Example Candidate Responses

Cambridge International AS & A Level

Cambridge International AS & A Level Computer Science

9608

Paper 2

Cambridge Advanced

PAST PAPERS INSIDE
Your education hub

Cambridge International Examinations retains the copyright on all its publications. Registered Centres are permitted to copy material from this booklet for their own internal use. However, we cannot give permission to Centres to photocopy any material that is acknowledged to a third party even for internal use within a Centre.

© Cambridge International Examinations 2016
Version 1

PAST PAPERS INSIDE
Your education hub

Contents

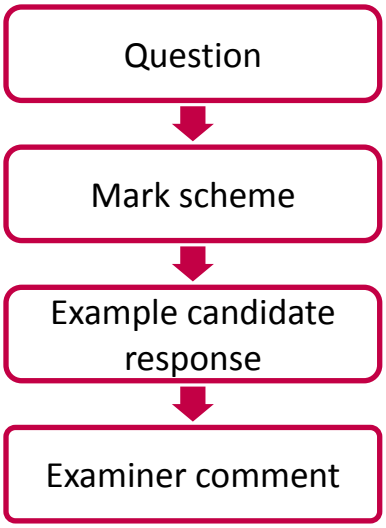
PAST PAPERS INSIDE
Your education hub

Introduction

The main aim of this booklet is to exemplify standards for those teaching Cambridge International AS & A Level Computer Science (9608), and to show how different levels of candidates' performance relate to the subject's curriculum and assessment objectives.

In this booklet candidate responses have been chosen to exemplify a range of answers. Each response is accompanied by a brief commentary explaining the strengths and weaknesses of the answers.

For ease of reference the following format for each component has been adopted:



Each question is followed by an extract of the mark scheme used by examiners. This, in turn, is followed by examples of marked candidate responses, each with an examiner comment on performance. Comments are given to indicate where and why marks were awarded, and how additional marks could have been obtained. In this way, it is possible to understand what candidates have done to gain their marks and what they still have to do to improve their marks.

This document illustrates the standard of candidate work for those parts of the assessment which help teachers assess what is required to achieve marks beyond what should be clear from the mark scheme. Some question types where the answer is clear from the mark scheme, such as short answers and multiple choice, have therefore been omitted.

Past papers, Examiner Reports and other teacher support materials are available on Teacher Support at https://teachers.cie.org.uk

PAST PAPERS INSIDE
Your education hub

Assessment at a glance

For Cambridge International AS and A Level Computer Science, candidates may choose:

- to take Papers 1, 2, 3 and 4 in the same examination series, leading to the full Cambridge International A Level
- to follow a **staged** assessment route by taking Papers 1 and 2 (for the AS Level qualification) in one series, then Papers 3 and 4 (for the full Cambridge International A Level) in a later series
- to take Papers 1 and 2 only (for the AS Level qualification).

Components All candidates take	Weighting (%)	
	AS	A
Paper 1 Theory Fundamentals This written paper contains short-answer and structured questions. There is no choice of questions. 75 marks Externally assessed 1 hour 30 minutes	50	25
Paper 2 Fundamental Problem-solving and Programming Skills This written paper contains short-answer and structured questions. There is no choice of questions. Topics will include those given in the pre-release material.[1] 75 marks Externally assessed 2 hours	50	25
Paper 3 Advanced Theory This written paper contains short-answer and structured questions. There is no choice of questions. 75 marks Externally assessed 1 hour 30 minutes	–	25
Paper 4 Further Problem-solving and Programming Skills This written paper contains short-answer and structured questions. There is no choice of questions. Topics will include those given in the pre-release material.[1] 75 marks Externally assessed 2 hours	–	25

Advanced Subsidiary (AS) forms 50% of the assessment weighting of the full Advanced (A) Level.

Teachers are reminded that the latest syllabus is available on our public website at **www.cie.org.uk** and Teacher Support at **https://teachers.cie.org.uk**

PAST PAPERS INSIDE
Your education hub

Paper 2 – Fundamental Problem-solving and Programming Skills

Question 1

Throughout the paper you will be asked to write either **pseudocode** or **program code**.

Complete the statement to indicate which high-level programming language you will use.

Programming language .....

1 A marathon runner records their time for a race in hours, minutes and seconds.

An algorithm is shown below in structured English.

```
INPUT race time as hours, minutes and seconds
CALCULATE race time in seconds
STORE race time in seconds
OUTPUT race time in seconds
```

(a) The identifier table needs to show the variables required to write a program for this algorithm.

Complete the table.

Identifier	Data type	Description
`RaceHours`	`INTEGER`	The hours part of the race time.

[3]

(b) Before the program is written, the design is amended.

The new design includes input of the runner's current personal best marathon time (in seconds).

The output will now also show one of the following messages:

- "Personal best time is unchanged"
- "New personal best time"
- "Equals personal best time"

(i) Show the additional variable needed for the new design.

Identifier	Data type	Description

[1]

PAST PAPERS INSIDE
Your education hub

Question 1, continued

(ii) Write **program code** for the new design.

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 .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[7]

PAST PAPERS INSIDE
Your education hub

Question 1, continued

(c) The program code will be tested using white-box testing.

(i) Explain what is meant by white-box testing.

.....

.....

.....[2]

(ii) Complete the table heading.

Complete Test Number 1.

Add the data for Test Number 2 and Test Number 3.

	Input values				Output	
Test number	Race hours	Race minutes	Race seconds	.....	Total time (seconds)	Message
1	3	4	13	11053	11053	
2				11053		
3				11053		

[6]

PAST PAPERS INSIDE
Your education hub

Mark scheme

1 (a)

Identifier	Data Type	Description
`RaceHours`	`INTEGER`	The hours part of the race time
`RaceMinutes`	`INTEGER`	the minute part of the race time
`RaceSeconds`	`INTEGER // REAL`	the seconds part of the race time
`RaceTime`	`INTEGER // REAL`	the race time in seconds

3 × (meaningful name + data type) [3]

(b) (i)

Identifier	Data Type	Description
`PersonalBestTime`	`INTEGER/REAL`	Personal best time in seconds

meaningful name + data type [1]

(ii) Mark as follows:

- Declarations/comments for variables – at least 2
- Input (+ prompts) for hours, minutes, seconds
- Input (+ prompt) of personal best time
- Correct calculation of `RaceTimeInSeconds` (don't allow use of 'x' for '*')
- Output `RaceTimeInSeconds`
- Correct logic and output message for < personal best
- Correct logic and output message for > personal best
- Correct logic and output message for = personal best [max 7]

(c) (i)

- Choosing data/values…
- Test every possible 'logic path' through the code
  // with knowledge of the structure/code

Ignore any reference to normal/boundary/extreme … [2]

(ii)

- `PersonalBest` column labelled
- Test number 1 message: "Equals personal best time"/or similar
- Test 2/Test 3 – data for better performance …
- Described with suitable message
- Test 2/Test 3 – data for worse performance …
- Described with suitable message [6]

PAST PAPERS INSIDE Your education hub

Example candidate response – high

Throughout the paper you will be asked to write either **pseudocode** or **program code**.

Complete the statement to indicate which high-level programming language you will use.

Programming language ..... Visual Basic .....

1 A marathon runner records their time for a race in hours, minutes and seconds.

An algorithm is shown below in structured English.

```
INPUT race time as hours, minutes and seconds
CALCULATE race time in seconds
STORE race time in seconds
OUTPUT race time in seconds
```

(a) The identifier table needs to show the variables required to write a program for this algorithm.

Complete the table.

Identifier	Data type	Description
RaceHours	INTEGER	The hours part of the race time.
Race Minutes	Integer	The minutes part in time
RaceSeconds	Integer	the seconds part of the time
TotalTime	Integer	Total time in secs

[3]

(b) Before the program is written, the design is amended.

The new design includes input of the runner's current personal best marathon time (in seconds).

The output will now also show one of the following messages:

- "Personal best time is unchanged"
- "New personal best time"
- "Equals personal best time"

(i) Show the additional variable needed for the new design.

Identifier	Data type	Description
CurrentBest	Integer	total seconds for PB

[1]

PAST PAPERS INSIDE Your education hub

Example candidate response – high, continued

(ii) Write **program code** for the new design.

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 Sub Main

```
DIM RaceHours as Integer
DIM RaceMinutes as Integer
DIM RaceSeconds as Integer
DIM TotalTime as integer
DIM CurrentBest as Integer
Console.Write("Input Runner's hours:")
RaceHours = Console.Readline()
Console.Write("Input Runners minutes:")
RaceMinutes = Console.Readline()
Console.Write(Input Runner's seconds:)
RaceSeconds = Console.Readline()
    TotalTime = (RaceHours * 3600) + (RaceMinutes
    * 60) + RaceSeconds)
Console.Write = ("Input Current Personal best:")
Current Best = Console.Readline()
Console.Write = ("Your total time is" & TotalTime)
    If CurrentBest > TotalTime Then
        Console.Write ("New personal Best time"
    Else if Current Best = TotalTime then
        Console.Write("Equals Personal Best time")
    Else if CurrentBest TotalTime then
        Console.Write(Personal best time is unchanged")
    End If
End
```

[7]

Example candidate response – high, continued

(c) The program code will be tested using white-box testing.

(i) Explain what is meant by white-box testing.

It is when the testers know the program code and used different data to check for different routes for the result. [2]

(ii) Complete the table heading.

Complete Test Number 1.

Add the data for Test Number 2 and Test Number 3.

	Input values				Output	
Test number	Race hours	Race minutes	Race seconds	Current Best	Total time (seconds)	Message
1	3	4	13	11053	11053	Equals personal Best time
2	3	4	16	11053	11056	Personal Best time is unchanged
3	3	4	1	11053	11041	New Personal Best time.

[6]

Examiner comment – high

In parts (a) and (b)(i) the candidate has correctly identified from the given algorithm the variables needed.

In part (b)(ii) candidates must write the name of the programming language which is to be used; that is, either Visual Basic.NET, Python or Pascal. The candidate here has written that the code will be contained in the procedure sub_Main. This information is for the benefit of the marker. The candidate has followed the instructions given and started the coding with the declaration of the variables to be used. The code follows exactly the order of statements in the given algorithm. The input of the three numbers, the calculation of the total time in seconds and the final input of the personal best are all correctly coded. However, the logic for the final stage of the program is incorrect. The candidate should realise that if the new time is greater than the current personal best then the personal best will remain unchanged. However, the candidate's logic for an equal time is correct and so scores one of the available three marks.

This is a clear answer for part (c)(i). The two key points looked for in the answer are firstly – that the programmer will need to have a detailed knowledge of the program code. Secondly – the programmer will need to carefully choose data values designed to test the various paths through the code.
This clear thinking is put into practice for the final part (ii). The candidate has selected one time which is greater that the current personal best and one which is less. The numeric time values are complemented which a clear descriptive message.

