for this function. [6]

2 **Figure 1** shows a bubble sort algorithm represented using pseudo-code. The algorithm sorts the data in a list `L`.

Figure 1

```
01   PROCEDURE BubbleSort(L)
02        N ← LEN(L) - 2
03        Count1 ← 0
04        WHILE Count1 < LEN(L) - 1
05             FOR Count2 ← 0 TO N
06                  IF L[Count2] > L[Count2 + 1] THEN
07                       Temp ← L[Count2]
08                       L[Count2] ← L[Count2 + 1]
09                       L[Count2 + 1] ← Temp
10                  ENDIF
11             ENDFOR
12             Count1 ← Count1 + 1
13        ENDWHILE
14   ENDPROCEDURE
```

(a) Describe two changes with additional statements, for example `16.1, 16.2 …` in between `16` and `17` that could be made to this bubble sort algorithm that would be likely to result in fewer comparisons being made when sorting the list `L`. The algorithm should still be a bubble sort algorithm if your suggested changes were made. [4]

Figure 2 lists some time complexities, where n is the size of the problem input and k denotes a constant.

Figure 2

```
O(1)
O(n^k)
O(k^n)
O(n)
O(log n)
O(n log n)
```

(b) State which of the time complexities shown in **Figure 2** is the time complexity of the linear search algorithm and explain why it has that time complexity. [2]

(c) State which of the time complexities shown in **Figure 2** is the time complexity of the binary search algorithm and explain why it has that time complexity. [2]

3 A computer program is being developed for a car hire company. The program must store, in a file, details of the 600 vehicles that the company owns.
The records in the file will be stored and retrieved using hashing.
An alternative method that could be used instead of hashing would be to store the records in order of registration number and use a search algorithm such as binary search for retrieval.

(a) (i) State one advantage of organising the data using hashing instead of organising the data in order by registration number. [1]

(ii) State one advantage of organising the data in order by registration number instead of organising the data using hashing. [1]

Each vehicle is uniquely identified by its registration number. A registration number consists of:

- two alphabetic characters
- followed by two numeric digits.
- followed by three further alphabetic characters.

An example registration number is **DA18CFE**.
The programmer has chosen the hash function below to calculate a hash value from a registration number.

```
Hash value = ( position in alphabet of letter at position 1 +
position in alphabet of letter at position 2 * 10 +
numeric digit at position 3 * 100 +
numeric digit at position 4 * 500 ) MOD 1000
```

where **MOD** is the **modulo** operator, which returns the remainder of dividing two numbers

For the example **DA18CFE** the hash value would be calculated as follows:

```
Hash value = ( position in alphabet of 'D' (4) +
               position in alphabet of 'A' (1) * 10 +
               1 * 100 +
               8 * 500 ) MOD 1000
             = 4114 MOD 1000
             = 114
```

(b) Calculate the hash values for the following two registration numbers. You may use the space provided for working, if required.

(i) AE21KWB [1]

Working /Hash value

(ii) KD70DAF [1]

Working /Hash value

(c) Calculating the hash values for the two registration numbers in part **3(b)** has produced a collision.

In the context of storing data in files using hashing, explain the effect of this collision and how this might be dealt with. [2]

4 A stack Abstract Data Type (ADT) has these associated operations:

- create stack
- add item to stack (push)
- remove item from stack (pop)

The stack ADT is to be implemented as a linked list of nodes.

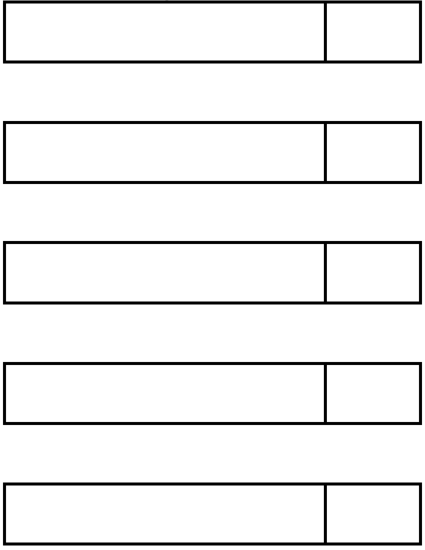
Each node consists of data and a pointer to the next node.

(a) There is one pointer: the top of stack pointer, which points to the last item added to the stack.
Fill in **Figure 3** to show the final state of the stack after the following operations are carried out.

```
CreateStack
Push("Ali")
Push("Jack")
Pop
Push("Ben")
Push("Ahmed")
Pop
Push("Jatinder")
```

Add appropriate labels to **Figure 3** to show the final state of the stack. Use the space on the left as a workspace. Show your final answer in the node shapes on the right: [3]

Figure 3



(b) Using pseudocode, a record type, Node, is declared as follows:

```
TYPE Node
    DECLARE Name : STRING
    DECLARE Pointer : INTEGER
ENDTYPE
```

The statement

```
DECLARE Stack : ARRAY[1:10] OF Node
```

reserves space for 10 nodes in array `Stack`.

