2.2.2 Arrays

Oct/NOV 2002

7. (a) Describe how an array is initialised in the memory of a computer. **[4]**

(b) Describe how an array may be searched serially to find a specific data item. **[4]**

(c) In a high-level programming language of your choice, write the code to define the record type for the record structure in part (a). **[3]**

May/June 2011. P21/ P22

3. Raul writes a program which will keep a tally of the number of times each letter appears in a given text. He uses an array of size 26 to store the totals for each letter. He then initialised each element of the array.

   (a) What value should Raul give each element? **[1]**
   (b) Define the array and initialise each element of the array using a high-level programming language of your choice. **[4]**
   (c) Write the statements required to update the array when a letter has been read. **[3]**

May/June 2011. P23

2. Amber stores the names of her favourite song tracks in an array. She has 56 track names at the moment and expects to reach 150. She defined the array as one-dimensional, size 150 and will store strings. She then initialised each element of the array.

   (a) (i) What value should Amber give each element? **[1]**
        (ii) Define the array and initialise each element of the array using a high-level programming language of your choice. **[4]**
   (b) Write additional code which will search the contents of the array for a particular track name. Output the position of this track name in the array. (You may assume the track name exists in the array). **[5]**
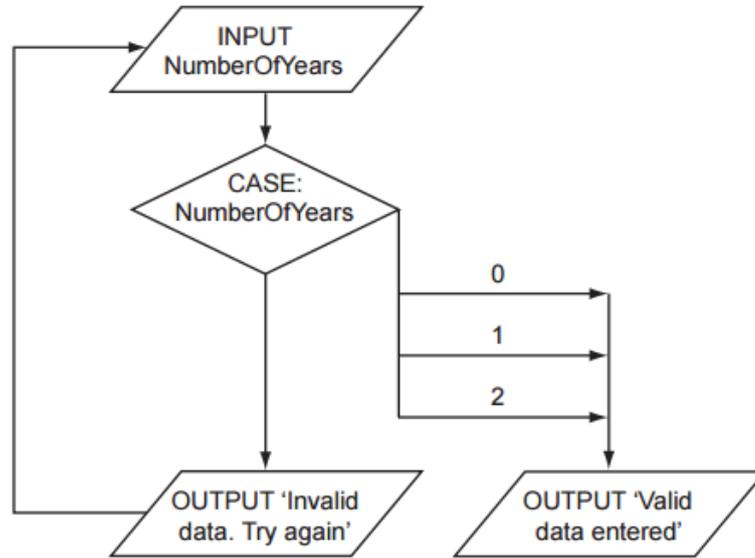
Oct/NOV 2011 P22/P23

2 Ahmed is writing a program to record the data of members of the school football squad.
The input data will be validated. One input is the number of years a member has played for the team. This will be 0, 1 or 2.
The flowchart for the validation of number of years is shown below.

2.2.2 Arrays

(g) The squad has 30 members. Ahmed stores the records in an array called Squad. To calculate how many females there are he designs this pseudocode.

NoOfFemales ← 0

Index ← 1

WHILE Index < 30

    IF Squad[Index].Sex = 'f'

      THEN

        NoOfFemales ← NoOfFemales + 1

    ENDIF

    Index ← Index + 1

ENDWHILE

This pseudocode will only consider the first 29 records in the array.

(h) Write this updated pseudocode using a FOR loop ensuring that it will check all records in the array. **[3]**

2.2.2 Arrays

May/June 2012. P21/22

2 Philipe is trying different ways of designing the process of entering data into an array.

He declares a variable called `ArraySize` and sets it to 3.

He declares an array `Number[ArraySize]`.

He then writes the following pseudocode.

```
Element ← 1

WHILE Element < ArraySize DO

                INPUT Number[Element]

                Element ← Element + 1

ENDWHILE
```

(d) Philipe wants to increase the size of the array to 500. This is too large to check with a trace table.

Describe how you would check that the logic of the pseudocode is correct for 500 iterations.   **[3]**

May/June 2012. P23

3 Liliane wants to write a program to play chess. She will represent the board of 8 x 8 squares, using the 2-dimensional array Board[8,8].

Each element of the array will need initialising to zero. Later, if a chess piece is on a square, it will take a value of 1.