Marks awarded for part (a) = 3/3
Marks awarded for part (b) = (i) 1/1, (ii) 5/7
Marks awarded for part (c) = (i) 2/2, (ii) 6/6

Total marks awarded = 17 out of 19

PAST PAPERS INSIDE Your education hub

Example candidate response – middle

Throughout the paper you will be asked to write either **pseudocode or program code.**

Complete the statement to indicate which high-level programming language you will use.

Programming language ..... Visual Basic. .....

1 A marathon runner records their time for a race in hours, minutes and seconds.

An algorithm is shown below in structured English.

```
INPUT race time as hours, minutes and seconds
CALCULATE race time in seconds
STORE race time in seconds
OUTPUT race time in seconds
```

(a) The identifier table needs to show the variables required to write a program for this algorithm.

Complete the table.

Identifier	Data type	Description
RaceHours	INTEGER	The hours part of the race time.
Race minutes	Integer	minutes Part of Time
Race seconds	Integer	Second part of Time
Total time Second	Integer	Total time of race.

[3]

(b) Before the program is written, the design is amended.

The new design includes input of the runner's current personal best marathon time (in seconds).

The output will now also show one of the following messages:

- "Personal best time is unchanged"
- "New personal best time"
- "Equals personal best time"

(i) Show the additional variable needed for the new design.

Identifier	Data type	Description
Best time	Integer	Previous Record.

[1]

PAST PAPERS INSIDE Your education hub

(ii) Write **program code** for the new design.

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 Visual Basic.

```
Module
  Sub Main ()
  Console.writeline("Enter The race timings")
  Console.write("Enter Hours")
  Dim RaceHours As Integer
  Dim RaceMunute as Integer
  Dim Race Seconds as Integer
  Dim Total time Second as Integer
    Race Hours = Console.write line
  Console .Write line ("Enter Race munites)
  Racemiunutes = Console. Read line
  Console .writeline (" Enter Race seconds")
Seconds = Console. Readline
  Total timeseconds = [(Race Hours x 3600) + (Race minute x 60)
                      + Race Seconds]
Console.writeline["Add Current Best time in seconds"]
  Best time = Console. Read line.
      If Best time = Total time seconds then
        Console.writeline ("Equal personal Best time")
      else if Best time < Total time seconds the
        Console.write line("Personal Best time unchange")
      else if  Best time > Total time seconds then.
        Console. Write line("New Personal Best time" & Total time seconds)
      end if
end Sub.
```

[7]

PAST PAPERS INSIDE Your education hub

Example candidate response – middle, continued

(c) The program code will be tested using white-box testing.

(i) Explain what is meant by white-box testing.

Testing without running the code on computer system. The code will be analysed by the analysit. [2]

(ii) Complete the table heading.

Complete Test Number 1.

Add the data for Test Number 2 and Test Number 3.

	Input values				Output	
Test number	Race hours	Race minutes	Race seconds	Add Pervious Best Time	Total time (seconds)	Message
1	3	4	13	11053	11053	Personal Best time unchange
2	3	4	13	11053	11053	Personal Best time unchange
3	3	4	13	11053	11053	Personal Best time unchange

[6]

Examiner comment – middle

In parts (a) and (b)(i) the candidate has followed the given algorithm and used appropriate identifier names.

The program code in part (b)(ii) has gained the maximum seven marks. The candidate has done as instructed and stated the language used before the first line of code. The candidate has used brackets, which are not required for the calculation of the `TotalTimeSeconds` variable, but the mark is still awarded for this statement. The logic for the final IF statement is correct.

In part (c)(i) the candidate is suggesting that white-box testing is the same as a dry-run trace away from the computer. Candidates should appreciate that knowledge of the code is required and test data must be carefully selected. This selection of data is then required in practice for the final part (c)(ii). The candidate has incorrectly repeated the same values.

Marks awarded for part (a) = 3/3
Marks awarded for part (b) = (i) 1/1, (ii) 7/7
Marks awarded for part (c) = (i) 0/2, (ii) 2/6

Total marks awarded = 13 out of 17

PAST PAPERS INSIDE Your education hub

Example candidate response – low

Throughout the paper you will be asked to write either **pseudocode** or **program code**.

Complete the statement to indicate which high-level programming language you will use.

Programming language Visual basic

1 A marathon runner records their time for a race in hours, minutes and seconds.

Variables. ⇒ race time

An algorithm is shown below in structured English.

```
INPUT race time as hours, minutes and seconds
CALCULATE race time in seconds
STORE race time in seconds
OUTPUT race time in seconds
```

(a) The identifier table needs to show the variables required to write a program for this algorithm.

Complete the table.

Identifier	Data type	Description
RaceHours	INTEGER	The hours part of the race time.
RaceMinutes	INTEGER	The minuts part of the race time
RaceSeconds	INTEGER	The seconds part of the race time
~~RaceLeastBestTime~~	~~INTEGER~~	~~The complete race time showing hours minutes and seconds~~
RacePersonalBestTime	INTEGER	The runners personal best time

[3]

(b) Before the program is written, the design is amended.

The new design includes input of the runner's current personal best marathon time (in seconds).

The output will now also show one of the following messages:

- "Personal best time is unchanged" ✓
- "New personal best time" ✓
- "Equals personal best time"

(i) Show the additional variable needed for the new design.

Identifier	Data type	Description
RaceBestTime	Integer	The shortest time completed by the runner

[1]

PAST PAPERS INSIDE Your education hub

Example candidate response – low, continued

(ii) Write **program code** for the new design.

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 Visual Basic

```
Private Sub CmdCalculate_Click()
Dim RM, RS, RH, RT, RB As Integer
RM = TxtRaceminutes.text
RS = TxtRaceseconds.text
RH = TxtRacehours.text
RT = TxtPersonalBestTime.text
RB = TxtRaceBestTime.text

RB = If RB < RT then
    Msgbox("New personal best time")
    Elseif RB = RT then
    Msgbox("Equals personal best time")
    Else If RB > RT then
    Msgbox("Personal best time is unchanged")
    Endif
    Endif
    Endif

End Sub
```

[7]

PAST PAPERS INSIDE Your education hub

Example candidate response – low, continued

(c) The program code will be tested using white-box testing.

(i) Explain what is meant by white-box testing.

White boxing is an in depth testing mechanism that analyses the internal processes of the program to obtain test data that will test the program to give every possible result for every possible input. [2]

(ii) Complete the table heading.

Complete Test Number 1.

Add the data for Test Number 2 and Test Number 3.

	Input values				Output	
Test number	Race hours	Race minutes	Race seconds	Personal best time seconds	Total time (seconds)	Message
1	3	4	13	11053	11053	Personal best time unchanged
2	3	4	13	11053	11053	Equals personal best time
3	2	10	0	11053	7800	New personal best time

[6]

Examiner comment – low

The candidate seems to have confused the current total time with the personal best time. The choice of the third variable for part (a)(i) should match closely to the algorithm given in the question rubric.

For the program code in part (b)(ii) although the candidate has stated this is 'Visual Basic', it is clearly code that would have been written in the Visual Basic forms-based environment. The syllabus guideline states that if the chosen language is Visual Basic, this must use console mode. Also the variables used for the code do not match with those asked for in the identifier tables for the earlier parts (a) and (b)(i). Candidates should be encouraged the see the question as a whole and look for a progression which was the intention here.

The examiner has given the candidate the benefit of any doubt and assumed that the candidates' intention is for `RT` to store the 'current race time' and `RB` to store the 'race best time' i.e. the personal best time. The final statements demonstrate the correct logic and so gain credit.

In part (c)(i) the explanation for white-box testing gains one of the available two marks. The concept of the need for test data is present.

For the final test data table in part (c)(ii), the candidate correctly shows a set of data values which give a new personal best time with a suitable commentary. However, the other data set is for another time which equals the current personal best and this test case is already given in the table.

Marks awarded for part (a) = 2/3
Marks awarded for part (b) = (i) 1/1, (ii) 4/7
Marks awarded for part (c) = (i) 1/2, (ii) 3/6

Total marks awarded = 11 out of 17

PAST PAPERS INSIDE Your education hub

Question 2

2 A program displays a menu with choices 1 to 4. The code to display the menu is written as the procedure `DisplayMenu`.

(a) Pseudocode which uses this procedure is:

```
CALL DisplayMenu
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
UNTIL Choice >= 1 AND Choice <= 4
```

(i) Describe what this pseudocode will do.

.....

.....

.....

.....[3]

(ii) State why a loop is required.

.....

.....[1]

(b) The following pseudocode is a revised design.

```
CONSTANT i ← 3
CALL DisplayMenu
NoOfAttempts ← 0
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
   NoOfAttempts ← NoOfAttempts + 1
UNTIL (Choice >= 1 AND Choice <= 4) OR NoOfAttempts = i
```

(i) Give the maximum number of inputs the user could be prompted to make.

..... [1]

(ii) State why this algorithm is an improvement on the one given in **part (a)**.

.....

.....[1]

PAST PAPERS INSIDE
Your education hub

Question 2, continued

(c) The pseudocode is in its initial stage of development.

The table below shows the action currently taken by the pseudocode following each menu choice.

Menu choice	Description	Program response
1	Read data from the customer file	Calls a procedure `ReadFile` which for testing purposes outputs the message "Read file code"
2	Add a customer	Outputs message "Add customer code"
3	Search for a customer	Outputs message "Search customer code"
4	Terminates the program	Ends

Complete the pseudocode for the design in **part (b)**, shown again below, to respond to each menu choice.

```
CONSTANT i ← 3
CALL DisplayMenu
NoOfAttempts ← 0
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
   NoOfAttempts ← NoOfAttempts + 1
UNTIL (Choice >= 1 AND Choice <= 4) OR NoOfAttempts = i
```

.....

.....

.....

.....

.....

.....

