2.2.3 Files

1 Anna wants to find out about her fellow students' reading habits. It will be part of her Literature coursework.

She will ask questions online, so starts by designing a screen layout. The first four questions will ask for:

- student's first name
- date of birth
- type of book they prefer (printed, audio-book or e-book)
- whether student reads novels (yes/no)

(f) The records will be held in a serial file.

Give three statements from a high-level language that may be used for the file handling and explain what each does.

[6]

1 Anna wants to find out about her fellow students' sporting activities. It will be part of her Sports Studies coursework.

She will ask questions online, so starts by designing a screen layout. The first four questions will ask for:

• student's first name

• age (16,17,18 or 19)

• favourite sport

• whether student is a member of a sports club (yes/no)

(f) The records will be held in a direct access file.

Give four statements from a high-level programming language that may be used for the file handling and explain what each does. **[8]**

4 Super Bikes will store the data in files.

(a) One file will store the following data:

- bike registration
- purchase cost of bike
- insurance rating (A, B or C only)
- whether or not service is due

(c) (ii) Bike records are stored in a file SuperBikes.

Write the code to read every bike record, count the number of bikes where service is due and output the result. You should assume that the file is not already in use. **[9]**

May/June 2013. P21/22

1 Meena wants to develop a program to keep a record of her coursework assignments.

She will want to enter, sort and print out data.

She decides to modularise the solution.

One way of storing her data will be to use a file of records.
Each record will contain at least the following data:

Data	Identifier	Description of input data
subject	Subject	Name of the subject, for example Physics
title	Title	Title of assignment
date set	DateSet	Format DDMMYYYY
hand-in date	HandInDate	Format DDMMYYYY
marked?	IsMarked	Y or N
date returned	DateReturned	Format DDMMYYYY
mark	Mark	Range 0 to 100

(c) Describe what the function EOF() does when used in a program. **[2]**

(d) Meena creates a sequential file, MyAssignments, of Assignment records.

Using pseudocode write the algorithm to search this file for the first Physics assignment.

OPENFILE MyAssignments FOR OUTPUT

CLOSEFILE MyAssignments **[4]**

May/June 2013. P23

1 Meena wants to develop a program to keep a record of her examination results.

She will want to enter, sort and print out data which is stored as a file of records.

Each record will contain at least the following data:

- subject
- examination title
- level
- date sat
- mark

An example of an examination title is 'General Certificate of Education'.

The DateSat field will contain only the month and year that the examination was taken.

For all subjects the mark is between 0 and 100 inclusive.

The level is 'O' or 'A'.

(e) Describe what the function EOF() does when used in a program.   **[2]**

Oct/Nov 2013. P21

1 The Computing Department has a problem keeping track of its teaching resources. Juan, a student, has been asked to design and program a solution as his computing project. It will be the first large problem he has worked on.

He intends to write one large program that follows the process right through. His teacher tells him to break the problem into smaller parts.

(e) Juan will need the program to hold large amounts of data. His design will include reading the data from the sequentially-organised resource file into several arrays.

Describe **three** differences between the features of an array and a sequential file.   **[6]**

Oct/Nov 2014.P21/P22

2 Ali sets up user IDs and passwords for his program.

When the user types in their user ID, the program looks up the stored password for this user ID.

The stored password is the encrypted version of the user's password.

(b) Ali uses a sequential file of records to store the user IDs and encrypted passwords.

When a user types in their User ID, the program calls a function, `FindPassword`, with parameter `ThisUserID`.

The function searches each record in the file for `ThisUserID` and returns the encrypted password.

If `ThisUserID` is not stored in the file, the function returns an error code.

Complete the pseudocode:

2.2.3 Files

```
FUNCTION FindPassword(ThisUserID : STRING) RETURNS ................

   DECLARE Found : BOOLEAN

   OPENFILE FOR INPUT // for reading

   Found ← FALSE

   WHILE ........................................................

      FILEREAD next record

      IF .......................................................

         THEN

            ...................................................

      ENDIF

ENDWHILE

IF .........................................................

   THEN

      ...................................................

   ELSE

      ...................................................

   ENDIF

   CLOSEFILE

ENDFUNCTION                                                    [8]
```

2.2.3 Files

1 (a) Sheena wants to develop a program to keep details of the books she has read.
One way of storing this data will be to use records. Each record will contain the fields shown in the table below.