She starts by writing pseudocode for the initialisation of a 4 x 4 board. This is easier to trace.

```
01 RowNo ← 1

02 WHILE RowNo < 4 DO

03        ColumnNo ← 1

04         WHILE ColumnNo < 4 DO

05                 Board[RowNo,ColumnNo] ← 0

06                 ColumnNo ← ColumnNo + 1

07        ENDWHILE

08        RowNo ← RowNo + 1

09 ENDWHILE
```

2.2.2 Arrays

(a) To test this pseudocode she traces it until the first five elements of `Board` have been initialised.

Complete the headings and the trace for these first five elements of the array.

RowNo	ColumnNo	RowNo<4	ColumnNo<4	Board [1,1]				
1	1	True	True	0				

[5]

(e) On a full 8 x 8 board, state the relative positions of the squares `Board[1,8]` and `Board[8,1]`. **[1]**

(f) Liliane's next task is to indicate that there are pieces occupying the first two rows of the 8 x 8 board.

Each square in rows 1 and 2 will be given the value 1.

Draw a flowchart that shows how to do this, using loop structures and the variable names previously used (`Board, RowNo, ColumnNo`). **[5]**

May/June 2013. P21/22

3 Meena needs to be aware of her average grade and declares a variable AvMark, which she decides will be a global variable.

To make future computation more straightforward Meena retrieves her marks from the file and stores them in an array, Marks. This array has 30 elements, and marks range from 0 to100.

(d) State a suitable value to initialise each element of the array. Justify your choice. **[1]**

(e) Write program code that will declare and initialise the array. **[4]**

2.2.2 Arrays

May/June 2013. P23

3 Meena needs to be aware of her average mark and declares a variable with identifier `MyAvMark` which she decides will be a global variable.

The program will read the marks from the file into a one-dimensional array, `MyMarks`. The array has 50 elements, and marks range from 0 to 100 inclusive.

(c) State a suitable value to initialise each element of the array. **[1]**

(d) Write program code that will declare and initialise the array. **[4]**

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.

Juan decides that the design will include the following modules:

- update the resource file when a new teaching resource is purchased
    - input all the data about the resource
    - generate a resource ID for the resource
    - store in the resource file
- update the resource file when a current resource is discarded

(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]**

(f) Juan wants to use an array, NumberOfCopies to store whole numbers.

In a high-level programming language:

- declare the array of size 5000
- initialise the array to 0

[4]

Oct/Nov 2013. P22

2 James plans to store the titles and authors of his favourite ebooks in an array. He has 56 ebooks at the moment and expects to reach 150.

2.2.2 Arrays

A two-dimensional array `MyEbooks` is to be used.

    (a) In a high-level programming language, write the array declaration statement. **[3]**

    (b) (i) Why is it good programming practice to initialise the array `MyEbooks`? **[1]**

           (ii) Write program code to initialise the array `MyEbooks`. **[3]**

Oct/Nov 2013. P23

2 (c) Ashvin will use an array `Equipment` to hold equipment names.

In a high-level programming language:

- declare the array of size 500
- initialise the array **[4]**

May/June 2014. P21/P22

2 The teacher wants a new program to test children. A sequence of multiplication questions is displayed. If a child inputs 3 incorrect answers for a question, the program outputs the number of questions the child answers correctly. Then the program stops.

The program design uses the following functions and procedures.

Function or procedure name	Description
`Random(X)`	A function that returns a whole number in the range 1 to `X` inclusive
`Display(Number1, Number2)`	A procedure that produces the output $5 \times 8 = \qquad ?$ when `Number1` is 5 and `Number2` is 8
`CheckAnswerCorrect(Number1, Number2)`	The procedure from Question **1(d)**
`TestScoreTotal`	A function that returns the number of questions answered correctly

(b) During the school day, several children in the class will use this program. The teacher wants to store each child's name and their best test score so far.

Assume there will be no more than 30 children in the class.

    (i)   Two one-dimensional arrays, `Name` and `BestScore`, are to store the name and score for each child in the class.
Use a high-level programming language to declare these two arrays. **[2]**

    (ii)   Alternatively, the data for one child could be stored in a record `StudentScore`.