.....[3]

PAST PAPERS INSIDE Your education hub

Question 2, continued

(d) The algorithm in **part (c)** is to be amended. The program will:

- repeatedly display the menu and respond to the user's choice
- terminate when the user enters 4

Write **program code** for this final design which will be made up of:

- the main program
- procedure `ReadFile`
- procedure `DisplayMenu`

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 .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[8]

PAST PAPERS INSIDE
Your education hub

2 (a) (i) Displays the menu (choices)
Repeats the prompt and input …
…the input is a number between 1 and 4 // Checks number is between 1 and 4

"within range" is not enough [3]

(ii) …the input number is validated [1]

(b) (i) 3 [1]

(ii) Previous design repeated indefinitely // (new design) limits number of attempts

Penalise "Program terminates/closes" [1]

(c)

```
IF Choice = 1 THEN (CALL) ReadFile                    (1)
IF Choice = 2 THEN OUTPUT "Add Customer code"         (1)
IF Choice = 3 THEN OUTPUT "Search Customer code"      (1)
IF Choice = 4 THEN END                                (1)
```

alternative answer:

mark as follows:

```
CASE OF Choice // Select CASE Choice      1 mark
1: (CALL) ReadFile                        1 mark (allow CASE = 1)
2: OUTPUT "Add Customer code"             1 mark
3: OUTPUT "Search Customer code"          1 mark
4: END
ENDCASE
```

Output strings must match [max 3]

PAST PAPERS INSIDE
Your education hub

Mark scheme, continued

(d) Mark as follows:

- `Choice / NoOfAttempts` declared/commented as integer
  Must appear within the 'main' program
  Allow: different identifier names
- Constant `i` assigned a value 3
- There is an 'outer' loop to repeatedly display the menu
- Input 'choice' variable
- Three `IF` statements (or equivalent) for processing menu choices 1, 2 and 3
  Note: they must be correctly formed as 'nested' or 'independent'
- Choice 1 calls procedure `ReadFile`
- Choice 2 outputs "Add Customer Code"
  + Choice 3 outputs "Search Customer Code"
- Outer loop terminates correctly with '`Choice` = 4' //or equivalent
- Procedure `DisplayMenu` shows the four menu options
- Procedure `ReadFile` is present …
  and contains a single output message 'Read file code' [max 8]

PAST PAPERS INSIDE
Your education hub

Example candidate response – high

2 A program displays a menu with choices 1 to 4. The code to display the menu is written as the procedure `DisplayMenu`.

(a) Pseudocode which uses this procedure is:

```
CALL DisplayMenu
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
UNTIL Choice >= 1 AND Choice <= 4
```

(i) Describe what this pseudocode will do.

It will first call the procedure DisplayMenu. It will then ask the user for a value. It will keep on asking until the user enters a valid option - 1, 2, 3 or 4.
* Execute [3]

(ii) State why a loop is required.

To make sure that if the user enters an invalid option, he is asked again [1]

(b) The following pseudocode is a revised design.

```
CONSTANT i ← 3
CALL DisplayMenu
NoOfAttempts ← 0
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
   NoOfAttempts ← NoOfAttempts + 1
UNTIL (Choice >= 1 AND Choice <= 4) OR NoOfAttempts = i
```

(i) Give the maximum number of inputs the user could be prompted to make.

3 [1]

(ii) State why this algorithm is an improvement on the one given in **part (a)**.

This method is child safe. Accidental errors will only happen once or twice, therefore, no need to give infinite tries [1]

PAST PAPERS INSIDE Your education hub

Example candidate response – high, continued

(c) The pseudocode is in its initial stage of development.

The table below shows the action currently taken by the pseudocode following each menu choice.

Menu choice	Description	Program response
1	Read data from the customer file	Calls a procedure `ReadFile` which for testing purposes outputs the message "Read file code"
2	Add a customer	Outputs message "Add customer code"
3	Search for a customer	Outputs message "Search customer code"
4	Terminates the program	Ends

Complete the pseudocode for the design in **part (b)**, shown again below, to respond to each menu choice.

```
CONSTANT i ← 3
CALL DisplayMenu
NoOfAttempts ← 0
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
   NoOfAttempts ← NoOfAttempts + 1
UNTIL (Choice >= 1 AND Choice <= 4) OR NoOfAttempts = i
```

CASE Choice OF

1: OUTPUT "Read file code" CALL ReadFile

2: OUTPUT "Add Customer code"

3: OUTPUT "Search customer code"

4: END

ENDCASE

PROCEDURE ReadFile { OUTPUT "Read file code" }

[3]

Example candidate response – high, continued

(d) The algorithm in **part (c)** is to be amended. The program will:

- repeatedly display the menu and respond to the user's choice
- terminate when the user enters 4

Write **program code** for this final design which will be made up of:

- the main program
- procedure `ReadFile`
- procedure `DisplayMenu`

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 Python

```
def DisplayMenu():
    print "1... Read Data from customer file"
    print "2... Add a customer"
    print "3... Search for a customer"
    print "4... End"
def ReadFile():
    print "Read file code"

While TRUE: DisplayMenu()
    print "Enter choice (1..4)"
    choice = int(raw_input("")) // Choice is an INTEGER
    IF choice == 1:
        ReadFile()
    ELIF choice == 2:
        print "Add Customer Code"
    ELIF choice == 3:
        print "Search Customer code"
    ELIF choice == 4:
        END
```

PAST PAPERS INSIDE Your education hub

Examiner comment – high

In part (a)(i) the intention here was a description of what the code would appear to do for the user. Hence the first mark point is for describing that the list of menu choice will be displayed to the user. This is very different from an answer which simply repeats the pseudocode "it calls DisplayMenu". Other key mark points are that the user sees a prompt and the program waits for an input from the user. The program will again prompt the user until the value entered is 1, 2, 3 or 4.

In part (a)(ii) the intention in the design is that the loop is there to perform validation of the user input. Many candidates – including this answer – used the words 'invalid' or 'valid' with a suitable explanation and gained credit.

Part (b)(i) required a simple trace of the algorithm to establish the three iterations. The answer here for part (b)(ii) is clear realising that the improvement is the user will only be prompted a limited number of times. Most candidates answered part (c) with a sequence of IF statements. This candidate has used the alternative CASE structure with the correct composition.

In part (d) this is almost exemplary Python program code. The candidate has stated that the language used is Python and used appropriate indentation. The syntax of the statements is all correct except for the candidate's use of the comment statement. Candidates need to be clear that for this question it is program code that is required and so should use the appropriate comment symbol. The candidate has here incorrectly used syntax for comments in pseudocode.

Marks awarded for part (a) = (i) 2/3, (ii) 1/1
Marks awarded for part (b) = (i) 1/1, (ii) 1/1
Marks awarded for part (c) = 3/3
Marks awarded for part (d) = 7/8

Total marks awarded = 15 out of 17

PAST PAPERS INSIDE
Your education hub

2 A program displays a menu with choices 1 to 4. The code to display the menu is written as the procedure `DisplayMenu`.

(a) Pseudocode which uses this procedure is:

```
CALL DisplayMenu
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
UNTIL Choice >= 1 AND Choice <= 4
```

(i) Describe what this pseudocode will do.

The pseudocode will keep asking for inputs until a valid choice from 1 to 4 is chosen. If an invalid choice such as 5 is chosen, it will ask for an input again. [3]

(ii) State why a loop is required.

So that if an invalid choice is chosen, it can ask for an input again. [1]

(b) The following pseudocode is a revised design.

```
CONSTANT i ← 3
CALL DisplayMenu
NoOfAttempts ← 0
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
   NoOfAttempts ← NoOfAttempts + 1
UNTIL (Choice >= 1 AND Choice <= 4) OR NoOfAttempts = i
```

(i) Give the maximum number of inputs the user could be prompted to make.

3 [1]

(ii) State why this algorithm is an improvement on the one given in **part (a)**.

.....

.....[1]

PAST PAPERS INSIDE Your education hub

Example candidate response – middle, continued

(c) The pseudocode is in its initial stage of development.

The table below shows the action currently taken by the pseudocode following each menu choice.

Menu choice	Description	Program response
1	Read data from the customer file	Calls a procedure `ReadFile` which for testing purposes outputs the message "Read file code"
2	Add a customer	Outputs message "Add customer code"
3	Search for a customer	Outputs message "Search customer code"
4	Terminates the program	Ends

Complete the pseudocode for the design in **part (b)**, shown again below, to respond to each menu choice.

```
CONSTANT i ← 3
CALL DisplayMenu
NoOfAttempts ← 0
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
   NoOfAttempts ← NoOfAttempts + 1
UNTIL (Choice >= 1 AND Choice <= 4) OR NoOfAttempts = i
```

If Choice = 1 then Select Case choice
Case 1 = Call Read File
Case 2 = Console.Writeline ("Add customer code")
Case 3 = Console.Writeline ("Search customer code")
Case 4 = End

[3]

Example candidate response – middle, continued

(d) The algorithm in **part (c)** is to be amended. The program will:

- repeatedly display the menu and respond to the user's choice
- terminate when the user enters 4

Write **program code** for this final design which will be made up of:

- the main program
- procedure `ReadFile`
- procedure `DisplayMenu`

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 Sub Main

```
DIM Choice as Integer
DIM customerCode as Integer
Call DisplayMenu
Repeat
Console.Write("Input Choice")
Choice = Console.Readline()
If Choice = 1 then
    Call ReadFile
    console.write("Read file code")
Else if Choice = 2 then
console.write("Add customer code")
CustomerCode = Console.Readline()

Else if choice = 3 then
    Console.write("Search customer code")

Else if choice = 4 then
end if
end
```

PAST PAPERS INSIDE Your education hub

Examiner comment – middle

Candidates are asked to state the programming language used in the code which follows.
In part (a)(i) the candidate should realise that all lines of the pseudocode will generate some action. There is no mention of the display of the list of actions coded as procedure `DisplayMenu`.

In part (a)(ii) the candidate has correctly identified that the code is designed to trap an invalid response from the user.

In part (b)(i) the candidate has correctly stated that the loop will iterate at most three times but is then unable to comment on the practical reason for this in part (b)(ii). Candidates should appreciate that the code has shown 'good practice' and used meaningful identifier names – so use of the identifier `NoOfAttempts` in the question rubric should provide a trigger.

For part (c) the candidate has not used the exact pseudocode syntax documented in the syllabus section 2.3.3 which has been condoned. Also note the omission of the speech mark delimiters from the case 3 text. Candidates need to pay attention to detail such as this.

For the program code in part (d) there are some correctly formed statements. The candidate needs to appreciate the overall structure of the requirement. The `DisplayMenu` procedure needs to be inside a loop which terminates when the user selects option 4 from the menu.

Marks awarded for part (a) = (i) 2/3, (ii) 1/1
Marks awarded for part (b) = (i) 1/1, (ii) 0/1
Marks awarded for part (c) = 3/3
Marks awarded for part (d) = 4/8

Total marks awarded = 11 out of 17

PAST PAPERS INSIDE
Your education hub

Example candidate response – low

2 A program displays a menu with choices 1 to 4. The code to display the menu is written as the procedure `DisplayMenu`.

(a) Pseudocode which uses this procedure is:

```
CALL DisplayMenu
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
UNTIL Choice >= 1 AND Choice <= 4
```

(i) Describe what this pseudocode will do.

The DisplayMenu program is called from the storage device as it is premade. This program displays a menu that keeps asking for an option from 1 to 4 to be chosen. If option is not chosen in parameters, it will run in a loop until condition is satisfied. [3]

(ii) State why a loop is required.