(i) The `CreateStack` operation links all nodes and initialises the `TopOfStackPointer` and `FreePointer`.

Complete **Figure 4** to show the value of all pointers after `CreateStack` has been executed. [4]

Figure 4

`TopOfStackPointer`

`FreePointer`

`Stack`

	`Name`	`Pointer`
[1]		
[2]		
[3]		
[4]		
[5]		
[6]		
[7]		
[8]		
[9]		
[10]		

(ii) The algorithm for adding a name to the stack is written, using pseudocode, as a procedure with the header

```
PROCEDURE Push (NewName)
```

Where `NewName` is the new name to be added to the stack. The procedure uses the variables as shown in the identifier table.

Identifier	Data type	Description
`Stack`	`Array[1:10] OF Node`	
`NewName`	`STRING`	Name to be added
`FreePointer`	`INTEGER`	Pointer to next free node in array
`TopOfStackPointer`	`INTEGER`	Pointer to first node in stack
`TempPointer`	`INTEGER`	Temporary store for copy of FreePointer

```
PROCEDURE Push(BYVALUE NewName : STRING)
     // Report error if no free nodes remaining
     IF FreePointer = 0
          THEN
               Report Error
          ELSE
               // new name placed in node at head of free list
               Stack[FreePointer].Name ← NewName
               // take a temporary copy and
               // then adjust free pointer
               TempPointer ← FreePointer
               FreePointer ← Stack[FreePointer].Pointer
               // link current node to previous top of stack
               Stack[TempPointer].Pointer ← TopOfStackPointer
               // adjust TopOfStackPointer to current node
               TopOfStackPointer ← TempPointer
     ENDIF
ENDPROCEDURE
```

Complete the **pseudocode** for the procedure `Pop`. Use the variables listed in the identifier table. [5]

```
PROCEDURE Pop()

// Report error if Stack is empty





            OUTPUT Stack[                              ].Name

            // take a copy of the current top of stack pointer



            // update the top of stack pointer



            // link released node to free list



ENDPROCEDURE
```

(iii) With procedures `Push` and `Pop` from **part b(ii)** and operations given in **part (a),** complete the trace table to show the procedure calls and value of all pointers after `CreateStack` has been executed. Show your final answer in the node on the right. [5]

Push or Pop	TopOfStackPointer	FreePointer

Stack

	Name	Pointer
[1]		
[2]		
[3]		
[4]		
[5]		
[6]		
[7]		
[8]		
[9]		
[10]		

5 A recursively defined procedure X is defined below:

```
PROCEDURE X(BYVALUE n : INTEGER)
     IF (n = 0) OR (n = 1)
          THEN
               OUTPUT n
          ELSE
               CALL X(n DIV 2)
               OUTPUT (n MOD 2)
     ENDIF
ENDPROCEDURE
```

(a) Explain what is meant by recursively defined. [2]

____

____

(b) Explain how a stack is used during the execution of a recursive procedure. [4]

____

____

____

(c) Dry run the procedure X by completing the trace table for the procedure call: [3]
CALL X(40)

Call number	**n**	**(n = 0) OR (n = 1)**	**n DIV 2**	**n MOD 2**
1	40	FALSE	20	
2				
3				
4				
5				
6				

Output: ____ [2]

(d) State the process that is carried out by procedure X. [1]

____

6 **(a)** A router is connected to a LAN.
Describe the function of a router. [2]

(b) Describe the details of computer addresses that are stored by a router [3]

(c) Figure 5 shows the layers in the TCP/IP stack.