2.2.2 Arrays

Use a high-level programming language to declare `StudentScore`. **[2]**

(iii) Use the same programming language as in part (b)(ii) to declare a one-dimensional array of records to store the data for the whole class. Use the identifier `Student`. **[2]**

(iv) Write the program statements to assign the following values to the third record in the array:
- Name: 'Anji'
- Score: 15 **[3]**

Oct/Nov 2014.P21/P22

1 Rema surveys the students in her class to find out which is the most popular sport.

She draws a tally chart:

1	Cricket	\\\\\\\\
2	Football	\\
3	Tennis	\\\
4	Swimming	\\\\\\

Rema plans to collect sport data from students in the whole school. She designs a program to:

- input the number of the sport a student likes best (1, 2, 3 or 4)
- repeatedly ask for input until the input is 0 (zero)
- keep a count of each choice
- on completion of data entry, print out the results as a tally chart (as shown above)

Rema's first attempt is the following pseudocode:

```
Cricket ← 0

Football ← 0

Tennis ← 0

Swimming ← 0

REPEAT

    INPUT Choice

    CASE Choice OF

        1: Cricket ← Cricket + 1

        2: Football ← Football + 1

        3: Tennis ← Tennis + 1
```

2.2.2 Arrays

```
        4: Swimming ← Swimming + 1

    ENDCASE

UNTIL Choice = 0

OUTPUT "Cricket ", Cricket

OUTPUT "Football ", Football

OUTPUT "Tennis ", Tennis

OUTPUT "Swimming ", Swimming
```

Her friend Aisha suggests that the pseudocode could be improved by:

- using a one-dimensional array, Tally, instead of four variables to store the counts
- modularising the design. The main program should just consist of three procedure calls:
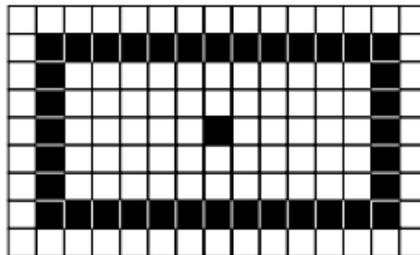 ```
  InitialiseArrayCounts
  InputStudentChoices
  OutputTallyChart
  ```

(a) Write a pseudocode statement to declare the array `Tally`.                                    **[2]**

4 Ahmed combines white tiles with tiles of one other colour to make a pattern.

He draws a design. Here is one example:



Ahmed wants a program to calculate how many tiles he needs of each colour.

(a) Ali stores the design in a 2-dimensional array, `FloorDesign`. The length and width of the room will be no more than 35 tiles each.

    (i)      Suggest a value Ali should use to represent the white and coloured tiles.                 **[1]**

    (ii)     Initially, every tile should be white.

            In a high-level language, write statements to declare the array  `FloorDesign` and initialise it.

                                                    **[5]**

(b) Ali stores a design of size 15 tiles by 10 tiles in array `FloorDesign`.
    Using pseudocode or a flowchart, show the algorithm to calculate the number of white tiles and the number of coloured tiles in the design.                                                                          **[6]**

## 2.2.2 Arrays

3 A game is played by two players. Player A uses white tokens ( ◯ ). Player B uses black tokens (●). The players take turns dropping tokens into a vertical grid. The tokens fall straight down and occupy the next available space in the chosen column. The aim of the game is to connect four of one's own colour tokens. This must be done in a vertical, horizontal or diagonal line.

Here is one example after Player A has had 2 turns and Player B has had 1 turn:
