

File Handling

- 1 Each line of a text file contains data organised into fixed-length fields as shown:

<Field 1><Field 2><Field 3>

An algorithm is required to search for the first instance of a given value of Field 2 and, if found, to output the corresponding values for Field 1 and Field 3.

Describe the algorithm needed.

Do **not** include pseudocode statements in your answer.

[6]

- 2 (a) A concert venue uses a program to calculate admission prices and store information about ticket sales.

A number of arrays are used to store data. The computer is switched off overnight and data has to be input again at the start of each day before any tickets can be sold. This process is very time consuming.

- (i) Explain how the program could use text files to speed up the process.

.....

.....

.....

.....

.....

..... [2]

- (ii) State the characteristic of text files that allow them to be used as explained in **part (a)(i)**.

.....

..... [1]

- (iii) Information about ticket sales will be stored as a booking. The booking requires the following data:

- name of person booking
- number of people in the group (for example a family ticket or a school party)
- event type.

Suggest how data relating to each booking may be stored in a text file.

.....

.....

.....

..... [2]

(b) A procedure `Preview()` will:

- take the name of a text file as a parameter
- output a warning message if the file is empty
- otherwise output the first five lines from the file (or as many lines as there are in the file if this number is less than five).

Write pseudocode for the procedure `Preview()`.

[5]

- 3 The following data items will be recorded each time a student successfully logs on to the school network:

Data item	Example data
Student ID	"CJL404"
Host ID	"Lib01"
Time and date	"08:30, June 01, 2021"

The Student ID is six characters long. The other two data items are of variable length.

A single string will be formed by concatenating the three data items. A separator character will need to be inserted between items two and three.

For example:

"CJL404Lib01<separator>08:30, June 01, 2021"

Each string represents one log entry.

A programmer decides to store the concatenated strings in a 1D array `LogArray` that contains 2000 elements. Unused array elements will contain an empty string.

- (a) Suggest a suitable separator character **and** give a reason for your choice.

Character

Reason

..... [2]

- (b) The choice of data structure was made during one stage of the program development life cycle.

Identify this stage.

..... [1]

(c)

- take a Student ID as a parameter
- for each element in the array that matches the Student ID parameter:
 - add the value of the array element to the existing text file `LogFile`
 - assign an empty string to the array element
- count the number of lines added to the file
- return this count.

Write pseudocode for the function `LogEvents()`.

[illegible]

- 4 A company has several departments. Each department stores the name, email address and the status of each employee in that department in its own text file. All text files have the same format.

Employee details are stored as three separate data strings on three consecutive lines of the file. An example of the first six lines of one of the files is as follows:

File line	Comment
1	First employee name
2	First email address
3	First employee status
4	Second employee name
5	Second email address
6	Second employee status

A procedure `MakeNewFile()` will:

- take three parameters as strings:
 - an existing file name
 - a new file name
 - a search status value
- create a new text file using the new file name
- write all employee details to the new file where the employee status is **not** equal to the search status value
- count the number of sets of employee details that were in the original file
- count the number of sets of employee details that were written to the new file
- produce a summary output.

An example summary output is as follows:

```
File Marketing contained 54 employee details
52 employee sets of details were written to file NewMarketingList
```

- (a) Write pseudocode for the procedure `MakeNewFile()`.

.....

.....

.....

.....

.....

.....

.....

.....

[7]

(b) An alternative format could be used for storing the data.

A text file will still be used.

(i) Describe the alternative format.

.....
..... [1]

(ii) State **one** advantage **and one** disadvantage of the alternative format.

Advantage
.....
Disadvantage
..... [2]

- Use **structured English** to describe an algorithm that would prompt and input a new user name and output a message to indicate whether or not it is unique.

Use **only** the functions and operators described in the **Appendix** on pages 18–19.

