Cambridge International AS & A Level

Cambridge International Examinations
Cambridge International Advanced Subsidiary and Advanced Level

CANDIDATE NAME									
CENTRE NUMBER						CANDIDATE NUMBER			

5509243651

COMPUTER SCIENCE **9608/21**

Paper 2 Fundamental Problem-solving and Programming Skills **May/June 2015**

2 hours

Candidates answer on the Question Paper.

No Additional Materials are required.

No calculators allowed.

READ THESE INSTRUCTIONS FIRST

Write your Centre number, candidate number and name in the spaces at the top of this page.
Write in dark blue or black pen.
You may use an HB pencil for any diagrams, graphs or rough working.
Do not use staples, paper clips, glue or correction fluid.
DO **NOT** WRITE IN ANY BARCODES.

Answer **all** questions.
No marks will be awarded for using brand names of software packages or hardware.

At the end of the examination, fasten all your work securely together.
The number of marks is given in brackets [ ] at the end of each question or part question.

The maximum number of marks is 75.

This document consists of **19** printed pages and **1** blank page.

© UCLES 2015

CAMBRIDGE International Examinations

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]

(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]

(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]

© UCLES 2015

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]

(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]

(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]

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

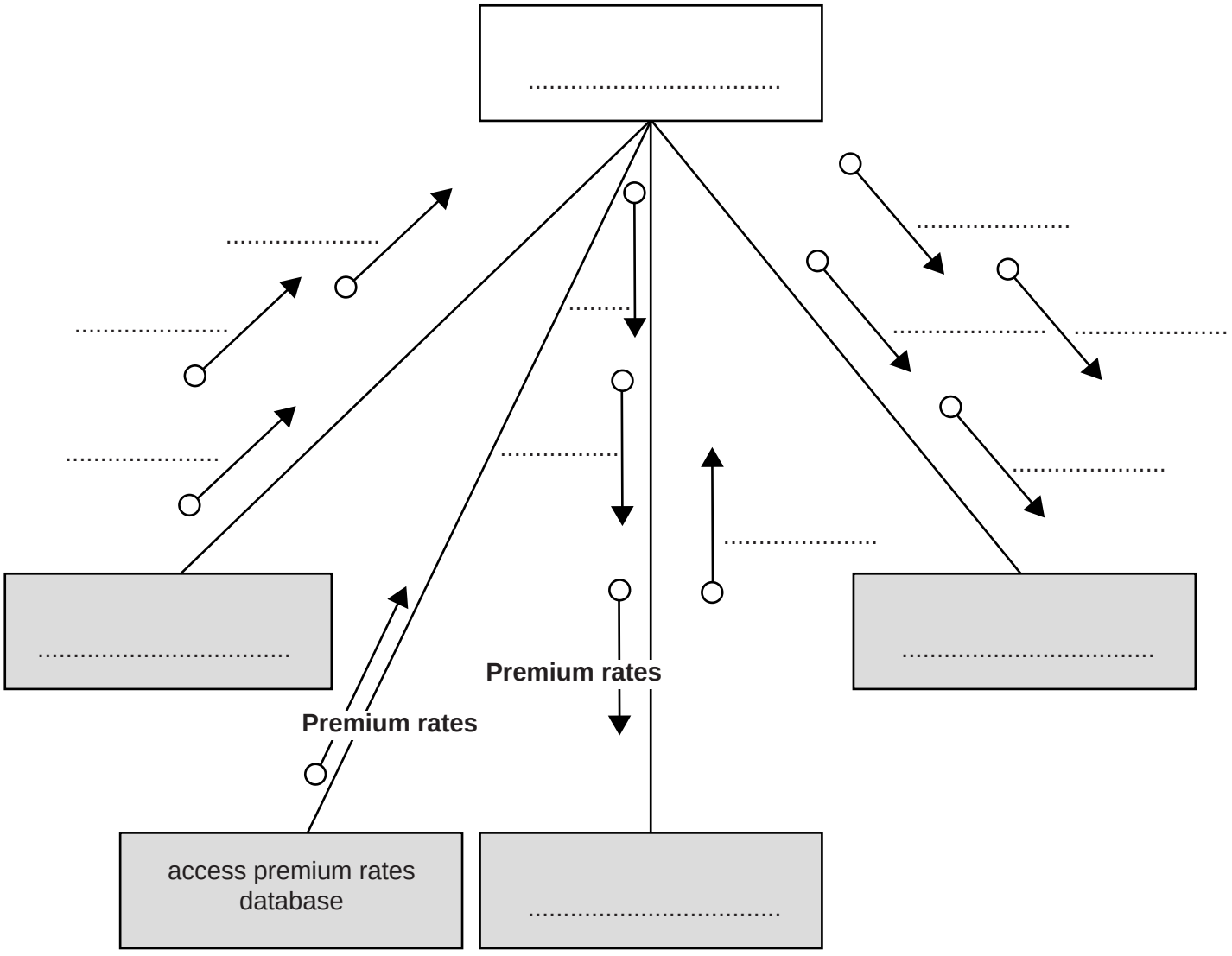
The price of the insurance is calculated from:

- 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:



(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]

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]

Question 5 begins on page 12.

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]

(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]

(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]

(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.

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]

Question 5(e) continues on page 18.

(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:
    - o "No discount on this date"
    - o "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]

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

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

Programming language .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[7]

BLANK PAGE

Permission to reproduce items where third-party owned material protected by copyright is included has been sought and cleared where possible. Every reasonable effort has been made by the publisher (UCLES) to trace copyright holders, but if any items requiring clearance have unwittingly been included, the publisher will be pleased to make amends at the earliest possible opportunity.

To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced online in the Cambridge International Examinations Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at www.cie.org.uk after the live examination series.

Cambridge International Examinations is part of the Cambridge Assessment Group. Cambridge Assessment is the brand name of University of Cambridge Local Examinations Syndicate (UCLES), which is itself a department of the University of Cambridge.