Cambridge International AS & A Level

Cambridge International Examinations
Cambridge International Advanced Subsidiary and Advanced Level

CANDIDATE NAME

CENTRE NUMBER | | | | | |

CANDIDATE NUMBER | | | | |

COMPUTER SCIENCE **9608/23**

Paper 2 Fundamental Problem-solving and Programming Skills **May/June 2018**

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 **15** printed pages and **1** blank page.

CAMBRIDGE International Examinations

1 **(a)** A program controls the heating system of an energy-efficient house.

Give a suitable **identifier name** for each of the data items.

Description of data item	Suitable identifier name
The temperature inside the house	
The temperature outside the house	
The wind speed	
Whether it was raining or not	

[4]

(b) **(i)** Program variables have values as follows:

Variable	Value
`Quality`	`'D'`
`DayNumber`	`20`
`MyName`	`"Stephen"`
`QualityConfirmed`	`TRUE`
`Factor`	`6.5`

Evaluate each expression in the following table.

If an expression is invalid, write ERROR.

For the built-in functions list, refer to the **Appendix** on page 15.

Expression	Evaluates to
`MID(MyName, 4, 4) & "ol"`	
`QualityConfirmed AND (Factor >= 6.5)`	
`20 + ASC(Quality)`	
`QualityConfirmed + 3`	
`MOD(Factor * 2, 9)`	

[5]

(ii) Programming languages support different data types.

Give an appropriate data type for each of these variables from **part (b)(i)**.

Variable	Data type
`QualityConfirmed`	
`DayNumber`	
`Factor`	
`Quality`	
`MyName`	

[5]

2 The following is a function design in pseudocode.

Line numbers are given for reference only.

```
01  FUNCTION GradeCalc(Mark : ARRAY[] OF INTEGER) RETURNS INTEGER
02
03    DECLARE DGradeCount : INTEGER
04    DECLARE n : INTEGER
05    DECLARE Grade : STRING
06    DECLARE ThisMark : INTEGER
07
08    DGradeCount ← 0    // initialise variables
09    n ← 1
10    ThisMark ← 1
11    REPEAT
12       ThisMark ← Mark[n]    // get next mark from Mark array
13
14       IF ThisMark > 74
15          THEN
16             Grade ← "Distinction"
17             DGradeCount ← DGradeCount + 1 // one more Distinction
18          ELSE
19             IF ThisMark > 59
20                THEN
21                   Grade ← "Merit"
22                ELSE
23                   IF ThisMark > 39
24                      THEN
25                         Grade ← "Pass"
26                      ELSE
27                         Grade ← "Fail"
28                   ENDIF
29             ENDIF
30       ENDIF
31
32       OUTPUT "Grade for Student " & n & " is " & Grade
33
34       n ← n + 1
35    UNTIL n = 101 // no more elements in the array
36
37    RETURN Grade // return the number of Distinction grades
38
39  ENDFUNCTION
40
41  DECLARE Mark : ARRAY[1:100] OF INTEGER
42  DECLARE DistinctionGrades : INTEGER
43
44  CALL GetMarks(Mark[])  // fill the array with student marks
45  DistinctionGrades ← GradeCalc(Mark[])
```

(a) This pseudocode includes features that are examples of good practice.

Explain why it is useful to include the two following features.

Comments .....

.....

Indentation .....

.....

[2]

(b) Study the function `GradeCalc()`. Identify the features of the function in the following table.

Feature	Answer
A line number containing an example of an integer assignment statement	
A line number containing the start of a selection structure	
A line number containing the end of a selection structure	
The upper bound of the `Mark` array	
The number of dimensions of the `Mark` array	
The name for the type of loop structure used	
A line number containing an unnecessary assignment statement	
The number of times `OUTPUT` is called	
The number of local variables	

[9]

(c) **(i)** There is a mistake in the pseudocode that would produce a data type mismatch error if a programmer were to write similar program code.

Describe this mistake and how it may be corrected.

.....

.....

.....

.....[2]

(ii) Lines 14 to 30 of the pseudocode on page 4 could be written using a `CASE` structure.

Write the equivalent `CASE` structure in **pseudocode**.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

3 A modular program design consists of four modules:

`Module1` has three sub-tasks. Each sub-task is implemented by a single subroutine (either a function or a procedure).

The subroutine headings are defined as follows:

```
FUNCTION  Module2 (Weight : REAL) RETURNS BOOLEAN
PROCEDURE Module3 (Weight : REAL, Customer : STRING, Purchased : DATE)
FUNCTION  Module4 (Purchased : DATE, Account : INTEGER) RETURNS INTEGER
```

(a) State the term given to values passed between modules.

.....[1]

(b) Draw a structure chart to represent the program design.

Use the letters in the table to label the values passed between modules.

Value	Label
Boolean return value	A
Integer return value	B
`Account`	C
`Customer`	D
`Purchased`	E
`Weight`	F

Structure chart

[4]

4 A program controls a chemical process in a factory.

The temperature is monitored as part of the control process. The temperature is measured at fixed time intervals and the value is stored in an array, `PTemp`. The array contains 100 elements, representing 100 temperature values. The first element is `PTemp[1]`.