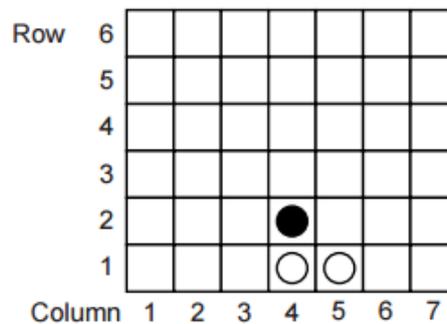


Nathan wants to write a program to allow two users to play the game on the computer.

The program will display a simplified version of the above grid which is redrawn after every turn.

(a) Before any tokens have been dropped into the grid, all grid cells are empty.

    (i)       Suggest values Nathan should use to represent:
            Empty cell
             White token
            Black token        **[2]**

    (ii)     The array Grid is to be used to represent the contents of the game grid.
            In a high-level programming language, write statements to:
            • declare the array Grid
            • assign the value for an empty cell to all cells.    **[7]**

    (iii)    Write the statement to assign the value for a black token to the cell shown in the diagram.    **[2]**

3 A board game is designed for two players, O and X.

At the beginning, all cells of a 3 x 3 grid are empty.

The players take turns in placing their marker in an empty cell of the grid; player O always starts.

The game ends when one player completes a row, column or diagonal or the grid is full.

Here is one example after three turns:

## 2.2.2 Arrays



Ali wants to write a program to play the game.

(a) The array `Grid` is to be used to represent the contents of the grid.
Rows and columns are to be numbered from 1 to 3.

    (i)      To take their turn, the player inputs a row number and a column number to place their marker in an empty cell. Write the values player X has input to place their marker, 'X', in the above diagram. **[1]**

    (ii)    State the value Ali could use to represent an empty cell. **[1]**

    (iii)   In a high-level programming language, write statements to:
        • declare the array `Grid`
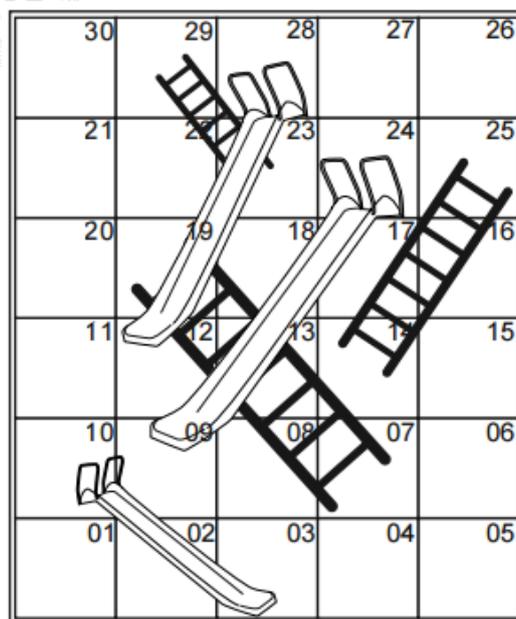        • assign the value for an empty cell to all cells **[7]**

**May/June 2015.P23**

2 Ali has designed a board game.

The board consists of numbered squares.

Slides and ladders connect some of the squares.

An example layout of a board is:

## 2.2.2 Arrays

The object of the game is for a player to move from the bottom square (Square 01) to the top square (Square 30).

A roll of a 6-sided die determines how many squares the player moves.

The table below shows example moves. These use the board layout above, with the player currently on Square 04.

| Number rolled by die | Square moved to | Consequence | Final position |
|---|---|---|---|
| 2 | 06 | – | 06 |
| 3 | 07 | The base of a ladder, so climb to top of ladder | 19 |
| 6 | 10 | The top of a slide, so move down the slide | 03 |

Ali plans to write a program to test different board designs. Each board design will have different numbers, positions and lengths for the slides and ladders.

The program is to test each board design to check that it is possible to complete the game in a reasonable number of moves.

For each board design, the program will simulate playing the game 1000 times and report the average number of moves.

(b) Ali decides to store the board design in a one-dimensional array `Board`.

Each array element represents one square of the board.

If the square is the **start** of a slide or ladder, the array element will store the number of the square where the slide or ladder ends.

For example:

```
                          Board
 [1]   [2]   [3]   [4]   [5]   [6]   [7]   [8]   [9]  [10]       [28]  [29]  [30]
┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐ ┌─────┬─────┬─────┐
│     │     │     │     │     │     │ 19  │     │     │  3  │ │ 12  │     │     │
└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘ └─────┴─────┴─────┘
```

(i)   State what value Ali should use for the array elements that do not contain the start of a slide or ladder. Explain your choice.   **[2]**

(ii) In a high-level programming language, write statements to declare the array `Board`   **[2]**

(iii) In the high-level programming language you chose in part (b)(ii), write statements to initialise the array `Board` before there are any slides or ladders.   **[2]**

2.2.2 Arrays

1 (a) A college provides courses for the local community. The Computing teacher, Ravi, wants to develop a program to keep details of the courses. 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)
course code	CourseCode		015110217	
title	Title		Programming for Beginners	
tutor (3-letter initials)	Tutor		PGL	
day of week (1 – Monday 7 – Sunday)	Day		2	
lab based?	IsLabBased		TRUE	
session duration in hours	SessionHours		2.5	
fee ($)	CourseFee		25.50	
date course starts	StartDate		02/11/2015	
date course ends	EndDate		03/12/2015	

(c) Ravi decides to store course data in the array Course. The array will consist of 50 records.
Initially, the array elements will not contain course data.
Use the high-level language from part (b) to:
- declare the array Course
- initialise its contents with appropriate dummy values for each field

[6]

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.

2.2.2 Arrays

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. **[2]**

(e) Sheena has entered some more book data.
Write pseudocode to save the array records to the serial file `BookData.DAT`
Only those array elements containing book data should be saved to the file. **[5]**

(g) Sheena wants to know the title of her favourite book. She plans to write code to search the array `Book`.
(i) Write pseudocode to output the title of the book with the highest rating. For now, assume that only one book in the `Book` array has the highest rating. **[6]**

(ii) There may be more than one book with the highest rating. Sheena wants to output all the titles that have this rating. The pseudocode will need to be changed.
Describe the changes you would need to make.
Do not write more pseudocode. **[2]**

Computer Science (9608)

May/June 2015.P23

5 A firm employs workers who assemble amplifiers. Each member of staff works an agreed number of hours each day.

The firm records the number of completed amplifiers made by each employee each day.

2.2.2 Arrays

Management monitor the performance of all its workers.

Production data was collected for 3 workers over 4 days

Daily hours worked

Worker 1	5
Worker 2	10
Worker 3	10

Production data

	Worker 1	Worker 2	Worker 3
Day 1	10	20	9
Day 2	11	16	11
Day 3	10	24	13
Day 4	14	20	17

A program is to be written to process the production data.

