National Qualifications 2017

X716/77/11 Computing Science

TUESDAY, 16 MAY
1:00 PM–3:00 PM

Total marks — 60

Attempt ALL questions.

Write your answers clearly in the answer booklet provided. In the answer booklet you must clearly identify the question number you are attempting.

Use **blue** or **black** ink.

Before leaving the examination room you must give your answer booklet to the Invigilator; if you do not, you may lose all the marks for this paper.

MARKS

Total marks — 60

Attempt ALL questions

1. An annual shot put competition is to take place in October.

The program used each year to record and process competition data is being updated to allow participants to compete in three different categories: Senior, Adult or Junior.

(a) The organisers have reviewed feedback about the existing program and essential changes have been identified. Part of the outline plan for the development work is described below.

> The development aims to begin on Monday September 5, 2017, by identifying:
>
> - the user requirements
> - the business requirements
> - the scope and boundaries of the development
>
> These three tasks can be completed concurrently. Identification of the user requirements and the business requirements will each take 5 days whereas identification of the scope and boundaries will only take two days.
>
> The functional requirements can only be identified once the previous three tasks have been completed. This work will take three days.

Create a Gantt chart to schedule the first four development tasks outlined in the description above. **2**

(b) The aim of the competition is to throw the shot as far as possible. The competition has six rounds. The distance thrown in each round is recorded in metres. The winner in each category (Senior, Adult or Junior) is the competitor who throws the shot the furthest.

Each competitor is given a unique ID. The competition data collected for competitor 12 is shown below.

CompetitorID	Name	Category	Throw 1	Throw 2	Throw 3	Throw 4	Throw 5	Throw 6
12	Davidson L	Senior	11·0	14·3	14·8	17·4	16·7	17·1

Explain why an array of records would be more suitable than a 2-D array to store this competitor data. **2**

1. (continued)

(c) To retrieve the record for a particular competitor, the binary search algorithm will be applied to an array of records.

The structure of the array of records used to store the competition data for the Senior category is defined using the following code:

```
RECORD Competitor IS { INTEGER competitorID, STRING name, STRING category,
ARRAY OF REAL throw }
DECLARE seniorCompetitor AS ARRAY OF Competitor INITIALLY [ ]
```

The code used to retrieve the correct competitor record is shown below.

```
Line 1   PROCEDURE search(ARRAY OF Competitor seniorCompetitor, INTEGER
         requiredID, INTEGER low, INTEGER high)
Line 2       IF low > high THEN
Line 3           RETURN -1
Line 4       END IF
Line 5       SET mid TO (low + high)/2
Line 6       IF (seniorCompetitor[mid] = requiredID) THEN
Line 7           RETURN mid
Line 8       ELSE IF (seniorCompetitor[mid] < requiredID) THEN
Line 9           RETURN search(seniorCompetitor, requiredID, mid+1, high)
Line 10      ELSE
Line 11          RETURN search(seniorCompetitor, requiredID, low, mid-1)
Line 12      END IF
Line 13  END PROCEDURE
```

(i) Name the computational construct used in Line 9 and Line 11. **1**

(ii) A call stack is a stack data structure used to keep track of the active subprograms.

Explain the role of a stack data structure in the execution of this search procedure. **2**

(iii) Explain the need for Lines 2 to 4. **2**

[Turn over

1. (continued)

(d) To work out the winner in the Senior category, the program calculates the longest throw for each competitor and places it, together with the matching competitorID, in an array of records called seniorLongest. This array of records is defined using the following code:

```
RECORD Longest IS { REAL LongestThrow, INTEGER CompetitorID }
DECLARE seniorLongest AS ARRAY OF Longest INITIALLY [ ]
```

The program then uses a bubble sort algorithm to arrange the contents of the seniorLongest array into descending order of longest throw. A section of the code used for this purpose is shown below.

```
Line 50     FOR counter FROM 0 TO LengthOfArray-2 DO
Line 51         IF <adjacent items of seniorLongest array are in the wrong order>
                THEN
Line 52             swap (seniorLongest, counter)
Line 53         END IF
Line 54     END FOR
        …
Line 150    PROCEDURE swap(ARRAY OF Longest sLongest, INTEGER count)
Line 151        SET temp TO sLongest[count]
Line 152        ……
Line 153        ……
Line 154    END PROCEDURE
```