Figure 5

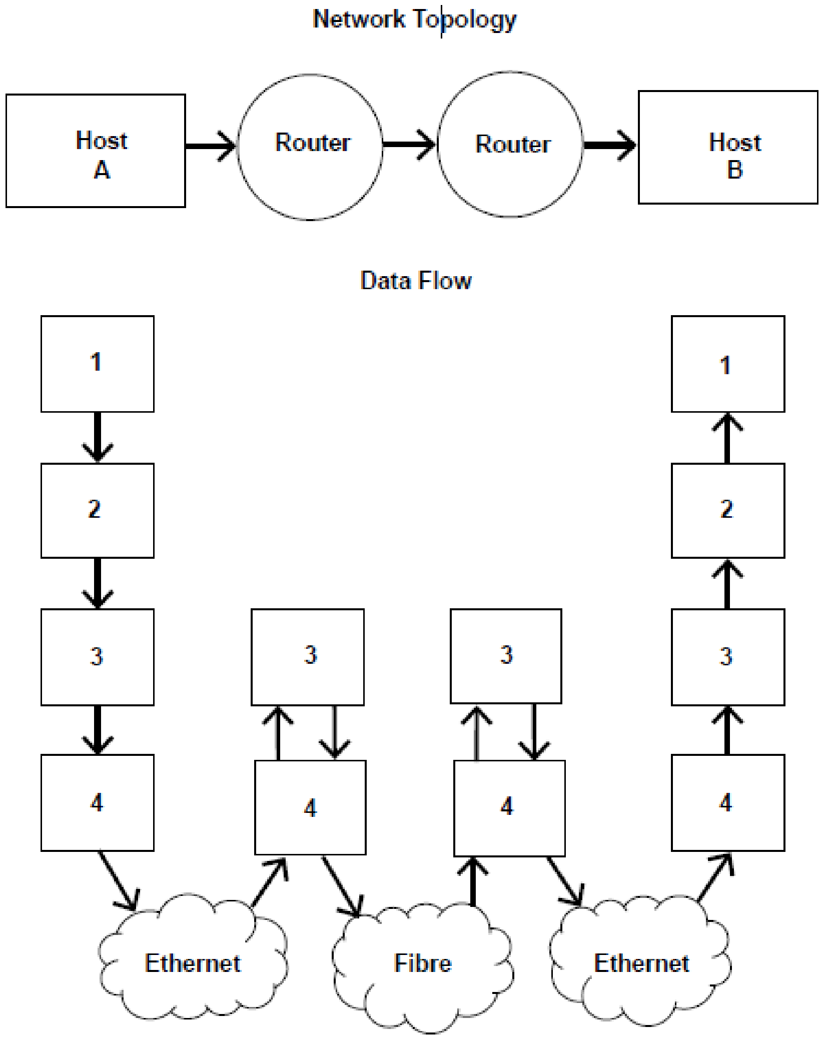


Table below shown the TCP/IP layers used in **Figure 5** above.

	Layer
1	Application (layer)
2	Transport (layer)
3	Network / internet (layer)
4	Link (layer)

(i) Describe the roles of each layer when two devices are communicating over the Internet. [8]

(ii) **Figure 5** shows how a packet travels from Host A to Host B through two routers. Describe, for a packet, the role of the two lower levels of the TCP/IP stack in the router. [4]

(d) The Internet can be used for video conferencing. Data can be transmitted over the Internet using either packet switching or circuit switching.

(i) State **two** problems that could arise if video conferencing were to use packet switching. [2]

(ii) Explain **how** the use of circuit switching overcomes the problems you have identified in **part (i)** [2]

____

____

____

____

____

____

(e) Host A and Host B are involved in a secure communication that uses asymmetric encryption. Host A is sending a message to Host B.

Each computer has a public key and a private key.

(i) Complete the missing words in the following paragraph. [2]

Host A will encrypt the message using ____________________ key. The message will be decrypted by

Host B using ____________________ key.

The security of the communication could be improved by the addition of a digital signature.

(ii) State two benefits of including a digital signature. [2]

____

____

____

____

____

____

7 A database is designed to store data about students at a college and the subjects which they study.

- All students are based in a tutor group.
- A tutor supervises all the students in their tutor group.
- Each subject has one subject teacher only.
- Students study a number of subjects.

This table `StudentSubjects` was a first attempt at the database design.

Table: `StudentSubjects`

StudentName	TutorGroup	Tutor	Subject	Level	SubjectTeacher
Tommy	6	TAN	Physics	H2	TAN
			Chemistry	H2	GOH
			General Studies	H1	WEE
Joe	7	GOH	Geography	H1	ROG
			French	H1	HEN
Samir	6	TAN	Computer Science	H2	VAR
			Chemistry	H2	GOH
			Maths	H2	ZEN
			General Studies	H2	WEE

(a)

(i) Explain why the table is not in First Normal Form (1NF). [1]

.

(ii) Explain your answer by referring to the data. [1]

(b) The design is changed to:

```
Student (StudentName, TutorGroup, Tutor)
StudentSubjectChoices (StudentName, Subject, Level,
SubjectTeacher)
```

Using the data given in the first attempt table, show how this data is now stored in the revised table designs. [3]

Table: `Student`

StudentName	TutorGroup	Tutor

Table: `StudentSubjectChoices`

StudentName	Subject	Level	SubjectTeacher

(c) **(i)** Explain what is meant by a primary key. [2]

(ii) A student is **not** allowed to choose the same subject at **H2** Level and **H1**. What is the primary key of table `StudentSubjectChoices`? [1]

(iii) There is a relationship between tables `Student` **and** `StudentSubjectChoices`. Explain how the relationship is **established** using a primary key and foreign key. [2]

(d) The design of table `StudentSubjectChoices` is:

```
StudentSubjectChoices (StudentName, Subject, Level,
SubjectTeacher)
```

Explain why this table is **not** in Second Normal Form (2NF). [2]

(e) The design of table `Student` is:

```
Student (StudentName, TutorGroup, Tutor)
```

Explain why this table is **not** in Third Normal Form (3NF). [2]

(f) Explain the conditions under which simultaneous access to a database could cause a problem, and how this could be dealt with. [3]

End of Paper