(a) The production data is to be stored in a 2-dimensional array `ProductionData`, declared as follows:
```
DECLARE ProductionData ARRAY[1:4, 1:3] : INTEGER
```

(i)   Describe two features of an array.   **[2]**

(ii)   Give the value of `ProductionData[3, 2]`.   **[1]**

(iii)   Describe the information produced by the expression:
```
ProductionData[2, 1] + ProductionData[2, 2] + ProductionData[2, 3]
```
[2]

May/ June 2016. P21/P22

3 A string encryption function is needed. The encryption uses a simple character-substitution method.
In this method, a new character substitutes for each character in the original string. This will create the encrypted string.
The substitution uses the 7-bit ASCII value for each character. This value is used as an index for a 1D array, Lookup, which contains the substitute characters.
Lookup contains an entry for each of the ASCII characters. It may be assumed that the original string and the substitute characters are all printable.
For example:
- 'A' has ASCII value 65
- Array element with index 65 contains the character 'Y' (the substitute character)
- Therefore, 'Y' substitutes for 'A'
- There is a different substitute character for every ASCII value

The programmer writes a function, `EncryptString`, to return the encrypted string. This function will receive two parameters, the original, `PlainText` string and the 1D array.

(b) Additional code needs to be written to allow the user to change some of the characters in the array `Lookup`.
The user will input:
- the array start position
- the number of elements to change
- each new substitute character
At the end, the program will finally output a confirmation message.

2.2.2 Arrays

The first version of the algorithm is represented by the flowchart on the following page.

(i) Write **program code** to declare the array `Lookup`. **[2]**

(ii) Write **program code** to implement the flowchart design.

In addition to the Lookup array, assume that the following variables have been declared:

`StartPos, NumToChange, n, NewChar` **[6]**
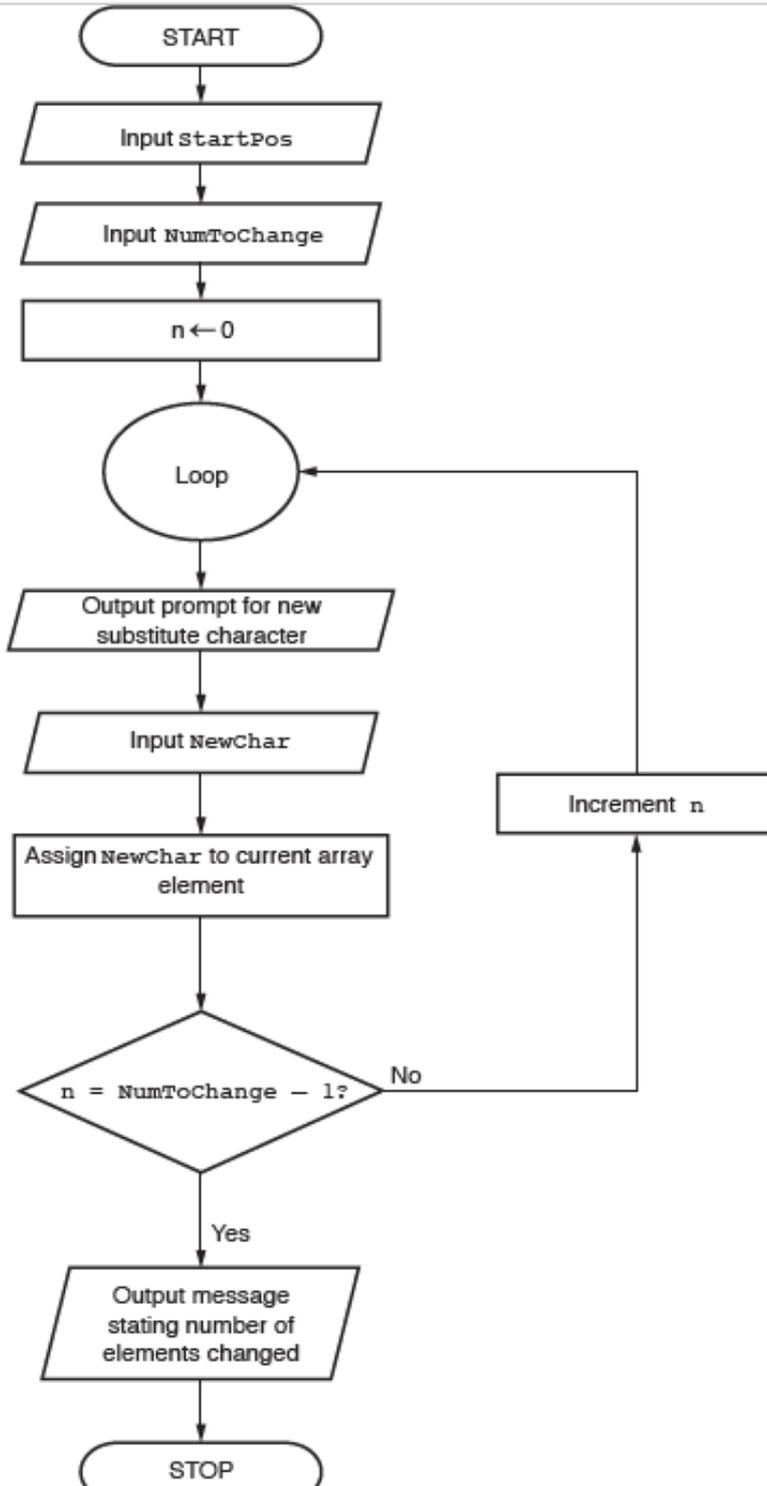
2.2.2 Arrays

3 String encryption was implemented using a simple character-substitution method. A function, Decrypt, is needed to reverse the encryption process and return the original character.

The encryption uses the 7-bit ASCII value for each character. This value is used as an index for a 1D array, Lookup, which contains the substitute characters.

Lookup contains an entry for each of the ASCII characters.

This function, Decrypt, will accept two parameters, a single character, CipherChar, and the 1D array, Lookup.

The steps involved in Decrypt are follows:

- Search for the character in the array
- Note the index value where the character is found (the index value is the ASCII value of the original character)
- Use the index value to obtain the original character

(b) A program is to be written to output part of the Lookup array.

The design of the algorithm is shown below.

It may be assumed that the characters output from Lookup are all printable.

2.2.2 Arrays