Using a programming language with which you are familiar, write the code needed to complete Line 152 and Line 153 of the swap procedure. Your code should make use of the data structure and variables stated. **2**

(e) A limit is set on the number of competitors who can participate in each category (Senior, Adult or Junior). Once this limit has been reached, a waiting list will be used to store the name and category of each additional person wishing to take part. If a competitor withdraws from the competition, they will be replaced by the next suitable person on the waiting list.

A queue data structure is considered for the waiting list.

Explain why a queue data structure would **not** be appropriate for this purpose. **2**

(f) The organisers of the shot put competition use cloud-based services to record and process competition data.

Describe one measure that providers of cloud-based services could take to ensure that their computer systems are environmentally friendly. **2**

[Turn over for next question

DO NOT WRITE ON THIS PAGE

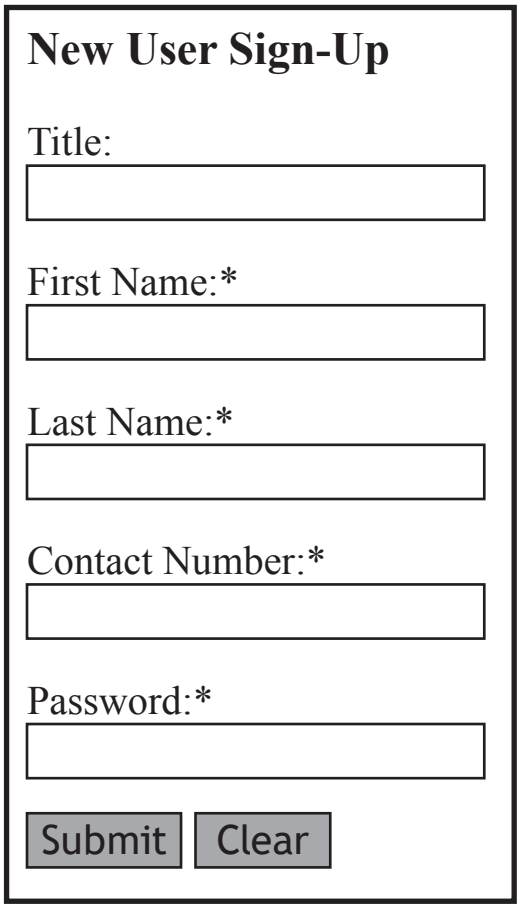
MARKS

2. A restaurant chain wants a new app which will allow registered users to make reservations for restaurants that are part of the chain. All users must register before they can use the app for the first time.

A development team is asked to create the new app.

(a) The developers begin by designing an SQL table called RegUser. This table will be used to store the details of all of the registered users. The table will be stored in a relational database called RestaurantApp.

The RegUser table will store a userID which is automatically created, the user's title, first name and last name, a phone number for contact purposes and a password. Users will be asked to provide all of these details when they register. This data will be captured using the HTML form shown below.



The userID will be generated automatically by the database server when a new record is added to the table. Data values must be provided for all fields marked *.

Copy and complete the two lines of the data dictionary below to show the structure of the SQL table called RegUser for the fields listed. Your data dictionary should indicate appropriate SQL data types. **3**

Field	Type/Size	Key	Constraints/Validation
userID			
title			

2. (continued)

(b) The HTML script used to generate the registration form is provided below.

```
<form action="signmeup.php" method="post">
    <h3>New User Sign-Up</h3>
    Title: <br><input name="Item1" type="text" /><br>
    First Name:* <br><input name="Item2" type="text" /><br>
    Last Name:* <br><input name="Item3" type="text" /><br>
    Contact Number:* <br><input name="Item4" type="text" /><br>
    Password:* <br><input name="Item5" type="password" /><br>
    <input name="Submit" type="submit" value="Submit" />
    <input name="Clear" type="reset" value="Clear" />
</form>
```

The submit button on the form is used to submit the registration data to a server-side script. This script is used to connect to the database server and add the registration details to the RegUser table of the RestaurantApp database.