A loop is required because if an error is made, it should run again until a correct option is chosen without a loop, it would be stored. [1]

(b) The following pseudocode is a revised design.

```
CONSTANT i ← 3
CALL DisplayMenu
NoOfAttempts ← 0
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
   NoOfAttempts ← NoOfAttempts + 1
UNTIL (Choice >= 1 AND Choice <= 4) OR NoOfAttempts = i
```

(i) Give the maximum number of inputs the user could be prompted to make.

3 [1]

(ii) State why this algorithm is an improvement on the one given in part (a).

This is because the algorithm gives a person a considerable amount of time during which user should be able to input correct data. Otherwise some malicious activity might be going on. [1]

PAST PAPERS INSIDE Your education hub

Example candidate response – low, continued

(c) The pseudocode is in its initial stage of development.

The table below shows the action currently taken by the pseudocode following each menu choice.

Menu choice	Description	Program response
1	Read data from the customer file	Calls a procedure `ReadFile` which for testing purposes outputs the message "Read file code"
2	Add a customer	Outputs message "Add customer code"
3	Search for a customer	Outputs message "Search customer code"
4	Terminates the program	Ends

Complete the pseudocode for the design in **part (b)**, shown again below, to respond to each menu choice.

```
CONSTANT i ← 3
CALL DisplayMenu
NoOfAttempts ← 0
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
   NoOfAttempts ← NoOfAttempts + 1
UNTIL (Choice >= 1 AND Choice <= 4) OR NoOfAttempts = i
```

```
IF
    Choise = 1
    CALL READFILE
        OUTPUT " Read File Code"
ELSE
    IF
        CHOICE >2
        OUTPUT " Add customer Code"
    ELSE
        IF
            CHOICE = 3
                OUTPUT " Search customer code"
        ENDIF
        ELSE
    END IF
END IF
END IF
END
```

[3]

PAST PAPERS INSIDE Your education hub

Example candidate response – low, continued

(d) The algorithm in **part (c)** is to be amended. The program will:

- repeatedly display the menu and respond to the user's choice
- terminate when the user enters 4

Write **program code** for this final design which will be made up of:

- the main program
- procedure `ReadFile`
- procedure `DisplayMenu`

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 PRO Python.

```
i=0 and
# Display Menu is the main program, it contains a
variable "choice" which is integer.
i=0
FOR i=3 OR CHOICE >= 1 AND CHOICE <= 4
    Print ("Enter choice (1 to 4)")
    choice = input (int (choice))
    i = i + 1
END FOR
IF
    CHOICE = 2
CustomerCode = input (int ("Add Customer Code", customer code)
IF
    CHOICE = 3
    SEARCH. APPEND (customer code)
IF
    CHOICE = 1
    CALL Read File.
        Print ("Read File code")
IF
    CHOICE = 4
END
END IF
END IF
END IF
END IF
END
```

[8]

© UCLES 2015

PAST PAPERS INSIDE Your education hub

Examiner comment – low

In part (a)(i) the candidate needs to realise that the use of the call statement is for a procedure. The candidate has incorrectly identified this as a 'program'. The candidate has then correctly identified there will be a prompt for the user but there is no mention of this being followed by the input from the user. Part (a)(ii) needs to be clearly expressed. Clear expression of some process is a key skill for the computer scientist.

In part (b)(i) the candidate has correctly identified that there will be at most three iterations but this is followed in part (ii) with a statement that there will be a 'considerable' amount of tries allowed. The key point needs to be made that the code is designed to limit the number of attempts to a stated maximum.

In part (c) the candidate has followed the rubric to write pseudocode and this shows three correctly formed nested IF statements.

In part (d) the candidate has stated that the language to be used in Python. However, they have not followed the question rubric for a Python answer and *shown a comment statement for each variable used with its data type*. This answer has few aspects of the key design present. There has been no attempt to show separate procedures for the `ReadFile` and `DisplayMenu` code. Also the question rubric to *repeatedly display the menu* is not present.

Marks awarded for part (a) = (i) 1/3, (ii) 0/1
Marks awarded for part (b) = (i) 1/1, (ii) 0/1
Marks awarded for part (c) = 3/3
Marks awarded for part (d) = 2/8

Total marks awarded = 7 out of 17

PAST PAPERS INSIDE
Your education hub

Question 3

3 When the guarantee on a computer runs out, the owner can take out insurance to cover breakdown and repairs.

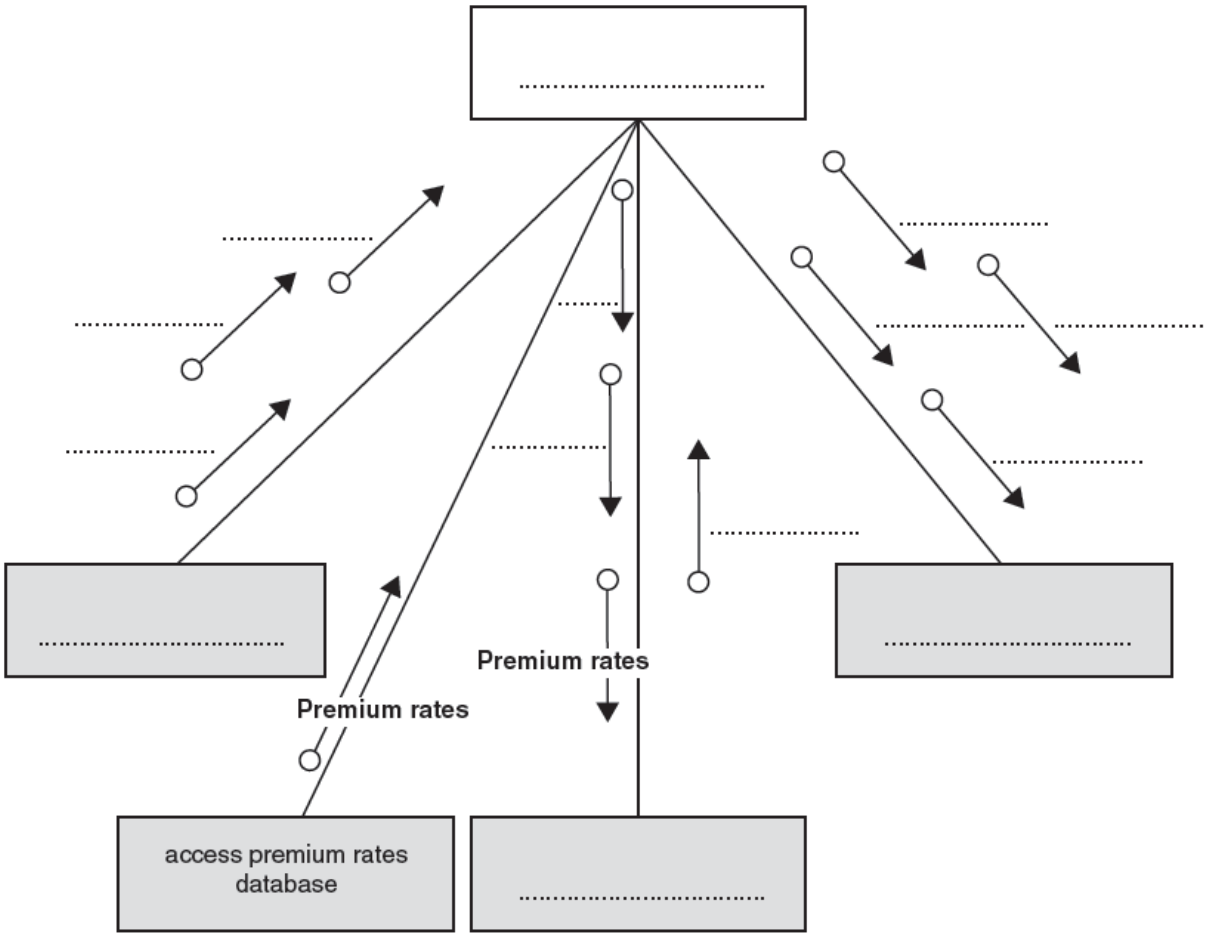
The price of the insurance is calculated from:

- the model of the computer
- the age of the computer
- the current insurance rates

Following an enquiry to the insurance company, the customer receives a quotation letter with the price of the insurance.

A program is to be produced.

The structure chart below shows the modular design for this process:



PAST PAPERS INSIDE
Your education hub

Question 3, continued

(a) Using the letters **A** to **D**, add the labelling to the chart boxes on the opposite page.

Modules	
A	Send quotation letter
B	Calculate price
C	Produce insurance quotation
D	Input computer details

[2]

(b) Using the letters **E** to **J**, complete the labelling on the chart opposite.

Some of these letters will be used more than once.

Data items	
E	`CustomerName`
F	`CustomerEmail`
G	`Model`
H	`Age`
I	`PolicyCharge`
J	`PolicyNumber`

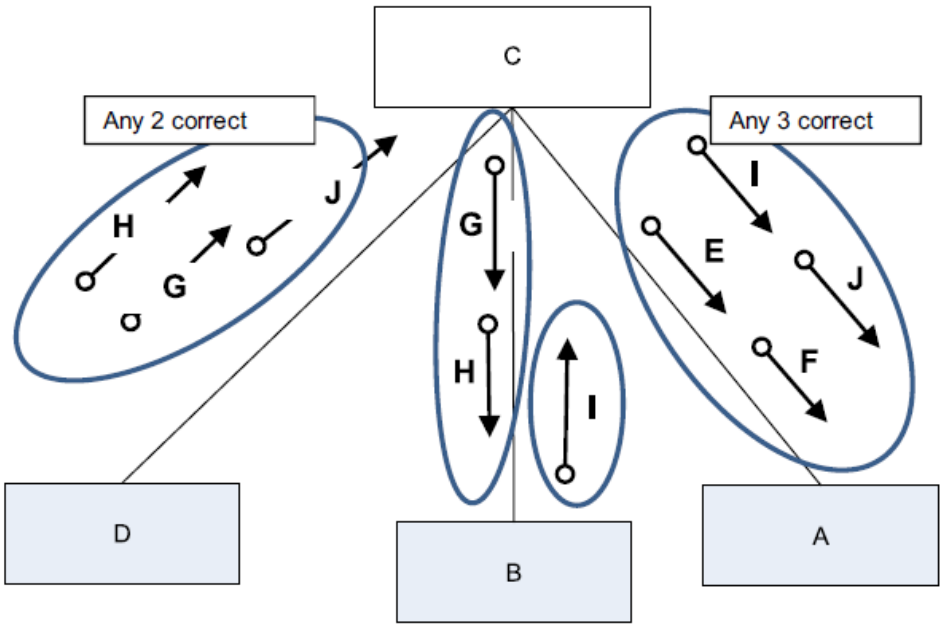
[4]