Field	Identifier	Data type	Example of input data	Field size (in bytes)
Title	Title		How to solve it	
Author	Author		G Polya	
International Standard Book Number	ISBN		9780691119663	
Number of pages	NumberOfPages		253	
Price ($)	BookPrice		12.50	
Date started to read book	DateStarted		28032012	
Date finished reading book	DateFinished		17052012	
Paperback?	IsPaperback		TRUE	
Rating (range 0 to 5)	Rating		4	

(c) Sheena declares an array of records to store the book data:
`DECLARE Book ARRAY[1 : 100] OF BookRecordType`
At the beginning, the array elements will not contain book data.
Explain how the array should be initialised.

(d) Ravi has entered some course data. His program saved the array records to a serial file
`CourseData.DAT`. Only those array elements containing entered course details were saved
to the file.
A program is to be written to read the data from `CourseData.DAT` into consecutive locations
of the array `Course`.

(i)   Explain what the function `EOF()` does when used in a program.   **[2]**

Ravi wants to write another module in his program. The new module will:
- read data from `CourseData.DAT`
- write these data into consecutive elements of the array `Course`

(ii)   Write pseudocode to read the data from the file `CourseData.DAT` into the array `Course`.   **[6]**

1 (f) Sheena wants to read previously saved data from the file.
Here is part of the pseudocode she wrote:
```
WHILE NOT EOF(BookData.DAT)
:
:
ENDWHILE
```
Explain what the function `EOF()` does when used in a program.   **[2]**

2.2.3 Files

Computer Science (9608)

5 A company creates two new websites, Site X and Site Y, for selling bicycles.

Various programs are to be written to process the sales data.

These programs will use data about daily sales made from Site X (using variable `SalesX`) and Site Y (using variable `SalesY`).

Data for the first 28 days is shown below.

	SalesDate	SalesX	SalesY
1	03/06/2015	0	1
2	04/06/2015	1	2
3	05/06/2015	3	8
4	06/06/2015	0	0
5	07/06/2015	4	6
6	08/06/2015	4	4
7	09/06/2015	5	9
8	10/06/2015	11	9
9	11/06/2015	4	1
...			
28	01/07/2015	14	8

(d) The company decides to offer a discount on selected dates. A program is written to indicate the dates on which a discount is offered.

The program creates a text file, DISCOUNT_DATES (with data as shown), for a number of consecutive dates.

2.2.3 Files

```
03/06/2015 TRUE
04/06/2015 FALSE
05/06/2015 FALSE
06/06/2015 FALSE
07/06/2015 FALSE
08/06/2015 FALSE
09/06/2015 FALSE
10/06/2015 TRUE
11/06/2015 FALSE

01/07/2015 FALSE
```

Each date and discount indicator is separated by a single character.

The discount indicators are:

- FALSE – indicates a date on which no discount is offered
- TRUE – indicates a date on which a discount is offered

A programming language has the built-in function `CONCAT` defined as follows:

```
CONCAT(String1 : STRING, String2 : STRING [, String3 : STRING] )
                                                RETURNS STRING
For example:       CONCAT("San", "Francisco") returns "SanFrancisco"
                   CONCAT("New", "York", "City") returns "NewYorkCity"

The use of the square brackets indicates that the parameter is optional.
```

The following incomplete pseudocode creates the text file DISCOUNT_DATES.

Complete the pseudocode.

2.2.3 Files

```
OPENFILE "DISCOUNT_DATES" FOR ............................................................................................................

INPUT ............................................................................................................................................

WHILE NextDate <>"XXX"

    INPUT Discount

    ....................................... = CONCAT(NextDate, " ", Discount)

    WRITEFILE "DISCOUNT_DATES", NextLine

    INPUT NextDate

......................................................................................................................................................

OUTPUT "File now created"

CLOSEFILE                                                                                                    [4]
```

May/June 2015.P23

4 A programming language has the built-in function CONCAT defined as follows:

```
CONCAT(String1 : STRING, String2 : STRING [, String3 : STRING] )  RETURNS STRING
For example:  CONCAT("San", "Francisco") returns "SanFrancisco"
              CONCAT("New", "York", "City") returns "NewYorkCity"

The use of the square brackets indicates that the parameter is optional.
```

(b) A country has a number of banks. There are cash dispensers all over the country. Each bank is responsible for a number of dispensers.

- banks have a three digit code in the range 001 – 999
- each dispenser has a five digit code in the range 00001 – 99999

A text file, DISPENSERS, is to be created.

It has one line of text for each dispenser. For example: 00342▼007.

This line in the file is the data for dispenser 00342 which belongs to bank 007.