The RegUser table has 6 fields: userID, title, firstName, lastName, contactNo and password.

The connection details used are:

server name:	sn001
user:	anon001
password:	ap001

Using a server-side scripting language with which you are familiar, write the script used to connect to the database server. The script should:

(i) assign the registration details to server-side variables **1**

(ii) create a connection to the database server **2**

(iii) execute the SQL query used to add the registration details to the RegUser table. **3**

[Turn over

2. (continued)

(c) Details of restaurant reservations will be stored in the relational database in a separate table called Reservation. Part of the Reservation table is shown below.

bookingID	restaurant ID	regUserID	date	time	sizeofgroup
123456	1	1	01/07/2017	18:00	4
123457	1	2	01/07/2017	18:00	2
123458	2	4	01/07/2017	18:30	2
123459	3	9	01/07/2017	18:15	5
123460	2	8	01/07/2017	19:30	2
123461	1	5	01/07/2017	19:30	6

Relationships
RegUser: Reservation 1:M

The manager of each restaurant needs a list of reservations made for the day.

Write an SQL statement to display a list of all reservations for restaurant 1 on the 1st of July 2017. The list should show the first name, last name and title for each reservation, the time of the reservation and the size of each group. The reservation details should be displayed so that the earliest reservation for the restaurant is listed first. **3**

(d) At the end of each month, the data captured from the app is exported to a program which is used by management to make decisions about stock levels.

Within the program, a function is used to arrange the restaurant details by overall monthly sales so that restaurants in the chain are listed in ascending order. This function makes use of the quicksort algorithm below.

1. set the pivot to the first element in the array
2. partition the array into 3 separate sub-arrays:

   LEFT sub-array: contains all elements less than the pivot

   PIVOT sub-array: contains the pivot itself

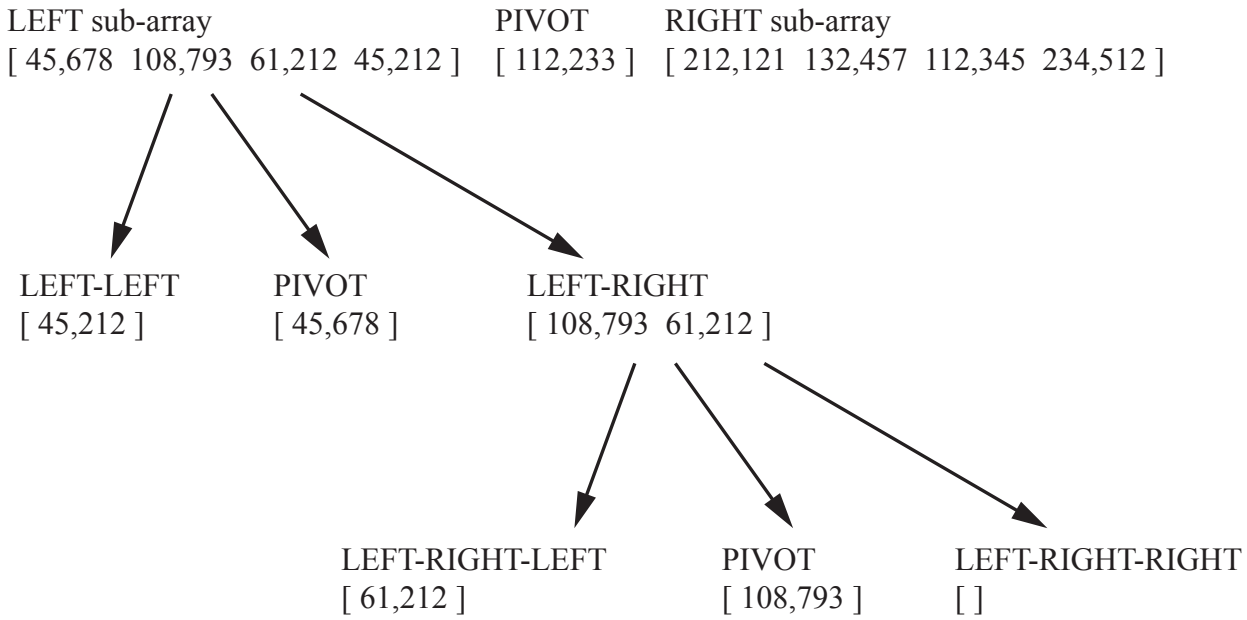
   RIGHT sub-array: contains all elements greater than or equal to the pivot
3. repeatedly apply the quicksort algorithm to the LEFT and the RIGHT sub-arrays until each sub-array contains exactly one or zero elements
4. concatenate the sub-arrays from left to right, ignoring any zero-element sub-arrays