PAST PAPERS INSIDE
Your education hub

3 (a) Control box – C // Produce insurance quotation [1]

D // Input customer details + A // Send quotation letter is correct positions [1]

(b)



Data items	
E	CustomerName
F	CustomerEmail
G	Model
H	Age
I	PolicyCharge
J	PolicyNumber

[4]

PAST PAPERS INSIDE
Your education hub

Example candidate response – high

3 When the guarantee on a computer runs out, the owner can take out insurance to cover breakdown and repairs.

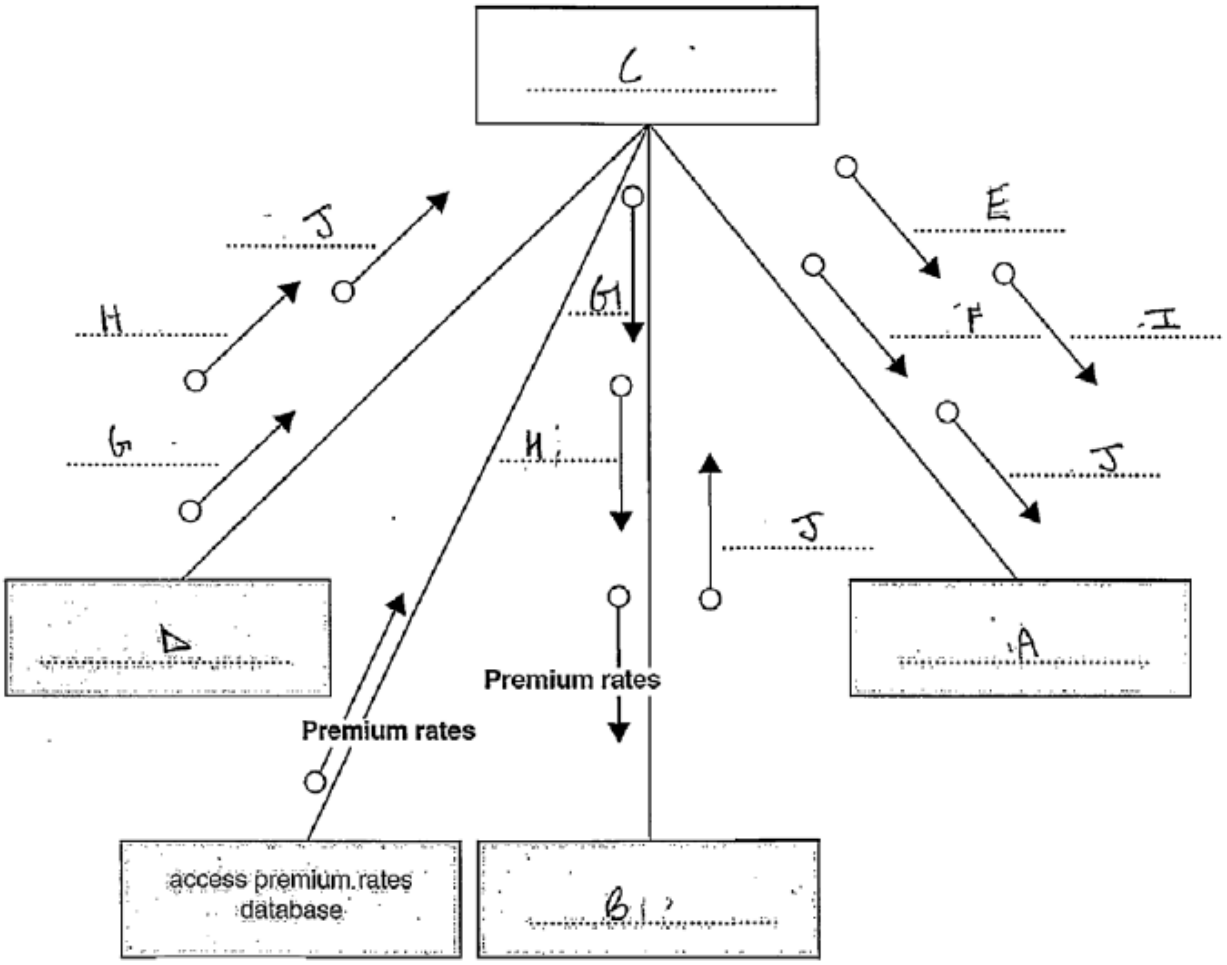
The price of the insurance is calculated from:

- the model of the computer
- the age of the computer
- the current insurance rates

Following an enquiry to the insurance company, the customer receives a quotation letter with the price of the insurance.

A program is to be produced.

The structure chart below shows the modular design for this process:



Examiner comment – high

This is a strong answer. The labelling of the top control box and the procedures is correct. The candidate has correctly identified the data items recorded from the input stage and then correctly used the model and age of the computer for the calculation of the premium price. However, it will be the policy charge which is the output item from procedure B. The items needed for procedure A (send the quotation letter) have been correctly identified.

Marks awarded for part (a) = 2/2
Marks awarded for part (b) = 3/4

Total marks awarded = 5 out of 6

PAST PAPERS INSIDE
Your education hub

Example candidate response – middle

3 When the guarantee on a computer runs out, the owner can take out insurance to cover breakdown and repairs.

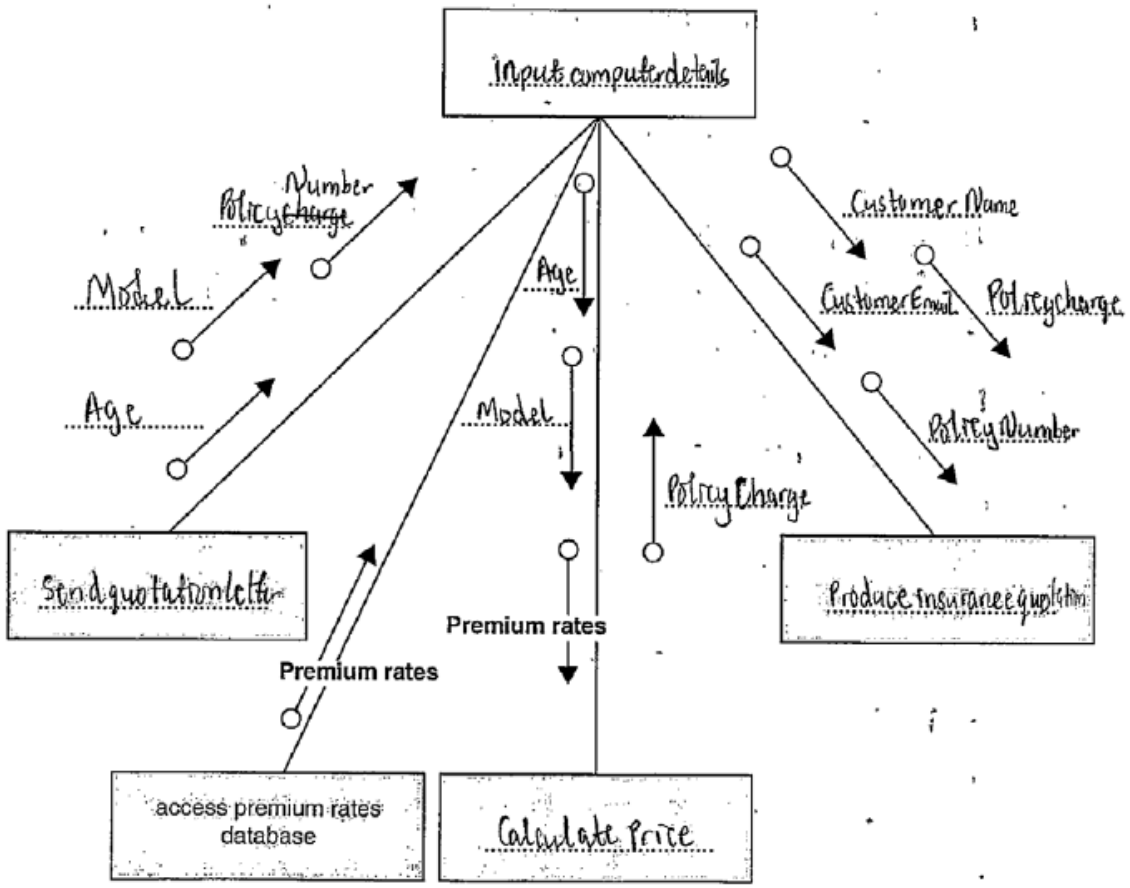
The price of the insurance is calculated from:

- the model of the computer
- the age of the computer
- the current insurance rates

Following an enquiry to the insurance company, the customer receives a quotation letter with the price of the insurance.

A program is to be produced.

The structure chart below shows the modular design for this process:



Examiner comment – middle

The candidate has not followed the rubric of the question and used the letters to complete the labelled of the structure chart. The question has been designed in this way to reduce the amount the candidate has to write.

The candidate has interpreted the text 'send quotation letter'; as a letter which is sent out to the computer owner as an invite to produce a quotation. The candidate should realise that this is a letter which is sent out at the end of the process – to notify the policyholder of the amount the insurance will cost.

The candidate has however 'recovered' with the correct labelling of all the data item arrows for the answer to part (b) and was not penalised for the failure to use letters for the labelling.