```
                    ┌─────────────────┐
                    │      START       │
                    └────────┬────────┘
                             │
                             ▼
                  ╱──────────────────────╲
                 │  Input startIndex and  │
                 │     NumberToOutput      │
                  ╲──────────────────────╱
                             │
                             ▼
                 ┌───────────────────────┐
                 │  Index ← StartIndex    │
                 └───────────┬───────────┘
                             │
                             ▼
                       ╭───────────╮
                       │    Loop    │◄──────────────────┐
                       ╰─────┬─────╯                    │
                             │                          │
                             ▼                          │
                 ┌───────────────────────┐              │
                 │ Produce OriginalChar   │              │
                 │ from Index (using      │    ┌─────────────────────┐
                 │ built-in function)     │    │   Increment Index   │
                 └───────────┬───────────┘    └──────────▲──────────┘
                             │                           │
                             ▼                           │
                 ┌───────────────────────┐               │
                 │ Assign current array   │               │
                 │ element to CipherChar   │               │
                 └───────────┬───────────┘               │
                             │                           │
                             ▼                           │
                  ╱──────────────────────╲               │
                 │  Output message line    │               │
                 │ (see examples on the    │               │
                 │    next page)           │               │
                  ╲──────────────────────╱               │
                             │                           │
                             ▼                           │
                       ╱──────────╲       NO             │
                      ╱  End of     ╲───────────────────┘
                      ╲  loop?      ╱
                       ╲──────────╱
                             │ YES
                             ▼
                    ┌─────────────────┐
                    │      STOP        │
                    └─────────────────┘
```
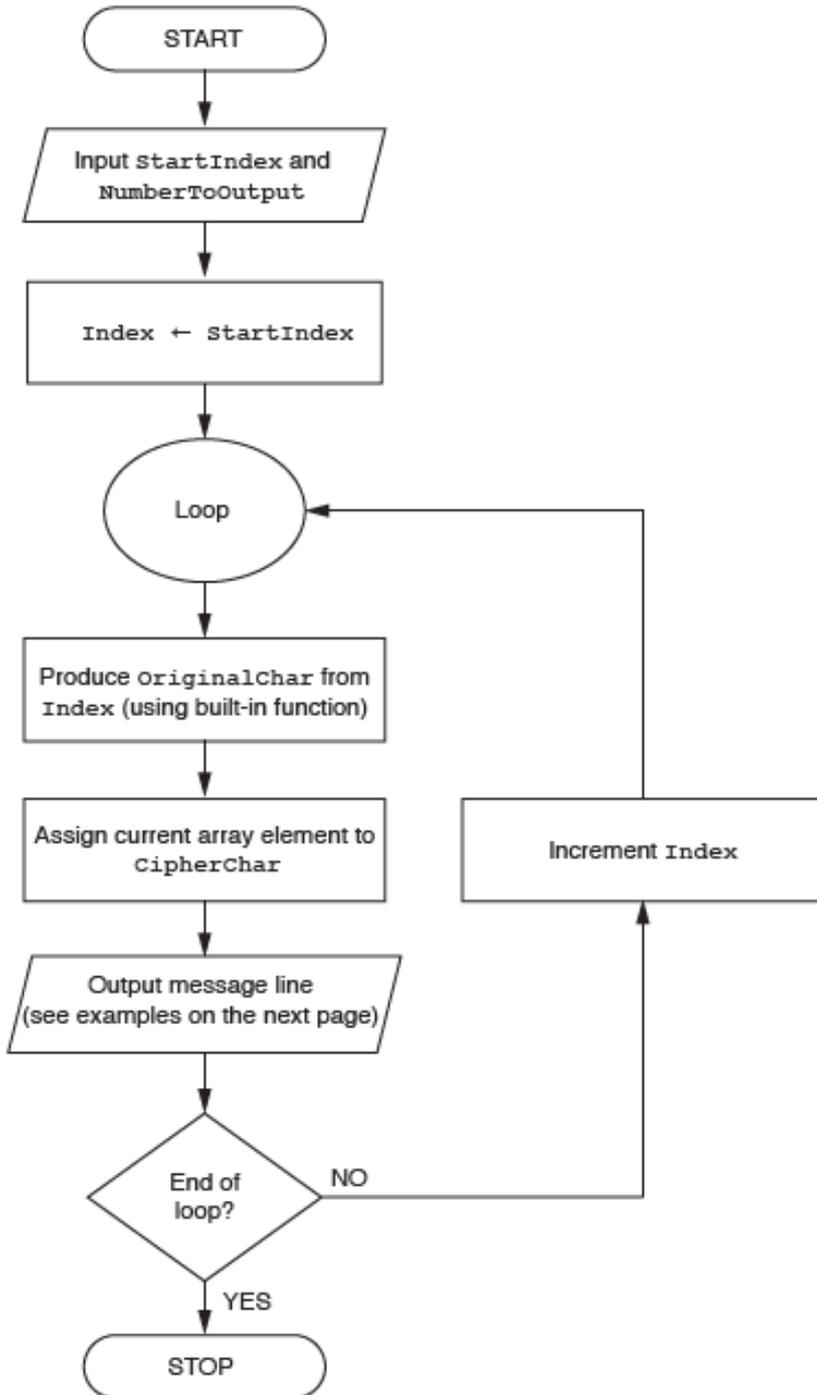
03-111-222-ZAK

OlevelComputer
AlevelComputer

@zakonweb

zak@zakonweb.com

www.zakonweb.com

2.2.2 Arrays

For example, for the input of 65 and 3, the output will be:
```
Index 65: Character A has substitute character Y
Index 66: Character B has substitute character Q
Index 67: Character C has substitute character F
```
Write **program code** to implement the flowchart design.
In addition to the Lookup array, assume that the following variables have been declared:
```
StartIndex, NumberToOutput, Index
```
[6]

Oct/Nov 2016. P22

4 A social club runs a weekly prize draw. A member can buy a ticket for $2. Each week, the club sells up to 150 tickets, with consecutive numbers starting from 1. There is a main cash prize of $120 and a further three prizes of $10. The computer program will generate the winning numbers.

Throughout this question, you will need to refer to the list of pseudocode functions in the **Appendix**.

(d) The organisers do not want the same number to be drawn more than once in a week. To deal with the issue of duplicate numbers being generated, the program in **part (b)** will require adaptive maintenance.

(ii) The club does not want the program to output the same number more than once in a week.

For each winning number, outline the extra steps that the program must have. Include any data structure that the program needs.

Do not write pseudocode or program code. **[3]**

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.

	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

2.2.2 Arrays

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

(a) A programmer designs the following pseudocode for the input of a player's score from one game.