2. (d) (continued)

The following array of nine elements is used to test the quicksort algorithm:

[112,233 212,121 132,457 45,678 108,793 61,212 45,212 112,345 234,512]

Steps 1 and 2 of the algorithm are applied to the array for the first time, and at step 3, the quicksort algorithm is applied to the LEFT sub-array. The following situation exists:



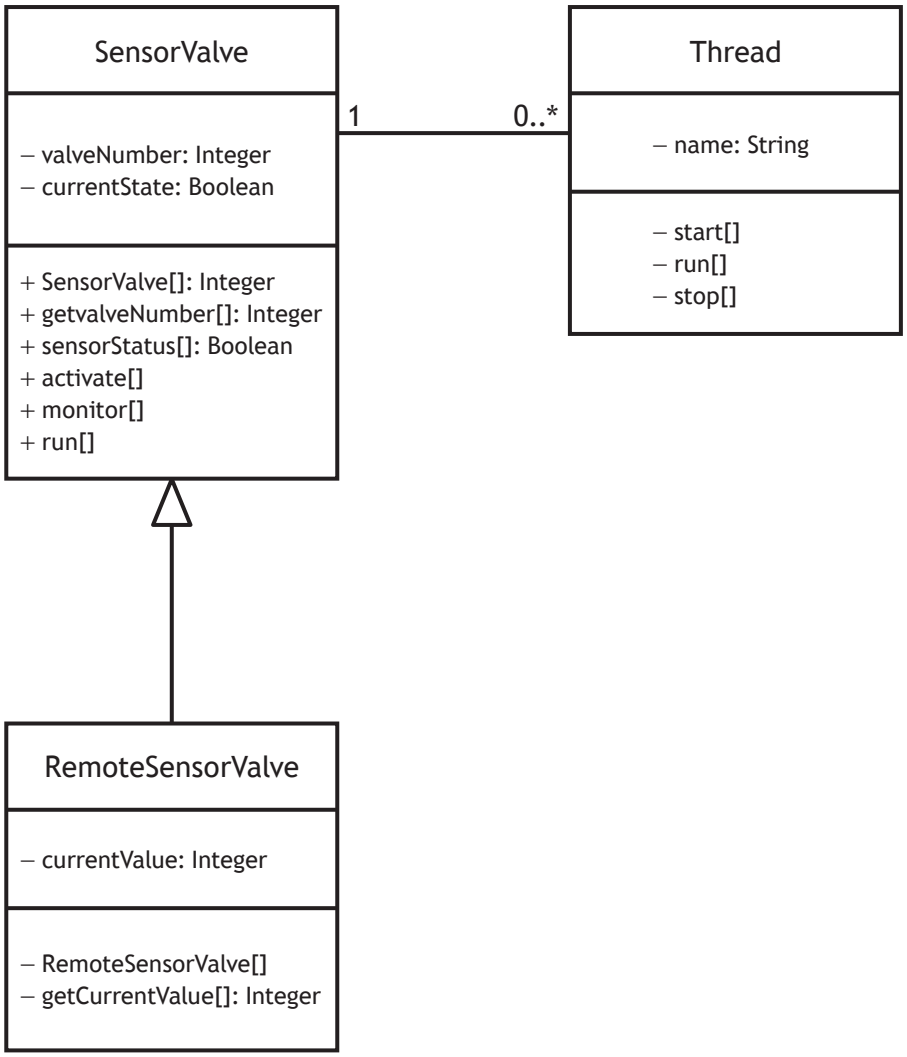
Show how the quicksort algorithm would be applied to the RIGHT sub-array. **3**

[Turn over

MARKS

3. Imogen is a programmer for an engineering company that manufactures sensors. She is creating a new application that is used to ensure that the sensors are working correctly.