Marks awarded for part (a) = 0/2
Marks awarded for part (b) = 4/4

Total marks awarded = 4 out of 6

PAST PAPERS INSIDE
Your education hub

Example candidate response – low

3 When the guarantee on a computer runs out, the owner can take out insurance to cover breakdown and repairs.

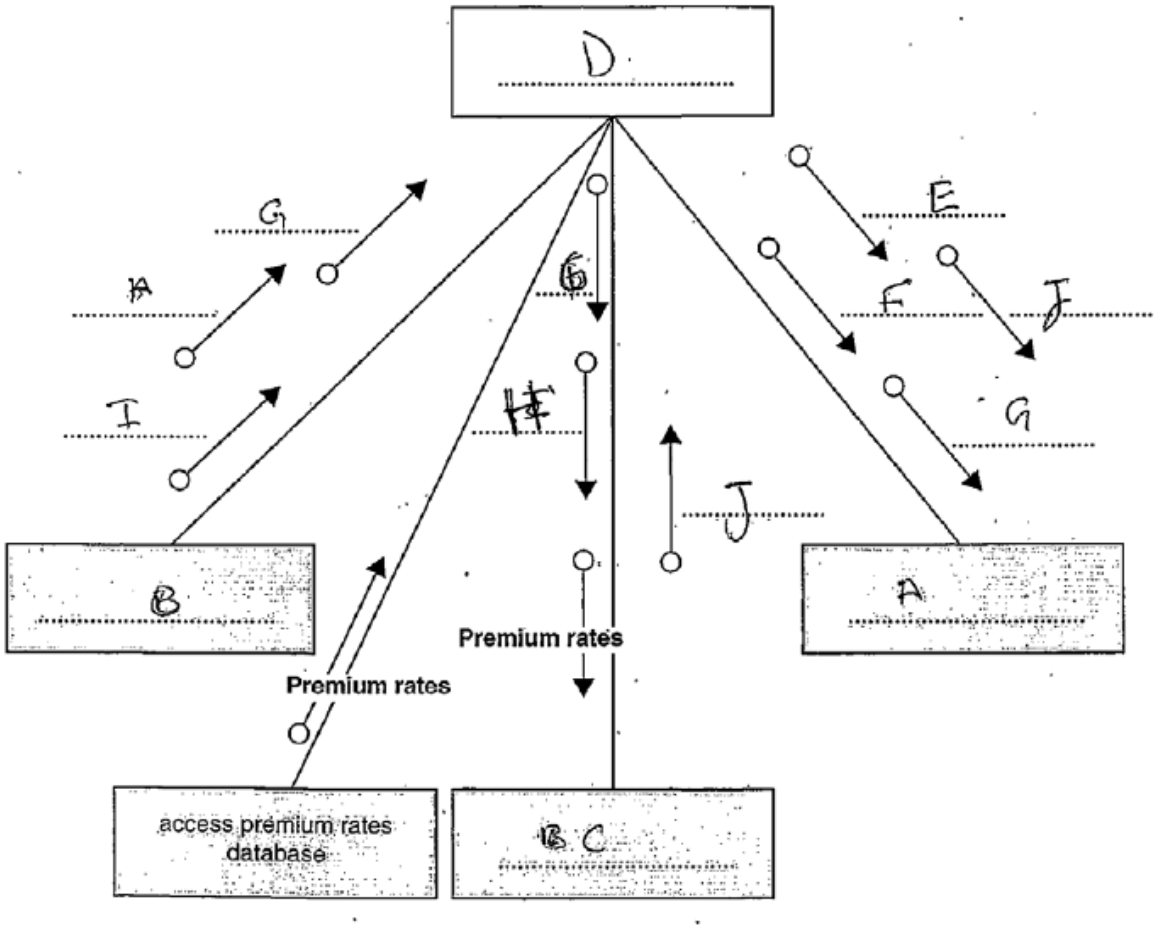
The price of the insurance is calculated from:

- the model of the computer
- the age of the computer
- the current insurance rates

Following an enquiry to the insurance company, the customer receives a quotation letter with the price of the insurance.

A program is to be produced.

The structure chart below shows the modular design for this process:



Examiner comment – low

Candidates need to make the link in the syllabus content between the drawing of a structure chart (section 2.1.2) and the use of procedures when writing program code (Section 2.3.6). Passing parameter values into and out of a procedure call is exactly what the arrows on this structure chart are depicting.

The candidate should appreciate that the control box will describe some overarching process – in this case 'produce insurance quotation'. It then follows that the three processes – in this case the sequence D, B and A – make up this process. The candidate has correctly identified A as the final procedure in the sequence but both D and A were required to secure a mark.

Marks awarded for part (a) = 0/2
Marks awarded for part (b) = 2/4

Total marks awarded = 2 out of 6

PAST PAPERS INSIDE
Your education hub

Question 4

4 A game is played between two players:

- they take turns at rolling a six-sided die (numbered 1 to 6) and record their throw
- a player scores 1 point if their throw is higher than their opponent
- they each roll the die 20 times
- if the player's throw is the same as their opponent, the total points is unchanged
- the winner is the player with the larger number of points after 20 throws

The pseudocode will use variable `NoOfThrows` as shown.

Identifier	Data type	Description
`NoOfThrows`	`INTEGER`	Loop control variable

(i) Complete the pseudocode for the given algorithm.

```
FOR ..........

  INPUT Player1Throw

  ..........

  IF Player1Throw > Player2Throw
    THEN

      ..........

  ENDIF
  IF Player2Throw > Player1Throw
    THEN
      Player2Total ← Player2Total + 1
  ENDIF

..........

IF Player1Total > Player2Total
  THEN
    OUTPUT "Player1 is the winner"
  ELSE
    OUTPUT "Player2 is the winner"
ENDIF
```

[5]

(ii) Identify the game result which will produce incorrect output.

.....

.....[1]

PAST PAPERS INSIDE
Your education hub

Mark scheme

4 (i)
```
FOR NoOfThrows ← 1 TO 20 / 0 TO 19
      1            1                                    (2)
   INPUT Player1Throw
   INPUT Player2Throw                                   (1)
   IF Player1Throw > Player2Throw
      THEN
      Player1Total ← Player1Total + 1                   (1)
   ENDIF
   IF Player2Throw > Player1Throw
      THEN
         Player2Total ← Player2Total + 1
   ENDIF
ENDFOR                                                  (1)

IF Player1Total > Player2Total
   THEN
      OUTPUT "Player1 is the winner"
   ELSE
      OUTPUT "Player2 is the winner"
END
```
[5]

(ii) Player scores equal // if `Player1Total = Player2Total` // there is no winner // a draw
[1]

Example candidate response – high

4 A game is played between two players:

- they take turns at rolling a six-sided die (numbered 1 to 6) and record their throw
- a player scores 1 point if their throw is higher than their opponent
- they each roll the die 20 times
- if the player's throw is the same as their opponent, the total points is unchanged
- the winner is the player with the larger number of points after 20 throws

The pseudocode will use variable `NoOfThrows` as shown.

Identifier	Data type	Description
NoOfThrows	INTEGER	Loop control variable

(i) Complete the pseudocode for the given algorithm.

```
FOR NoOfThrows ← 1 to 20
  INPUT Player1Throw
  Input Player2Throw
  IF Player1Throw > Player2Throw
    THEN Player1Total ← Player1Total + 1
      Player1Throw = Player1Total + 1
  ENDIF
  IF Player2Throw > Player1Throw
    THEN
      Player2Total ← Player2Total + 1
  ENDIF
End for

IF Player1Total > Player2Total
  THEN
    OUTPUT "Player1 is the winner"
  ELSE
    OUTPUT "Player2 is the winner"
ENDIF
```

[5]

(ii) Identify the game result which will produce incorrect output.

If both have entered the same input

If Player1Throw = Player2Throw [1]

PAST PAPERS INSIDE Your education hub

Examiner comment – high

This is a perfect answer for the pseudocode in part (i). Candidates should note that `ENDFOR` is one word in the syllabus pseudocode.

Part (ii) – like the earlier question 2(a)(i) and 2(b)(ii) – required a clear descriptive statement. Here the candidate needed to make clear it is the final score of each player that determines a draw result. The candidate should not have confused 'throws' with the final total.

Marks awarded for part (i) = 5/5
Marks awarded for part (ii) = 0/1

Total marks awarded = 5 out of 6

PAST PAPERS INSIDE
Your education hub

Example candidate response – middle

4 A game is played between two players:

- they take turns at rolling a six-sided die (numbered 1 to 6) and record their throw
- a player scores 1 point if their throw is higher than their opponent
- they each roll the die 20 times
- if the player's throw is the same as their opponent, the total points is unchanged
- the winner is the player with the larger number of points after 20 throws

The pseudocode will use variable `NoOfThrows` as shown.

Identifier	Data type	Description
`NoOfThrows`	`INTEGER`	Loop control variable

(i) Complete the pseudocode for the given algorithm.

```
FOR i = 1 to 20
  INPUT Player1Throw
  Input Player2Throw
  IF Player1Throw > Player2Throw
    THEN
      Player2Total ← Player2Total + 1
  ENDIF
  IF Player2Throw > Player1Throw
    THEN
      Player2Total ← Player2Total + 1
  ENDIF
Next
IF Player1Total > Player2Total
  THEN
    OUTPUT "Player1 is the winner"
  ELSE
    OUTPUT "Player2 is the winner"
ENDIF
```

[5]

(ii) Identify the game result which will produce incorrect output.

If the socre is tied.

[1]

PAST PAPERS INSIDE Your education hub

Examiner comment – middle

When a loop counter variable is described in the rubric of the question, the pseudocode answer which follows is expected to use this variable.

If the question rubric asks for pseudocode, the candidate should not use keywords that are language specific, such as `NEXT` (as an attempted alternative to `ENDFOR)` or `PRINT` (as an intended alternative to `OUTPUT`).

Marks awarded for part (i) = 3/5
Marks awarded for part (ii) = 1/1

Total marks awarded = 4 out of 6

Example candidate response – low

4 A game is played between two players:

- they take turns at rolling a six-sided die (numbered 1 to 6) and record their throw
- a player scores 1 point if their throw is higher than their opponent
- they each roll the die 20 times
- if the player's throw is the same as their opponent, the total points is unchanged
- the winner is the player with the larger number of points after 20 throws

The pseudocode will use variable `NoOfThrows` as shown.

Identifier	Data type	Description
NoOfThrows	INTEGER	Loop control variable

(i) Complete the pseudocode for the given algorithm.

```
FOR x = 1 to 20
  INPUT Player1Throw
  Input Player2 Throw
  IF Player1Throw > Player2Throw
    THEN
      Player 1 score = Player 1 score + 1
  ENDIF
  IF Player2Throw > Player1Throw
    THEN
      Player2Total ← Player2Total + 1
  ENDIF
  No of Throws = NoOfThrows + 1
IF Player1Total > Player2Total
  THEN
    OUTPUT "Player1 is the winner"
  ELSE
    OUTPUT "Player2 is the winner"
ENDIF
```

[5]

(ii) Identify the game result which will produce incorrect output.

Equal number of wins

[1]

PAST PAPERS INSIDE Your education hub

Examiner comment – low

The given loop counter `NoOfThrows` has not been used.
Note the minor spelling errors of the keyword `INPUT` and identifier `Player1Score` have not been penalised.

Marks awarded for part (i) = 3/5
Marks awarded for part (ii) = 0/1

Total marks awarded = 3 out of 6

Question 5 – parts (a), (b) and (c)

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

(a) Name the data structure to be used in a program for `SalesX`.

.....[2]

PAST PAPERS INSIDE
Your education hub

Question 5 – parts (a), (b) and (c), continued

(b) The programmer writes a program from the following pseudocode design.

```
x ← 0
FOR DayNumber ← 1 TO 7
   IF SalesX[DayNumber] + SalesY[DayNumber] >= 10
      THEN
         x ← x + 1
         OUTPUT SalesDate[DayNumber]
   ENDIF
ENDFOR
OUTPUT x
```

(i) Trace the execution of this pseudocode by completing the trace table below.

x	DayNumber	OUTPUT
0		

[4]

(ii) Describe, in detail, what this algorithm does.

