



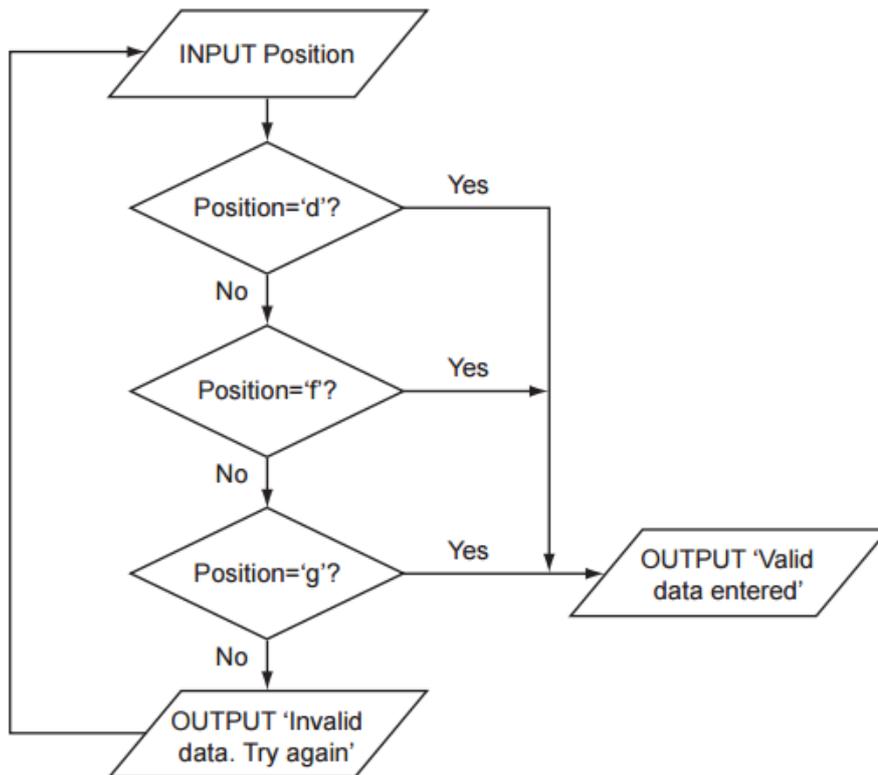
2.1.4 Adaptive Maintenance

Oct/NOV 2011. P21

1 Ahmed is writing a program to record the data of members of the school football squad.

The input data will need to be validated. One input will be the position of each player. Is he/she a forward, defender or goalkeeper? These will be entered as f, d or g.

The flowchart for the validation of the position is shown below.



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

```
Dtotal ← 0
```

```
ArrayPosition ← 1
```

```
REPEAT
```

```
  IF Squad[ArrayPosition].Position = 'd'
```

```
    THEN
```

```
      Dtotal ← Dtotal + 1
```

```
    ENDIF
```





2.1.4 Adaptive Maintenance

```
ArrayPosition ← ArrayPosition + 1
```

```
UNTIL ArrayPosition = 30
```

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

(h) Re-write this updated pseudocode using a WHILE loop ensuring that it will check all records in the array. [3]

May/June 2014.P23

1 (b) Sheena wants to write a function to return the number of boxes of eggs that she needs to buy. The function takes, as a parameter, the number of eggs required for a recipe. There are 6 eggs in a box. Sheena needs to buy enough eggs, but does not want any full boxes of eggs left over.

She knows that she can use the operators DIV and MOD to calculate the required number of boxes.

(ii) Complete the pseudocode:

```
FUNCTION CalculateNumberOfBoxes (NumberOfEggs : ..... )
    RETURNS .....
    DECLARE ..... : INTEGER
    NumberOfBoxes ← ..... // how many full boxes?
    IF NumberOfEggs MOD ..... // need part of a box?
        THEN ..... // increment number of boxes
    .....
    ENDIF
    RETURN NumberOfBoxes
ENDFUNCTION
```

[5]

(d) All programs should be maintainable. Sheena has followed good practice in writing her pseudocode. She has used features of maintainable programs.