(a) The application stores data from SensorValve and RemoteSensorValve objects. This is shown in the UML class diagram below.



(i) Using the class diagram above, describe the relationship between the SensorValve class and the RemoteSensorValve class. **2**

(ii) Using an object-oriented programming language with which you are familiar, write a Class definition for a RemoteSensorValve object. Your answer should include the property and methods indicated in the UML class diagram above. **3**

(iii) Write code to show how your Class in part(ii) could be instantiated. **1**

3. (continued)

(b) A 1-D array of RemoteSensorValve objects is used to store the properties and methods of 10,000 RemoteSensorValve objects.

Imogen writes a procedure to sort this array into ascending order based on the currentValue reading of each object.

She decides to use the insertion sort algorithm. Imogen uses the procedure call below to activate the insertion sort procedure:

```
insertionSort( remoteSensorValveArray[] )
```

Use pseudocode to design the insertionSort procedure used to sort the array of objects into ascending order. **5**

(c) Once it has been completed, the application will be a large, complex piece of software that will incorporate Imogen's procedure to sort the array of objects.

Explain the relevance of component testing in this situation and describe how it could be carried out. **2**

(d) Sensors manufactured by the engineering company are used by AirFlappe Airlines in their fleet of 700 aircraft.

These sensors provide data for real-time analysis of engine performance during a flight. On a daily basis, each aircraft generates approximately 12·5 million terabytes of data.

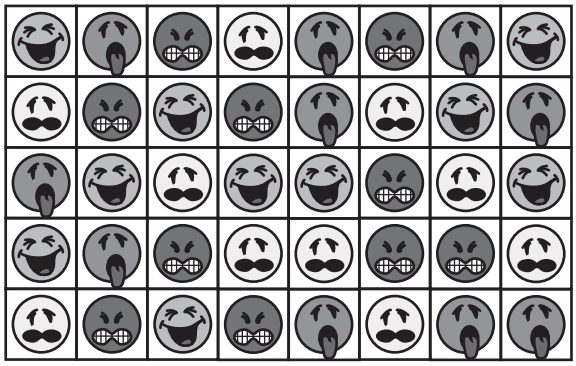
Describe one implication of this example of 'big data' analytics for AirFlappe Airlines. **2**

[Turn over

MARKS

4. Ali has been asked by a local primary school to create a program which allows young children to express their feelings. The program will be based on a pairs matching game.

When the program starts, 40 images representing the four most common feelings will be displayed and the child asked to match any two of the images.



(a) At the start of the development, Ali needs to research the four most common feelings experienced by young children.

Explain how a user survey could be used for this purpose. **2**

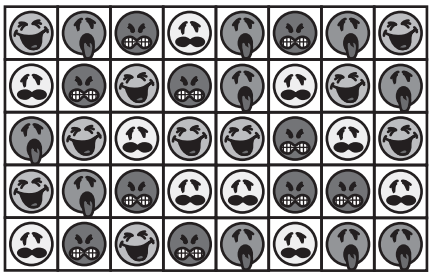
Ali wants to display images of each of the four feelings randomly when the program starts and decides to allocate each image a number between 1 and 4. The program will select a number between 1 and 4 before placing each number into a 2-D array.

Below, the contents of a sample game board array are shown on the left and the matching set of images displayed when the program starts is shown on the right.

Sample game board array

1	2	3	4	2	3	2	1
4	3	1	3	2	4	1	2
2	1	4	1	1	3	4	1
1	2	3	4	4	3	3	4
4	3	1	3	2	4	2	2

Matching set of images displayed



(b) Ali wants each of the four images to appear **exactly 10 times**. The images should be displayed in the layout indicated above, with eight images in each of the five rows.