Incomplete pseudocode follows for the creation of the file DISPENSERS.

For the creation of the file, data is entered by the user at the keyboard.

(i) Complete the pseudocode.

2.2.3 Files

```
OPENFILE ............................................................................................ FOR WRITE
........................................................................................................................................

    OUTPUT "Enter dispenser code (XXXXX to end)"

    INPUT DispenserCode

    IF DispenserCode <> "XXXXX"

        THEN

            OUTPUT "Enter bank code"

            INPUT BankCode

            LineString CONCAT(................................................................ , "▼", BankCode)

            // now write the new line to the file

            ....................................................................................................................

        ENDIF

UNTIL   ........................................................................................................................................

........................................................................................................................................

OUTPUT "DISPENSERS file now created"                                          [6]
```

(iv) Give the 'file mode' available in the programming language which will be used to address this issue.   **[1]**

2.2.3 Files

(c) The complete data file is created with the structure shown.

A new program is to be written to search the file.

The program will:

- input a bank code
- output a list of all the dispensers which belong to this bank
- output the total number of dispensers for this bank

An example of a run of the program is shown

```
Enter bank code 007
00001
00011
00022
00026
00027

There are 5 dispensers for this bank
```

```
00001▼007
00002▼001
00003▼002
00004▼003
00005▼101
00006▼004
00007▼004
∫
00024▼002
00025▼003
00026▼007
00027▼007
00028▼102
∫
99867▼013
```

Write the program code. Do not attempt to include any validation checks.

Visual Basic and Pascal: You should include the declaration statements for variables.

Python: You should show a comment statement for each variable used with its data type.

[10]

May/ June 2016. P21/P22

5 Toni has a large collection of jazz CDs that are stored in different places. She wants to record where the CDs are stored. She decides to write a program to do this.
The program must store the data in a file, `MyMusic`.
(a) (i) Why is a file needed? [1]
(ii) `MyMusic` is a text file with the data for each CD as one line of text.
Data for a typical CD are:
Title: Kind of Green
Artist: Miles Coltrane
Location: Rack1-5
The line will be formed by concatenating the three data items.
For the example above, the line stored will be:
`Kind of GreenMiles ColtraneRack1-5`

2.2.3 Files

Describe a problem that might occur when organising the data in this way.
Describe a possible solution. **[4]**

(b) Toni must input the data into the file for all of her CDs.

A procedure, `InputData`, is needed to do this.

Toni designs the procedure and chooses the following identifiers:

Identifier	Data type
CDTitle	STRING
CDArtist	STRING
CDLocation	STRING

The procedure repeatedly performs the following steps:
- input a CD title (A rogue value of "##" is to be used to end the input)
- input the artist
- input the location
- create the text line
- write the text line to the file

When the rogue value is encountered the file is closed.

Write **program code** for the procedure `InputData`.

Visual Basic and Pascal: You should include declaration statements for variables.

Python: You should show a comment statement for each variable used with its data type. **[8]**

May/ June 2016. P23

5 Claudia stores her large collection of music CDs in different places. Claudia wants to record where she stores each CD. She decides to write a program to do this.

Data items for a typical CD are:

Title: Kind of Green

Artist: Miles Coltrane

Location: Rack3-23

The data is to be stored in a text file, `MyMusic`. Each line of the text file will be a string, formed by concatenating the three data items.

Before concatenation, the title and artist will each be made into a fixed-length string of 40 characters. Space characters may need to be added to each data item.

The location is always 8 characters long.

(a) (i) Explain the benefit of making the stored data into fixed-length strings.

State a drawback of this file design. **[3]**

(ii) When Claudia buys a new CD, the CD data must be added to the existing file, `MyMusic`.

She has written a procedure in pseudocode. This has the following file-handling statements:

```
OPENFILE "MyMusic" FOR WRITE
WRITEFILE "MyMusic", OutputString
CLOSEFILE "MyMusic"
```

There is a problem with the logic of this pseudocode.

State the problem.