.....

.....

.....

.....[3]

PAST PAPERS INSIDE
Your education hub

Question 5 – parts (a), (b) and (c), continued

(c) The company wants a program to output the total monthly sales for one of the selected websites.

The programmer codes a function with the following function header:

```
FUNCTION MonthlyWebSiteSales(ThisMonth : INTEGER, ThisSite : CHAR)
                                                    RETURNS INTEGER
```

The function returns the total number of bicycles sold for the given month and website.

The function will use the following:

Identifier	Data type	Description
`ThisMonth`	`INTEGER`	Represents the month number e.g. 4 represents April
`ThisSite`	`CHAR`	Coded as: • X for website X • Y for Website Y

(i) Give the number of parameters of this function. .....[1]

(ii) Some of the following function calls may be invalid.

Mark each call with:

- a tick (✓), for a valid call
- a cross (✗), for an invalid call

For any function calls which are invalid, explain why.

Function call	Tick (✓) / cross (✗)	Explanation (if invalid)
`MonthlyWebSiteSales(1, "Y")`		
`MonthlyWebSiteSales(11, 'X', 'Y')`		
`MonthlyWebSiteSales(12, 'X')`		

[3]

PAST PAPERS INSIDE Your education hub

Mark scheme

5 (a) • 1D Array // List [1]

• INTEGER [1]

(b) (i)

x	DayNumber	OUTPUT
0	1	
	2	
1	3	5/6/2015
	4	
2	5	7/6/2015
	6	
3	7	9/6/2015
		3

Note: 'x' and 'output' entries must be on or below the relevant 'DayNumber' entry
Mark as above [4]

(ii)
- … Sales for the first seven days (1)
- … the number of days on which the total sales were **10 or over** (1)
- Outputs the corresponding dates (1)
- Output the final value/total (of x) (1) [max 3]

(c) (i) 2 [1]

(ii)

Tick Cross	Explanation (if invalid)
X // ✓	2nd parameter should be CHAR // accept just tick
X	Three parameters/should be 2 parameters
✓	

[3]

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

(a) Name the data structure to be used in a program for `SalesX`.

Arrays [2]

PAST PAPERS INSIDE
Your education hub

Example candidate response – high, continued

(b) The programmer writes a program from the following pseudocode design.

```
x ← 0
FOR DayNumber ← 1 TO 7
   IF SalesX[DayNumber] + SalesY[DayNumber] >= 10
      THEN
         x ← x + 1
         OUTPUT SalesDate[DayNumber]
   ENDIF
ENDFOR
OUTPUT x
```

(i) Trace the execution of this pseudocode by completing the trace table below.

x	DayNumber	OUTPUT
0	1	
	2	
1	3	05/06/2015
	4	
2	5	07/06/2015
	6	
3	7	09/06/2015 , 3

[4]

(ii) Describe, in detail, what this algorithm does.

This algorithm outputs the ~~no. of~~ Total number of days in a week in which the sales of X and Y were greater than or equal to 10. It also outputs the specific dates on which the sales of X and Y were greater than or equal to ten. [3]

PAST PAPERS INSIDE Your education hub

Example candidate response – high, continued

(c) The company wants a program to output the total monthly sales for one of the selected websites.

The programmer codes a function with the following function header:

```
FUNCTION MonthlyWebSiteSales(ThisMonth : INTEGER, ThisSite : CHAR)
                                                    RETURNS INTEGER
```

The function returns the total number of bicycles sold for the given month and website.

The function will use the following:

Identifier	Data type	Description
`ThisMonth`	`INTEGER`	Represents the month number e.g. 4 represents April
`ThisSite`	`CHAR`	Coded as: • X for website X • Y for Website Y

(i) Give the number of parameters of this function. ....2....[1]

(ii) Some of the following function calls may be invalid.

Mark each call with:

- a tick (✓), for a valid call
- a cross (✗), for an invalid call

For any function calls which are invalid, explain why.

Function call	Tick (✓) /cross (✗)	Explanation (if invalid)
`MonthlyWebSiteSales(1, "Y")`	✓	
`MonthlyWebSiteSales(11, 'X', 'Y')`	✗	The function has only two parameters whereas this one has three.
`MonthlyWebSiteSales(12, 'X')`	✗	' ' are used instead of " "

[3]

PAST PAPERS INSIDE Your education hub

Examiner comment – high

Candidates need to be aware that there are two types of array in the Section 2 syllabus – a one-dimensional and two-dimensional array. Also, an array will be declared with a lower and upper bound and be of a stated data type. The syntax to use for the pseudocode is given in syllabus section 2.2.2. A formal description was not required for this question – just a statement that a 1D array with data type integer would be the most appropriate data structure to use. There were many candidate answers which simply stated 'array' and did not gain credit.

For part (b) the trace shown was an exemplary answer and the candidate carries this through into part (ii) with a clear description of the algorithm.

For part (c) candidates need to realise that it is convention for paper 2 (and Paper 4) that string data will use double quote delimiters and a character value will have single quote delimiters around the single character. Candidates were however given the benefit of the doubt – as shown with this answer – and a statement that this first call was correctly formed was given credit. The explanation for the second call is exemplary with a clear reference to the need for two parameters. The candidate's misunderstanding about the use of single and double quotes is confirmed by their answer for the third procedure call.

Marks awarded for part (a) = 0/2
Marks awarded for part (b) = (i) 4/4, (ii) 3/3
Marks awarded for part (c) = (i) 1/1, (ii) 2/3

Total marks awarded = 10 out of 13

PAST PAPERS INSIDE
Your education hub

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

(a) Name the data structure to be used in a program for `SalesX`.

.....[2]

PAST PAPERS INSIDE
Your education hub

Example candidate response – middle, continued

(b) The programmer writes a program from the following pseudocode design.

```
x ← 0
FOR DayNumber ← 1 TO 7
   IF SalesX[DayNumber] + SalesY[DayNumber] >= 10
      THEN
         x ← x + 1
         OUTPUT SalesDate[DayNumber]
   ENDIF
ENDFOR
OUTPUT x
```

(i) Trace the execution of this pseudocode by completing the trace table below.

x	DayNumber	OUTPUT
0		
0	1	
0	2	
1	3	05/06/2015
1	4	
2	5	07/06/2015
2	6	
3	7	09/06/2015

[4]

(ii) Describe, in detail, what this algorithm does.

if the sales in x plus the Sale in Y More or equal to ten its going to increase Value of x by 1 and output the date

[3]

PAST PAPERS INSIDE Your education hub

Example candidate response – middle, continued

(c) The company wants a program to output the total monthly sales for one of the selected websites.

The programmer codes a function with the following function header:

```
FUNCTION MonthlyWebSiteSales(ThisMonth : INTEGER, ThisSite : CHAR)
                                                    RETURNS INTEGER
```

The function returns the total number of bicycles sold for the given month and website.

The function will use the following:

Identifier	Data type	Description
`ThisMonth`	`INTEGER`	Represents the month number e.g. 4 represents April
`ThisSite`	`CHAR`	Coded as: • X for website X • Y for Website Y

(i) Give the number of parameters of this function. 3 [1]

(ii) Some of the following function calls may be invalid.

Mark each call with:

- a tick (✓), for a valid call
- a cross (✗), for an invalid call

For any function calls which are invalid, explain why.

Function call	Tick (✓) /cross (✗)	Explanation (if invalid)
`MonthlyWebSiteSales(1, "Y")`	✓	
`MonthlyWebSiteSales(11, 'X', 'Y')`	✗	can only call from one site
`MonthlyWebSiteSales(12, 'X')`	✓	

[3]

PAST PAPERS INSIDE Your education hub

Examiner comment – middle

See comments for the high script for part (a).

For part (b) the trace is almost complete – only the output of the total has been omitted. For part (ii) the key word in the question is to 'describe'. What is required therefore is not a simple repeat of the wording of the statement in the algorithm. *The total numbers of sales from Site X and Site Y* is a clear description, the answer given here is a just a pseudocode statement re-word.

For part (c)(ii) the middle row does not gain credit as the candidate is expected to make reference to the number of parameters used.

For part (d) candidates need to have studied the pseudocode syntax given in the syllabus for file handling – Section 2.2.3. Candidates should appreciate that the file is opened for one of three possible 'modes'. That is, to write data to the file, read data from an existing file or to append new data to an existing file. Once the pseudocode has been introduced, teachers should encourage students to put this into practice in their chosen programming language.

Marks awarded for part (a) = 0/2
Marks awarded for part (b) = (i) 3/4, (ii) 1/3
Marks awarded for part (c) = (i) 0/1, (ii) 2/3

Total marks awarded = 6 out of 13

PAST PAPERS INSIDE
Your education hub

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

(a) Name the data structure to be used in a program for `SalesX`.

Stack ..... [2]

PAST PAPERS INSIDE
Your education hub

Example candidate response – low, continued

(b) The programmer writes a program from the following pseudocode design.

```
x ← 0
FOR DayNumber ← 1 TO 7
   IF SalesX[DayNumber] + SalesY[DayNumber] >= 10
      THEN
         x ← x + 1
         OUTPUT SalesDate[DayNumber]
   ENDIF
ENDFOR
OUTPUT x
```

(i) Trace the execution of this pseudocode by completing the trace table below.

x	DayNumber	OUTPUT
0	1	
1	2	
2	3	
3	4	
4	5	
5	6	
6	7	

[4]

(ii) Describe, in detail, what this algorithm does.

If the total sales from SalesX and SalesY is equal or bigger than 10, +1 is added to the counter 'x'. I suppose counter 'x' can be something similar to a Break even Point of the business [3]

(c) The company wants a program to output the total monthly sales for one of the selected websites.

The programmer codes a function with the following function header:

```
FUNCTION MonthlyWebSiteSales(ThisMonth : INTEGER, ThisSite : CHAR)
                                                    RETURNS INTEGER
```

The function returns the total number of bicycles sold for the given month and website.

The function will use the following:

Identifier	Data type	Description
ThisMonth	INTEGER	Represents the month number e.g. 4 represents April
ThisSite	CHAR	Coded as: • X for website X • Y for Website Y

(i) Give the number of parameters of this function. ....12.... [1]

(ii) Some of the following function calls may be invalid.

Mark each call with:

- a tick (✓), for a valid call
- a cross (✗), for an invalid call

For any function calls which are invalid, explain why.

Function call	Tick (✓) /cross (✗)	Explanation (if invalid)
`MonthlyWebSiteSales(1, "Y")`	✓	
`MonthlyWebSiteSales(11, 'X', 'Y')`	✓	
`MonthlyWebSiteSales(12, 'X')`	✓	

[3]

PAST PAPERS INSIDE Your education hub

Examiner comment – low

This is a weak answer for all three parts.