List four such features.

[4]

Oct/Nov 2014.P21/P23

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	





2.1.4 Adaptive Maintenance

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
        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
```





2.1.4 Adaptive Maintenance

(e) (i) All programs should be maintainable. Rema has followed good practice in writing her pseudocode. She used some features of maintainable programs.

List three such features.

[3]

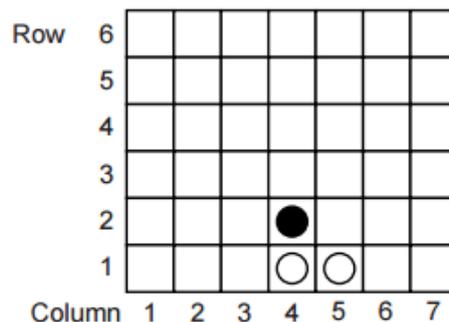
(ii) Give one further feature that Rema has not used in her solution so far.

[1]

Oct/Nov 2014.P22

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.

(d) The program stores in the variable `NextPlayer` the character 'A' or 'B' to show whose turn it is next.

The chosen column number is validated using the function from part (c)(i).

The program then sets the relevant empty grid cell to the player's token value.

Complete the pseudocode:

```

01 REPEAT
02     INPUT ChosenColumnNumber
03 UNTIL ColumnNumberValid( ..... )
04 Row ← 1 // start with bottom row and find first empty row
05 WHILE Grid[Row, ChosenColumnNumber] <> .....
06     .....
07 ENDWHILE
08 IF .....
09     THEN
10         Grid[Row, ChosenColumnNumber] ← .....
    
```





2.1.4 Adaptive Maintenance

```
11     ELSE
12         Grid[Row, ChosenColumnNumber] ← .....
13 ENDIF
```

(f) All programs should be maintainable. List three features of maintainable programs used in the pseudocode in part (d). [3]

Oct/Nov 2015.P21/P23

2 (b)

```
01 PROCEDURE ConvertToDenary(CodedNumber : STRING)
02
03     Denary ← 0
04
05     FOR i ← 1 TO LENGTH(CodedNumber)
06         ThisChar ← MID(CodedNumber, i, 1)
07         ThisNumber ← DenaryDigit(ThisChar)
08         Denary ← Denary + (ThisNumber * 10)
09     ENDFOR
10
11     OUTPUT Denary
12 ENDPROCEDURE
```

(c) (i) The pseudocode in part (b) has some features that make it easier to read and understand. State three such features. [3]

(ii) State one other feature Alia could have used to help the understanding of this code. [1]

Oct/Nov 2015.P22

2 At the end of movies, when credits are listed, the year of production is often shown in Roman numerals.

Roman numbers are formed by combining letters together and adding their value. The letters used and their values are:

M: 1000
D: 500
C: 100
L: 50
X: 10
V: 5
I: 1

For example, MMXV is $1000 + 1000 + 10 + 5 = 2015$.

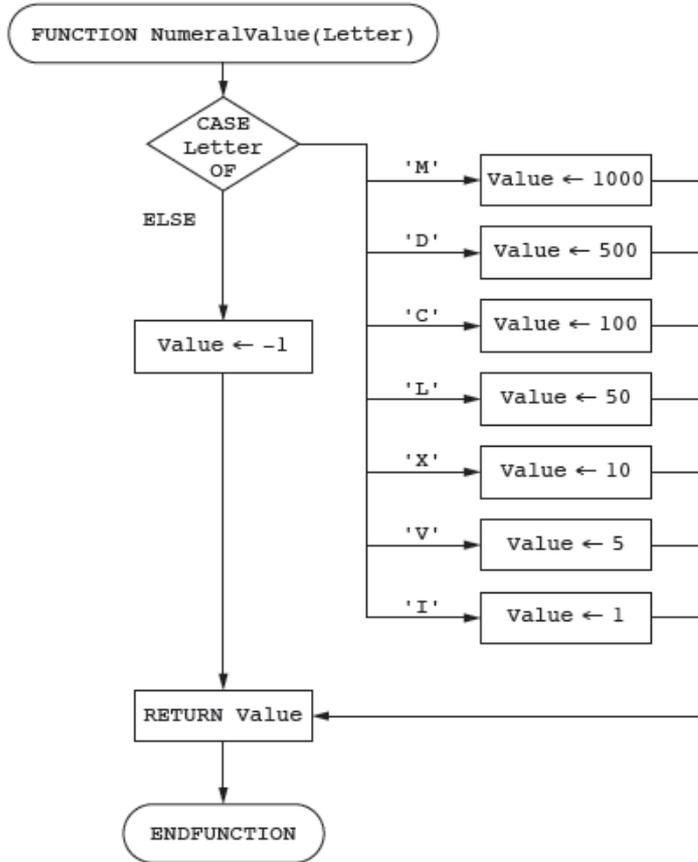
Ali wants to write a program to convert a year written in Roman numerals to denary.

He has drawn the flowchart as part of the design for his solution.





2.1.4 Adaptive Maintenance



(b) Ali writes pseudocode to convert a Roman number using:

- the function `NumeralValue()` from part (a)
- the string manipulation functions `MID` and `LENGTH`

Study the pseudocode:

```
RomanNumber ← "MDCLI"
Denary ← 0
FOR i ← 1 TO LENGTH(RomanNumber)
    ThisLetter ← MID(RomanNumber, i, 1)
    ThisNumber ← NumeralValue(ThisLetter)
    Denary ← Denary + ThisNumber
ENDFOR
OUTPUT Denary
```

(c) (i) The pseudocode has some features that make it easier to read and understand. Identify two such features.

[2]

Computer Science (9608)

Oct/Nov 2015.P21/P23

8 In this question you will need to use the given pseudocode built-in function:





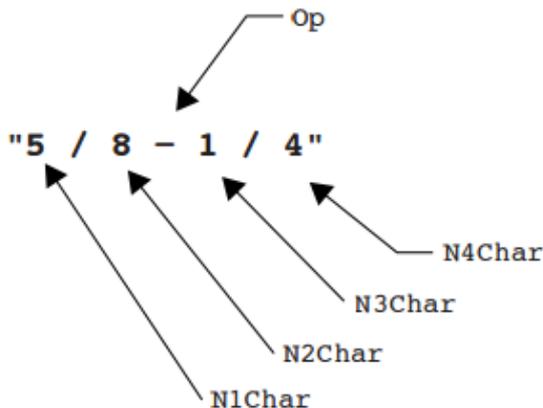
2.1.4 Adaptive Maintenance

ONECHAR(ThisString : STRING, Position : INTEGER) RETURNS CHAR
 returns the single character at position Position (counting from the start of the string with value 1)
 from the string ThisString.
 For example: ONECHAR("Barcelona", 3) returns 'r'.

A program reads a string entered by the user. The string represents the addition or subtraction of two fractions. Each part of the fraction within the string is always a single digit only and the top digit is always less than the bottom digit. Example strings are: "3/8+3/5" and "5/8-1/4"

The program steps are:

- the user enters the string
- the program isolates each digit and the operator
- the program computes the answer as either:
 - a fraction
 - a whole number followed by a fraction
 - a whole number
- the program displays the answer to the user



The identifier table shows the variables to be used to store the characters in the string as shown in the diagram.

Identifier	Data type	Description
FractionString	STRING	String input by user. For example: "5/8-1/4"
N1Char	CHAR	See diagram
N2Char	CHAR	See diagram
N3Char	CHAR	See diagram
N4Char	CHAR	See diagram
Op	CHAR	See diagram

(d) The programmer writes code from the given pseudocode design. The program works, but the design is limited. The programmer is to make amendments to the design following suggested specification changes.

(i) State the name for this type of maintenance.

[1]

(ii) Describe **three** specification changes which will make the program more useful.

[3]





2.1.4 Adaptive Maintenance

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.

(i) Describe what is meant by adaptive maintenance.

[2]