```
01 INPUT GameNumber
02 INPUT PlayerNumber
03 INPUT PlayerGameScore
04 PlayerScore[GameNumber, PlayerNumber]←——PlayerGameScore
```

Describe the data structure the programmer has used for the storage of all player scores.   **[2]**

May/June 2018. P21

6 (a) The following pseudocode includes references to a 1D array.

```
DECLARE StudentGrade : ARRAY[1:5] OF CHAR
DECLARE n : INTEGER
DECLARE x : CHAR
n <- 3
x <- StudentGrade[n]
```

(i) Use the correct technical terms to explain the meaning of `[1:5]` in this pseudocode.   **[2]**

(ii) Use the correct technical term to complete the following statement.

Integer `n` is used as the .... to `StudentGrade`.   **[1]**

(b) A 2D array, `Picture`, contains data representing a bitmap image. Each element of the array represents one pixel of the image. The image is grey-scale encoded where the value of each pixel ranges from 0 (representing black) to 255 (representing white) with intermediate values representing different levels of grey.
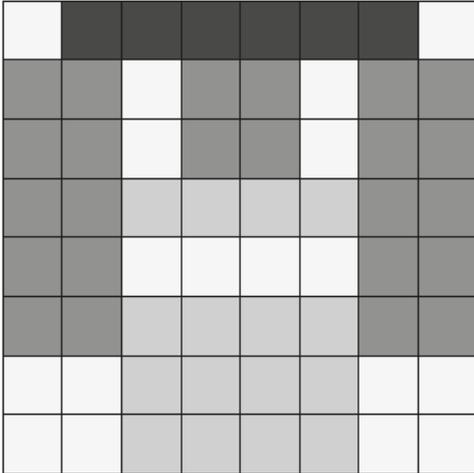
The following is an example of an image and the corresponding data values for the `Picture` array.

2.2.2 Arrays

Bitmap image

Values

240	10	10	10	10	10	10	240
80	80	240	80	80	240	80	80
80	80	240	80	80	240	80	80
80	80	150	150	150	150	80	80
80	80	240	240	240	240	80	80
80	80	150	150	150	150	80	80
240	240	150	150	150	150	240	240
240	240	150	150	150	150	240	240

In pseudocode, the array is declared as follows:
```
DECLARE Picture : ARRAY[1:8, 1:8] OF INTEGER
```

A function, `Lighten()`, is required to lighten the image. Lightening an image may cause it to 'burn out'. An image is said to be 'burnt out' if any pixel is set to the maximum value of 255.
The function `Lighten()` will:
1. increase the value of each pixel by 10%
2. return TRUE if the resultant image is 'burnt out'.
Write pseudocode to implement the `Lighten()` function.
Assume that the array Picture is a global variable. **[8]**

7 A function, `ProcessMarks()`, is required to analyse test marks for a class of students.
- There are 20 students in the class.
- A mark is between 0 and 100.
- The marks for the class are stored in an array, `Mark`, which has 20 elements.
- The array is passed to the function as a parameter.
- The function will output a message stating the average and highest marks. For example:
  `"The average mark is 34 and the highest mark is 76"`
- The function returns the subscript of the highest mark.
Write program code to implement the `ProcessMarks()` function.