[5]

6 The procedure `OutputLines()` outputs a number of lines from a text file.

An example of the use of the procedure is given by the following pseudocode:

```
CALL OutputLines(FileName, StartLine, NumberLines)
```

Parameter	Data type	Description
FileName	STRING	The name of the text file
StartLine	INTEGER	The number of the first line to be output
NumberLines	INTEGER	The number of lines to be output

The procedure is tested using the file `MyFile.txt` that contains 100 lines of text.

The procedure gives the expected result when called as follows:

```
CALL OutputLines("MyFile.txt", 1, 10)
```

- (a) The procedure is correctly called with three parameters of the appropriate data types, but the procedure does not give the expected result.

Give **three different** reasons why this might happen.

1

.....

2

.....

3

.....

(b) Write **program code** for the procedure `OutputLines()`.

Note: Parameter validation is **not** necessary.

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.

Programming language

Program code

This image shows a full page of primary-ruled paper. It features multiple sets of horizontal dashed lines spaced evenly down the page, providing a guide for handwriting practice. The paper is otherwise blank, with no text or other markings.

(c) Write **program code** for the module `LoadArrays()`.

The module description is repeated here for reference.

Module	Description
LoadArrays()	<ul style="list-style-type: none"> Values from the text file Backup.txt are stored in the two arrays The number of elements stored is returned

You should assume:

- each line of the file contains a string of the correct format and no validation checks are required
- there are no more than 10 000 lines in the file.

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.

Programming language

Program code

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

[8]

- 8** A text file, `MyCDs.txt`, stores information relating to a Compact Disc (CD) collection. Information about each CD is stored on three separate lines in the file as follows:

Line 1: <Artist Name>
Line 2: <CD Title>
Line 3: <Storage Location>

Information is stored as data strings.

A section of the file is shown:

File line	Data
100	"Green Floyd"
101	"Bowlful of Cereal"
102	"Shelf 4"
103	"Strolling Bones"
104	"Exile on Station Road"
105	"Box 12"

- (a) A program, `CDOrganiser`, will be written to manage the stored information. The program will consist of three modules: `AddCD`, `FindCD` and `RemoveCD`.

Give **three** reasons why it is good practice to construct the program using modules.

- 1
2
3

[3]

- (b) The module, `FindCD()`, will check whether a given CD exists in the collection. The module will be implemented as a function.

The function will:

- be called with two strings as parameters, representing the artist name and CD title
 - return a string that gives the storage location, or an empty string if the given CD has not been found.
-

Write **program code** for the function `FindCD()`.

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.

Programming language

Program code

[illegible]

- 9 Nadine is developing a program to store the ID and preferred name for each student in a school. For example, student Pradeep uses the preferred name “Prad”.

The program will:

1. prompt and input a valid user ID and a preferred name
2. write the user ID and preferred name to one of two files
3. allow the user to end the program or repeat from step 1.

The program will consist of three separate modules. Each module will be implemented using either a procedure or a function.

Nadine has defined the modules as follows:

Module	Description
<code>TopLevel()</code>	<ul style="list-style-type: none">• Call <code>GetInfo()</code> to obtain a string containing a valid user ID and a preferred name• Call <code>WriteInfo()</code> to write the string to either <code>File1.txt</code> or <code>File2.txt</code> depending on the first character of the user ID as follows:<ul style="list-style-type: none">◦ ‘A’ to ‘M’: writes to <code>File1.txt</code>◦ ‘N’ to ‘Z’: writes to <code>File2.txt</code>For example, a string with a user ID of "G1234" writes to <code>File1.txt</code>• End the program if the file write was unsuccessful• Input (Y/N) to either repeat for the next user ID or to end the program
<code>GetInfo()</code>	<ul style="list-style-type: none">• Input a user ID and repeat until the user ID is valid• Input a preferred name. This will be an empty string if no preferred name is input.• Concatenate the user ID and preferred name using a '*' character as a separator and return this string
<code>WriteInfo()</code>	<ul style="list-style-type: none">• Open the file• Append the concatenated string to the file• Close the file• Return a Boolean value:<ul style="list-style-type: none">◦ <code>TRUE</code> if the file write was successful◦ <code>FALSE</code> if the file write failed, for example, if the disk was full

A valid user ID:

- is five characters in length
- has a single upper case alphabetic character followed by four numeric characters, for example “G1234”.