The program will check whether the temperature is outside the acceptable range more than 20 times. This task is performed by a function, `IsTempOK()`.

The algorithm for the function `IsTempOK()` is expressed in structured English as follows:

1 Examine each array element and count the number of times that a temperature is less than `MinTemp` or more than `MaxTemp`.
2 If the count in step 1 exceeds 20, return `FALSE`, otherwise return `TRUE`.

Draw a program flowchart, on the next page, to represent the algorithm for the function `IsTempOK()`.

Assume:

- the array contains 100 valid temperature values
- `PTemp, MinTemp` and `MaxTemp` are global variables.

Note that variable declarations are not required in program flowcharts.

[9]

5 A golf club holds information about its members. When a member completes a round of golf, their score is stored along with their membership number and the date of the round.

A program is to be written to store and process the score information.

The information to be stored is as follows:

- `MembershipNumber` is a four-digit numeric string.
- `Date` is a six-digit numeric string in the format DDMMYY
- `Score` is a two-digit numeric string in the range "`50`" to "`99`".

This information is stored in a single string in the format:

```
<MembershipNumber><Date><Score>
```

(a) (i) The program designer considers storing the strings in an array.

State how many dimensions the array would require.

.....[1]

(ii) The designer has decided to store the strings in a file.

Give a reason for this choice.

.....

.....[1]

(b) The program will be developed using an Integrated Development Environment (IDE). One feature provided by an IDE is known as prettyprint.

Name **two** presentation features provided by prettyprint.

Feature 1 .....

.....

Feature 2 .....

.....
[2]

(c) The strings are stored in a text file, `ScoreDetails.txt`.

The program needs a function, `GetAverageScore()`.

The structured English representing the algorithm for this function is as follows:

1 Receive a membership number as its argument.
2 Search the `ScoreDetails.txt` file for all scores for that member.
3 Calculate the average score.
4 Return the average score.

You can assume that there will be at least one score relating to each member.

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

Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type.

Programming language .....

Program Code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[10]

6 **(a)** Individual elements in a 1D array are referenced using an integer variable.

For example, using the variable `n`, an individual element would be referenced in pseudocode as:

```
StudentGrade[n]
```

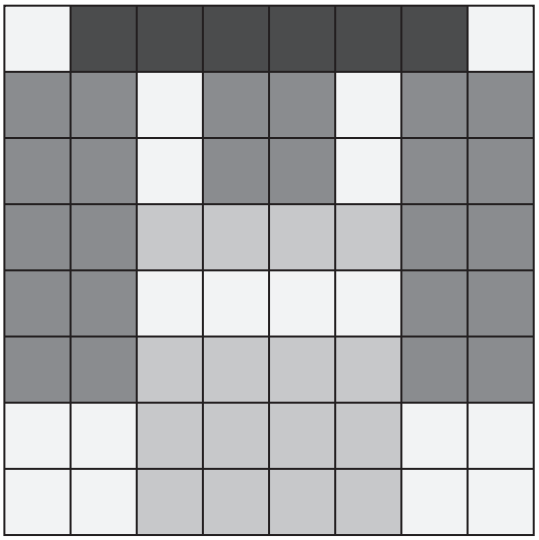
Give the correct technical term for the variable `n`.

.....[1]

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

The following is an example of an image and the corresponding data values for the `Picture` array.

Bitmap image



Values

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

In pseudocode, the array is declared as follows:

```
DECLARE Picture : ARRAY[1:8, 1:8] OF INTEGER
```

A function, `Clip()`, is required that will:

- take an integer parameter, `MaxVal`, to represent the maximum allowed value for each pixel
- limit each pixel value to the maximum allowed value
- return `TRUE` if any of the pixel values have been changed, otherwise return `FALSE`

An example use of the function is:

```
IsClipped ← Clip(234)
```

This will limit the maximum pixel value within the array to 234. Pixels with a value not greater than 234 will remain unchanged.

13

Write **pseudocode** to implement the `Clip()` function.

Assume that `Picture` is a global variable.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[9]

7 A function, `IsFactor()` takes two integer parameters and checks whether the second parameter is a factor of the first parameter.

For example `IsFactor(6,3)` would return `TRUE` as six divided by three has a remainder of zero.

This function returns `FALSE` if there is a non-zero remainder, or if the value of the second parameter is zero.

Write **program code** to implement the `IsFactor()` function.

Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type.

Programming language .....

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[6]

Appendix

Built-in functions (pseudocode)

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

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

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

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

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

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

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

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

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

`INT(x : REAL) RETURNS INTEGER`
returns the integer part of `x`

Example: `INT(27.5415)` returns `27`

`ASC(ThisChar : CHAR) RETURNS INTEGER`
returns the ASCII value of character `ThisChar`

Example: `ASC('A')` returns `65`

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

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

Operators (pseudocode)

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

BLANK PAGE