For part (a) see the comments for the high script about the use and description of arrays.

In part (b) the candidate has correctly iterated through the `FOR` loop to show the `DayNumber` values, but the corresponding `x` values in column 1 are incorrect.

In part (b)(ii) the key points to describe the algorithm have not been mentioned.

For part (c)(ii) see the comment made for the Grade A script concerning the use of double and single quote delimiters.

Marks awarded for part (a) = 0/2
Marks awarded for part (b) = (i) 1/4, (ii) 1/3
Marks awarded for part (c) = (i) 0/1, (ii) 2/3

Total marks awarded = 4 out of 13

PAST PAPERS INSIDE
Your education hub

Question 5 – parts (d) and (e)

(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.

```
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 <Space> 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.

PAST PAPERS INSIDE Your education hub

Question 5 – parts (d) and (e), continued

The following incomplete pseudocode creates the text file DISCOUNT_DATES.

Complete the pseudocode.

```
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]

(e) The DISCOUNT_DATES text file is successfully created.

The company now wants a program to:

- key in a date entered by the user
- search the text file for this date
- if found, output one of the following messages:
  - "No discount on this date"
  - "This is a discount date"
- if not found, output "Date not found"

(i) Add to the identifier table to show the variables you need for this new program.

Identifier	Data type	Description
DISCOUNT_DATES	FILE	Text file to be used

[3]

PAST PAPERS INSIDE
Your education hub

Question 5 – parts (d) and (e), continued

(ii) Write the **program code**.

Do not include any declaration or comment statements for the variables used.

Programming language ....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[7]

Mark scheme

(d)
```
OPENFILE "DISCOUNT_DATES" FOR WRITE / WRITING      (1)
INPUT NextDate                                     (1)
WHILE  NextDate <> "XXX"
    INPUT Discount
    NextLine = CONCAT(NextDate, " ", Discount)     (1)
    WRITEFILE "DISCOUNT_DATES",  NextLine
ENDWHILE                                           (1)
OUTPUT "File now created"
CLOSEFILE                                                [4]
```

(e) (i) Sensible Identifier + Data Type + Description (1 + 1 + 1)

For example:

`ThisDate`	`STRING/DATE`	date 'entered by user'
`Found`	`BOOLEAN`	flag to indicate `ThisDate` is 'present in the file'
`NextLine`	`STRING`	a single line 'from the text file'
`NextDate`	`STRING/DATE`	date 'from next line in the file'
`NextDiscount`	`STRING`	the discount value from `NextLine`
`ThisMonth`	`INTEGER`	the month part of the date (input or from file)
`MyStreamReader`	`STREAMREADER`	references `DISCOUNT_DATES` file

Reject 'generic' reserved words
Allow **one** instance variable to store output string(s)
Allow **one** instance of month/day/year number e.g. `ThisMonth` shown above [3]

(ii) *Mark as follows:*

Open file statement	(1)
File read statement for line text – `NextLine`	(1)
File close statement	(1)
Input of the required date – `ThisDate`	(1)
Isolate `NextDate` from `NextLine`	(1)
Isolate `NextDiscount` from `NextLine`	(1)
IF statement comparing the two dates	(1)
Uses Boolean variable `Found` to flag when found	(1)
Post/pre condition loop iterate through the file	(1)
Test for EOF or 'found'	(1)
Note: These must follow some correct logic to score …	
Output 'No discount on this date' **and** Output 'This is a discount date')	(1)
Output (when date not found) 'Date not found'	(1)

Accept 'any' identifier names [max 7]

PAST PAPERS INSIDE Your education hub

The following incomplete pseudocode creates the text file DISCOUNT_DATES.

Complete the pseudocode.

```
OPENFILE "DISCOUNT_DATES" FOR Write
INPUT NextDate
WHILE  NextDate <>"XXX"
   INPUT Discount
   Discountdate = CONCAT(NextDate, " ", Discount)
   WRITEFILE "DISCOUNT_DATES",  NextLine
   INPUT NextDate
END WHILE
OUTPUT "File now created"
CLOSEFILE
```

[4]

(e) The DISCOUNT_DATES text file is successfully created.

The company now wants a program to:

- key in a date entered by the user
- search the text file for this date
- if found, output one of the following messages:
  - "No discount on this date"
  - "This is a discount date"
- if not found, output "Date not found"

(i) Add to the identifier table to show the variables you need for this new program.

Identifier	Data type	Description
DISCOUNT_DATES	FILE	Text file to be used
Sr DiscountDates	FILE	Open file opened in reading mode
Date	String	Date entered by user
found	Boolean	To know if date is found in file or not.
datediscount	string	one line read from file
Length	Integer	Length of string datediscount
Discount	String	Discount's state from "datediscount"
Date saved	Integer	date from "datediscount"

[3]

PAST PAPERS INSIDE Your education hub

Example candidate response – high, continued

(ii) Write the **program code.**

Do not include any declaration or comment statements for the variables used.

Programming language Visual Basic

```
Sr Discount_Dates = IO.FILE.OPENTEXT("DISCOUNT_DATES.txt")
Console.writeline ("Enter Date")
Date = console.readline
While NOT SrDiscount_Dates.File.Peek=-1
Found = False
Do WHILE NOT SrDiscount_Dates.File.PEEK=-1 AND
'FOUND = FALSE
datediscount= console.readline
Length = len(datediscount)
Discount = Right(datediscount, (length-1))(length-1))
Date saved = Left(datediscount, 6)
IF datesaved = date THEN
    Found = True
Endif
EndWhile
IF Found = True THEN
    IF Discount = "True"
        console.writeline ("This is a discount date")
    Else
        console.writeline ("No discount on this date")
    Endif
Else
    console.writeline ("Date not found")
End if
```

[7]

Examiner comment – high

Part (d) is a sound answer with the candidate understanding the use of a file for reading data from it. The candidate has 'made up' their own identifier name on line 5. This was not necessary as it can be deduced from the statement given on the next line. This is a good example where the best prepared candidates will have been those who had been exposed by the pre-release materials with the `CONCAT` function being covered.

The intention for the final part (e) was to get the candidate to consider – before attempting to write any code – what variables would be needed for the task. This is a sound answer with the candidate using (mostly) appropriate identifier names, the correct data type and a clear description. `Date` as an identifier name was not considered appropriate.

This is an excellent answer for the program code. All aspects of the task have been correctly designed and coded. The candidate clearly has 'looked back' to the answers they gave for part (i) and used the same identifiers in the code for part (ii).

Marks awarded for part (d) = 3/4
Marks awarded for part (e) = (i) 3/3, (ii) 7/7

Total marks awarded = 13 out of 14

PAST PAPERS INSIDE
Your education hub

Example candidate response – middle

The following incomplete pseudocode creates the text file DISCOUNT_DATES.

Complete the pseudocode.

```
OPENFILE "DISCOUNT_DATES" FOR READ
INPUT NextDate
WHILE  NextDate <>"XXX"
   INPUT Discount
   ..... = CONCAT(NextDate, " ", Discount)
   WRITEFILE "DISCOUNT_DATES",  NextLine
   INPUT NextDate
ENDWHILE
OUTPUT "File now created"
CLOSEFILE
```

[4]

(e) The DISCOUNT_DATES text file is successfully created.

The company now wants a program to:

- key in a date entered by the user
- search the text file for this date
- if found, output one of the following messages:
  - "No discount on this date"
  - "This is a discount date"
- if not found, output "Date not found"

(i) Add to the identifier table to show the variables you need for this new program.

Identifier	Data type	Description
DISCOUNT_DATES	FILE	Text file to be used
KEYDATE		
DAY	INTEGER	Day part of date entered by user
MONTH	INTEGER	Month part of date entered by user
YEAR	INTEGER	Year part of date entered by user
FOUND	BOOLEAN	Flag true if data match and false when not
2DAY	INTEGER	Day part of date searched in file
2MONTH	INTEGER	Month part of date Searched in file
2YEAR	INTEGER	Year part of date searched in file.

[3]

PAST PAPERS INSIDE
Your education hub

Examiner comment – middle

The script has secured half the marks for the file handling pseudocode. Candidates need to be clear about the difference between opening a file in 'read' or 'write' mode.

Part (e)(i) shows a good attempt at identifying the data requirements for the task. `Year` was a questionable choice for an identifier name but has been given credit.

The candidate has demonstrated some sound understanding of some of the key statements in Visual Basic – for example, creating a 'streamreader' object and the input from the user of the required date. However this needs to be complemented with the design of the algorithm for the given task.

Marks awarded for part (d) = 2/4
Marks awarded for part (e) = (i) 2/3, (ii) 3/7

Total marks awarded = 7 out of 14

PAST PAPERS INSIDE
Your education hub

Example candidate response – low

The following incomplete pseudocode creates the text file DISCOUNT_DATES.

Complete the pseudocode.

```
OPENFILE "DISCOUNT_DATES" FOR 1 to EOF
INPUT Textfile
WHILE  NextDate <>"XXX"
   INPUT Discount
   ..... = CONCAT(NextDate, " ", Discount)
   WRITEFILE "DISCOUNT_DATES",  NextLine
   INPUT NextDate
.....
OUTPUT "File now created"
CLOSEFILE
```

[4]

(e) The DISCOUNT_DATES text file is successfully created.

The company now wants a program to:

- key in a date entered by the user
- search the text file for this date
- if found, output one of the following messages:
  - "No discount on this date"
  - "This is a discount date"
- if not found, output "Date not found"

(i) Add to the identifier table to show the variables you need for this new program.

Identifier	Data type	Description
DISCOUNT_DATES	FILE	Text file to be used
Discount	Boolean	is there a discount
Input_Date	String	the date want to search

[3]

PAST PAPERS INSIDE
Your education hub

Examiner comment – low

The attempt at part (d) shows no awareness of the pseudocode shown in the syllabus for file handling – section 2.2.3.

In part (e)(i) a good attempt has been made at identifying the data requirements for the task. As the questions shows a total of three marks, the candidate should have anticipated that at least three variables would need to be shown for maximum marks.

In part (e)(ii) there is some good evidence of knowledge of the syntax of certain Pascal statements and these have been credited. However, the candidate needs to appreciate that some of the marks will be awarded for demonstrating a design that works for the given task.

Marks awarded for part (d) = 0/4
Marks awarded for part (e) = (i) 2/3, (ii) 2/7

Total marks awarded = 4 out of 14

PAST PAPERS INSIDE
Your education hub

Cambridge International Examinations
1 Hills Road, Cambridge, CB1 2EU, United Kingdom
tel: +44 1223 553554 fax: +44 1223 553558
email: info@cie.org.uk www.cie.org.uk

A DIVISION OF
PAST PAPERS
INSIDE
Your education hub