Nadine has decided that global variables and nested modules must not be used.

Nadine wants all inputs to have suitable prompts.

(a) Write **program code** for the module `GetInfo()`.

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.

Programming language

Program code

[illegible]

(b) Write **program code** for the module `TopLevel()`.

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.

Programming language

Program code

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

(c) Write **pseudocode** for the module declaration of `WriteInfo()`.

.....

.....

.....

..... [3]

- 10** A text file, `StudentContact.txt`, contains a list of names and telephone numbers of students in a school. Not all students in the school have provided a contact telephone number. In this case, their name will not be in the file.

Each line of the file is stored as a string that contains a name and telephone number, separated by the asterisk character (`'*'`) as follows:

`<Name>'* '<TelNumber>`, for example:

`"Bill Smith*081234567"`

A 1D array, `ClassList`, contains the names of students in a particular class. The array consists of 40 elements of string data type. You can assume that student names are unique. Unused elements contain the empty string `""`.

A program is to be written to produce a **new** text file, `ClassContact.txt`, containing student names and numbers for all students in a particular class.

For each name contained in the `ClassList` array, the program will:

- search the `StudentContact.txt` file
- copy the matching string into `ClassContact.txt` if the name is found
- write the name together with `"*No number"` into `ClassContact.txt` if the name is not found.

The program will be implemented as three modules. The description of these is as follows:

Module	Description
<code>ProcessArray()</code>	<ul style="list-style-type: none">• Check each element of the array:<ul style="list-style-type: none">○ Read the student name from the array○ Ignore unused elements○ Call <code>SearchFile()</code> with the student name○ If the student name is found, call <code>AddToFile()</code> to write the student details to the class file○ If the student name is not found, call <code>AddToFile()</code> to write a new string to the class file, formed as follows: <code><Name>"*No number"</code>• Return the number of students who have not provided a telephone number
<code>SearchFile()</code>	<ul style="list-style-type: none">• Search for a given student name at the start of each line in the file <code>StudentContact.txt</code>:<ul style="list-style-type: none">○ If the search string is found, return the text line from <code>StudentContact.txt</code>○ If the search string is not found, return an empty string
<code>AddToFile()</code>	<ul style="list-style-type: none">• Append the given string to a specified file, for example, <code>AddToFile(StringName, FileName)</code>

(c) `ProcessArray()` is modified to make it general purpose. It will now be called with two parameters as follows:

- an array
- a string representing the name of a class contact file

It will still return the number of students who have not provided a contact telephone number.

Write **program code** for the header (declaration) of the modified `ProcessArray()`.

Programming language

Program code

.....

.....

.....

..... [3]

11 Account information for users of a library is held in one of two text files; `UserListAtoM.txt` and `UserListNtoZ.txt`

The format of the data held in the two files is identical. Each line of the file is stored as a string that contains an account number, name and telephone number separated by the asterisk character ('*') as follows:

`<Account Number>*<Name>*<Telephone Number>`

An example of one line from the file is:

`"GB1234*Kevin Mapunga*07789123456"`

The account number string may be **six** or **nine** characters in length and is **unique for each person**. It is made up of alphabetic and numeric characters only.

An error has occurred and the same account number has been given to different users in the two files. There is **no** duplication of account numbers **within each individual file**.

A program is to be written to search the two files and to identify duplicate entries. The account number of any duplicate found is to be written to an array, `Duplicates`, which is a 1D array of 100 elements of data type `STRING`.