Use pseudocode to design a procedure which assigns a random number between one and four to each element in the 2-D game board array. Your design should indicate:

- the selection of 40 random numbers between 1 and 4
- a check to ensure that each number appears exactly 10 times
- the storage of the numbers selected in the 2-D array **5**

You should assume that the game board array will be passed to the procedure as a parameter.

4. (continued)

Having matched a pair of images, the child is then required to complete a sentence about their feelings. The program allows the child to enter one or more sentence endings. Any sentence endings entered are added to the existing list of endings and stored so that they can be viewed the next time the game is played.

(c) Ali decides to use a linked list to store the sentence endings.

He plans to write the sentence endings held in the linked list to a sequential file called happyFeelings.txt.

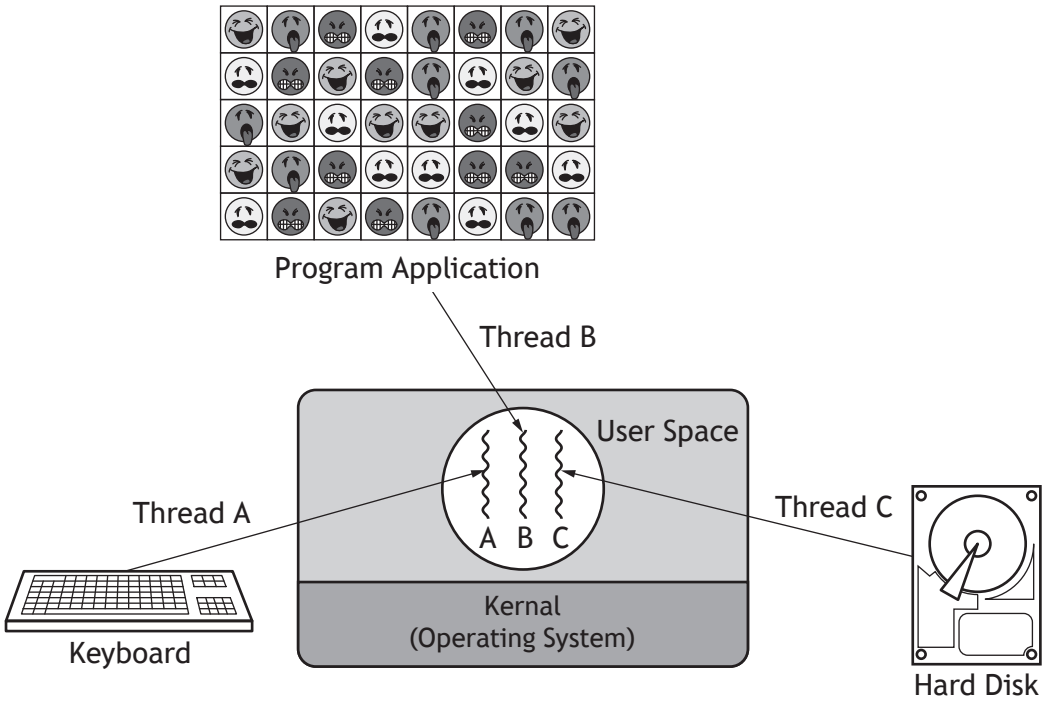
Using a programming language with which you are familiar, create a procedure to write the contents of the linked list to the happyFeelings.txt file.

You should state any assumptions you make about how the linked list is implemented within the procedure. **4**

[Turn over for next question

4. (continued)

(d) The image below demonstrates how Ali has written his program using concurrent programming techniques.



The program uses multiple threads.

Thread A is used to interact with the keyboard, Thread B is used to manage the program application and Thread C is used to automatically backup and save the program data every two minutes.

Explain why the use of multiple threads improves the performance of the program and gives a better user experience. **2**

(e) As a self-employed developer, Ali is concerned about his intellectual property rights in relation to the program.

Describe one form of automatic protection Ali receives when creating the program. **2**

[END OF QUESTION PAPER]

[BLANK PAGE]

DO NOT WRITE ON THIS PAGE

[BLANK PAGE]

DO NOT WRITE ON THIS PAGE