2.2.3 Files

Identify the effect it will have if the final code is implemented in this way.
Give a possible solution.   **[3]**
(b) Claudia needs to output a list of all the CDs in a particular location.
She designs a procedure, `OutputLocationList`, to do this. She also chooses the following identifiers:

Identifier	Data type
CDTitle	STRING
CDArtist	STRING
CDLocation	STRING

The procedure will:
- prompt for the name of the location
- input the location (such as "Rack3-23")
- search the file for all CDs at this location
- output the title and artist of each CD found
- output the total number of CDs found at that location (such as "17 CDs found")

Write **program code** for the procedure `OutputLocationList`.
Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type.   **[10]**

Oct/Nov 2016. P21/P23

4 A company employs Ahmed as a programmer.
(c) The company maintains a file of product data. Ahmed is to write a program to add a new product and search for a product based on the structure diagram shown:
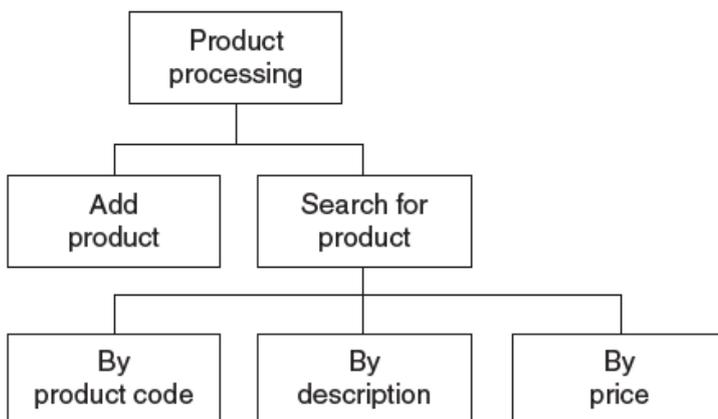
The program records the following data for each product:
- product code
- product description
- product retail price

The text file `PRODUCTS` stores each data item on a separate line, as shown below:

2.2.3 Files

File **PRODUCTS**

0198
Plums(10kg)
11.50
0202
Onions(20kg)
10.00
⌇
0376
Mango chutney(1kg)
02.99
⌇
0014
Mango(10kg)
12.75

The program uses the variables shown in the identifier table.

Identifier	Data type	Description
PRODUCTS	TEXT FILE	Storing the code, description and retail price for all current products
PCode	ARRAY[1:1000] OF STRING	Array storing the product codes
PDescription	ARRAY[1:1000] OF STRING	Array storing the product descriptions
PRetailPrice	ARRAY[1:1000] OF REAL	Array storing the product retail prices
i	INTEGER	Array index used by all three arrays

(i) The first operation of the program is to read all the product data held in file PRODUCTS and write them into the three 1D arrays.

Complete the pseudocode below.

```
OPEN ......................................................................................................................................................................
i ← 1
WHILE ....................................................................................................................................................................
        READFILE ("PRODUCTS",...................................................................................................................)
        READFILE ("PRODUCTS",...................................................................................................................)
        READFILE ("PRODUCTS",...................................................................................................................)
        ..............................................................................................................................................................
        ..............................................................................................................................................................
```

2.2.3 Files

....

```
ENDWHILE
CLOSE "PRODUCTS"
OUTPUT "Product file contents written to arrays"
```

[5]

File **PRODUCTS**

0198 Plums(10kg)	11.50
0202 Onions(20kg)	10.00
0376 Mango chutney(1kg)	02.99
0014 Mango(10kg)	12.75

When Ahmed designed the PRODUCTS file,
he considered the alternative file structure shown opposite.
It stores one product per line in the text file.

(ii) State **one** benefit and **one** drawback of this file design.   **[2]**

(d) To code the 'Search by product code' procedure, Ahmed draws a structure chart showing the different stages.
The procedure uses the variables shown in the identifier table.

| Identifier	Data type	Description
SearchCode	STRING	Product code input by the user
ThisIndex	INTEGER	Array index position for the corresponding product
ThisDescription	STRING	Product description found
ThisRetailPrice	REAL	Product retail price found