The program is to be implemented as several modules. The outline description of three of these is as follows:

Module	Outline description
<code>ClearArray()</code>	<ul style="list-style-type: none">Initialise the global array <code>Duplicates</code>. Set all elements to the empty string.
<code>FindDuplicates()</code>	<ul style="list-style-type: none">Read each line from the file <code>UserListAtoM.txt</code><ul style="list-style-type: none">Check whether the account number appears in file <code>UserListNtoZ.txt</code> using <code>SearchFileNtoZ()</code>If the account number does appear then add the account number to the array.Output an error message and exit the module if there are more duplicates than can be written to the array.
<code>SearchFileNtoZ()</code>	<ul style="list-style-type: none">Search for a given account number in file <code>UserListNtoZ.txt</code><ul style="list-style-type: none">If found, return <code>TRUE</code>, otherwise return <code>FALSE</code>

(a) State **one** reason for storing data in a file rather than in an array.

.....
..... [1]

(b) Write **program code** for the module `SearchFileNtoZ()`.

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.

Programming language

Program code

[7]

(c) Write **pseudocode** for the module `FindDuplicates()`.

The module description is given in the table on page 12.

[illegible]

.....

.....

.....

.....

.....

.....

..... [8]

(d) `ClearArray()` is to be modified to make it general purpose. It will be used to initialise any 1D array of data type `STRING` to any value.

It will now be called with three parameters as follows:

1. The array
2. The number of elements
3. The initialisation string

You should assume that the lower bound is 1.

(i) Write **pseudocode** for the modified `ClearArray()` procedure.

.....

.....

.....

.....

.....

.....

.....

..... [3]

(ii) Write **program code** for a statement that calls the modified `ClearArray()` procedure to clear the array `Duplicates` to "Empty".

Programming language

Program code

.....

..... [2]

12 A text file, `StudentList.txt`, contains a list of information about students in a school.

Each line of the file contains a reference, name and date of birth for one student. All the information is held as strings and separated by the asterisk character (' * ') as follows:

`<Reference>'*<Name>'*<Date Of Birth>`

An example of one line from the file is:

`"G1234*Aleza Hilton*05062001"`

The reference string may be five or eight characters in length and is unique for each student. It is made up of alphabetic and numeric characters only.

A global 1D array, `Leavers`, contains the references of all students who have recently left the school. The array consists of 500 elements of data type `STRING`. Unused elements contain the empty string `""`.

A program is to be written to produce a new text file, `UpdatedList.txt`, containing information only for students who are still attending the school.

The program is to be implemented as several modules. The outline description of three of these is as follows:

Module	Outline description
<code>ProcessStudentList()</code>	<ul style="list-style-type: none">• Read each line from the file <code>StudentList.txt</code><ul style="list-style-type: none">◦ Check whether the <code>Reference</code> appears in the array using <code>SearchLeavers()</code>◦ If the <code>Reference</code> does not appear then write the line to the file <code>UpdatedList.txt</code>• Return the number of lines not copied.
<code>SearchLeavers()</code>	<ul style="list-style-type: none">• Search for a given <code>Reference</code> in the array <code>Leavers</code>• If the <code>Reference</code> is found, return <code>TRUE</code>, otherwise return <code>FALSE</code>
<code>CountTimes()</code>	<ul style="list-style-type: none">• Take two parameters: the name of an array and a string.• Count the number of elements that are the same as the string. Return the count value.

- (a) Write **program code** for the module `SearchLeavers()`. Declare any additional variables you use.

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.

Programming language

Program code

[6]

(b) Write **pseudocode** for the module `ProcessStudentList()`.

The module description is repeated here for reference.

Module	Outline description
ProcessStudentList()	<ul style="list-style-type: none"> Read each line from the file <code>StudentList.txt</code> <ul style="list-style-type: none"> Check whether the <code>Reference</code> appears in the array using <code>SearchLeavers()</code> If the <code>Reference</code> does not appear then write the line to the file <code>UpdatedList.txt</code> Return the number of lines not copied.

[illegible]

The module description is repeated here for reference.

Write a statement in **program code** that uses `CountTimes()` to assign the count of unused elements to the variable `Result`.

Program code

..... [3]

13 A text file, `Library.txt`, stores information relating to a book collection. The file stores four pieces of information about each book on separate lines of the file, as follows:

```
Line n:      <Book Title>
Line n + 1:  <Author Name>
Line n + 2:  <ISBN>
Line n + 3:  <Location>
```

Information is stored as data strings.

Information relating to two books is shown:

File line	Data
100	"Learning Python"
101	"Brian Smith"
102	"978-14-56543-21-8"
103	"BD345"
104	"Surviving in the mountains"
105	"C T Snow"
106	"978-35-17635-43-9"
107	"ZX001"

(a) (i) A function, `FindBooksBy()`, will search `Library.txt` for all books by a given author.

The function will store the Book Title and Location in the array `Result`, and will return a count of the number of books found.

Array `Result` is a global 2D array of type `STRING`. It has 100 rows and 2 columns.

Write **pseudocode** to declare the array `Result`.

.....

.....

..... [3]

(ii) Function `FindBooksBy()` will:

- receive the Author Name as a parameter
- search `Library.txt` for matching entries
- store the Book Title and Location of matching entries in the Result array
- return an integer value giving the number of books by the author that were found.

Write **program code** for the function `FindBooksBy()`.

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.

Programming language

Program code

[illegible]

.....

.....

.....

.....

.....

..... [8]

(b) The function `FindBooksBy()` has already been called and has stored values in the array `Result`.

The procedure, `DisplayResults()`, will output the information from the array.

The procedure receives the following two parameters:

- a string containing the author name
- an integer value representing the number of books found

The output should be formatted as in the following example:

```
Books written by: Brian Smith

Title                               Location
Learning Python                     BD345
Arrays are not lists                 CZ562
Learning Java                        CZ589

Number of titles found: 3
```

If no books by the author are found, the following should be output:

```
Search found no books by: Brian Smith
```

Write **pseudocode** for the procedure `DisplayResults()`.

Refer to the **Appendix** on page 16 for the list of built-in functions and operators.

[7]

Appendix

Built-in functions (pseudocode)

Each function returns an error if the function call is not properly formed.

`LENGTH(ThisString : STRING) RETURNS INTEGER`
returns the integer value representing the length of string `ThisString`

Example: `LENGTH("Happy Days")` returns 10

`LEFT(ThisString : STRING, x : INTEGER) RETURNS STRING`
returns leftmost `x` characters from `ThisString`

Example: `LEFT("ABCDEFGH", 3)` returns string "ABC"

`RIGHT(ThisString: STRING, x : INTEGER) RETURNS STRING`
returns rightmost `x` characters from `ThisString`

Example: `RIGHT("ABCDEFGH", 3)` returns string "FGH"

`MOD(ThisNum : INTEGER, ThisDiv : INTEGER) RETURNS INTEGER`
returns the integer value representing the remainder when `ThisNum` is divided by `ThisDiv`

Example: `MOD(10,3)` returns 1

`MID(ThisString : STRING, x : INTEGER, y : INTEGER) RETURNS STRING`
returns a string of length `y` starting at position `x` from `ThisString`

Example: `MID("ABCDEFGH", 2, 3)` returns string "BCD"

`DIV(ThisNum : INTEGER, ThisDiv : INTEGER) RETURNS INTEGER`
returns the integer value representing the whole number part of the result when `ThisNum` is divided by `ThisDiv`

Example: `DIV(10,3)` returns 3

`NUM_TO_STRING(x : REAL) RETURNS STRING`
returns a string representation of a numeric value.
Note: This function will also work if `x` is of type `INTEGER`

Example: `NUM_TO_STRING(87.5)` returns "87.5"

`STRING_TO_NUM(x : STRING) RETURNS REAL`
returns a numeric representation of a string.
Note: This function will also work if `x` is of type `CHAR`

Example: `STRING_TO_NUM("23.45")` returns 23.45

Operators (pseudocode)

Operator	Description
&	Concatenates (joins) two strings Example: "Summer" & " " & "Pudding" produces "Summer Pudding"
AND	Performs a logical AND on two Boolean values Example: TRUE AND FALSE produces FALSE
OR	Performs a logical OR on two Boolean values Example: TRUE OR FALSE produces TRUE