You can assume that before the procedure is run, all the product data is read from file PRODUCTS and then stored in three 1D arrays as described in **part (c)(i)**.
Label the structure chart to show the input(s) and output(s).
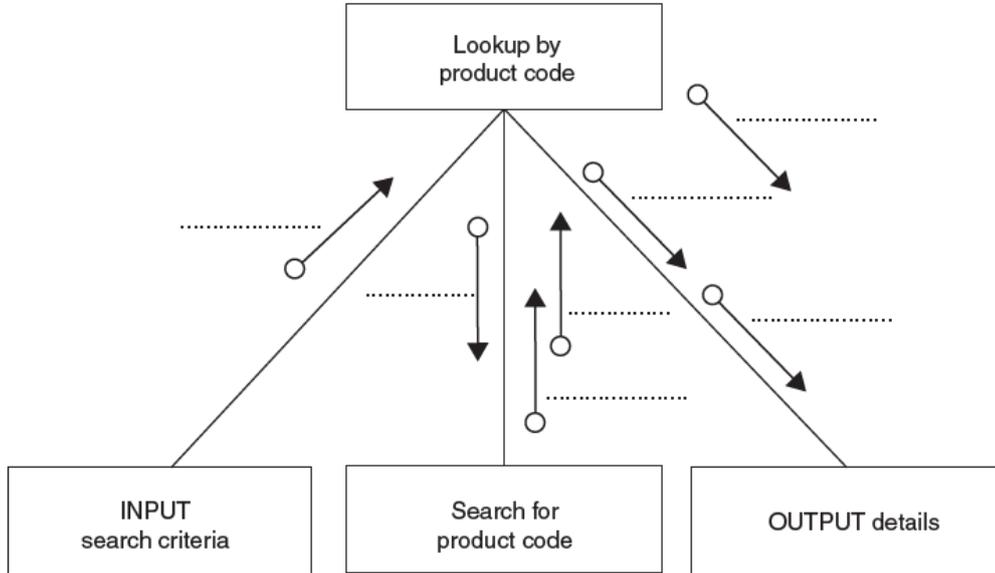
[4]

(e) A first attempt was made at writing the 'Search for product code' module.
Ahmed designs this as a function `ProductCodeSearch`.
The function returns an integer value as follows:
- if the product code is found, it returns the index position of the 1D array `PCode` being searched
- if the product code is not found, the function returns -1

Write **program code** for function `ProductCodeSearch`.
Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type. **[6]**

Oct/Nov 2016. P22

5 A team keeps a record of the scores made by each of their eight players in a number of games.
The data in the two tables below shows:
- the scores of the eight players after twenty games
- the eight player names.

2.2.3 Files

	1	2	3		8
1	12	17	67		31
2	35	82	44		29
3	61	39	80		17
4	81	103	21		11
5	56	0	98		4
...					
19	45	6	81		77
20	12	11	3		6

1	Vorma
2	Ravi
3	Chada
4	Nigam
5	Bahri
6	Smith
7	Goyal
8	Lata

The team wants a computer program to input and record the player data.

(b) The player names are permanently stored in a text file NAMES.TXT, with one name on each line. The player names will be read by the program and stored in a 1D array.

The design given in **part (a)** will be expanded so that the user is prompted for the player name instead of the player number. Step 02 now becomes:

02.1 Read the player names from file NAMES.TXT into the array PlayerName

02.2 INPUT ThisPlayerName

02.3 Search the PlayerName array for ThisPlayerName to find the PlayerNumber

- **(i)** State the computing term for the expansion of one or more steps in the original design. **[1]**
- **(ii)** Write the **program code** for step 02.1
  Visual Basic and Pascal: You should include the declaration statements for variables.
  Python: You should show a comment statement for each variable used with its data type. **[4]**
- **(iii)** Program code is to be designed and written for step 02.3
  The program will use these identifiers:

2.2.3 Files

Identifier	Data type	Description
PlayerName	ARRAY[1 : 8] OF STRING	Stores the player names (read from the file)
ThisPlayerName	STRING	Input by the user (step 02.2)
Found	BOOLEAN	Flags when ThisPlayerName is found when searching the PlayerName array
i	INTEGER	Array index

Write **program code** to carry out the linear search for step 02.3
There is no requirement to declare or comment on variables used. **[4]**

May/June 2018. P21

5 A golf club holds information about its members. When a member completes a round of golf, their score is stored along with their membership number and the date of the round.
(a) Explain why the club stores these data in a file rather than an array. **[1]**
(c) The information is stored in a text file, ScoreDetails.txt. The format of each line of the text file is as follows:
<MembershipNumber><Date><Score>
- MembershipNumber is a four-digit numeric string.
- Date is a six-digit numeric string in the format DDMMYY
- Score is a two-digit numeric string in the range "50" to "99".

A procedure, AddNewScores(), is being developed. This will allow the user to enter scores for several members on a particular date.
The procedure, AddNewScores(), will perform the following actions:
1. Prompt for the date of the scores.
2. Input the date of the scores.
3. Prompt for the membership number.
4. Input the membership number.
5. If the membership number is an empty string then end the procedure.
6. Prompt for the score.
7. Input the score.
8. Validate the score.
9. If the validation fails then repeat from step 6.
10. Form a text string from the data and write this to the ScoreDetails.txt file.
11. Repeat from step 3.

Write program code for the AddNewScores() procedure.
Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type. **[9]**