



UNIVERSITY OF CAMBRIDGE INTERNATIONAL EXAMINATIONS
General Certificate of Education
Advanced Subsidiary Level and Advanced Level

CANDIDATE
NAME

CENTRE
NUMBER

--	--	--	--	--

CANDIDATE
NUMBER

--	--	--	--



COMPUTING

9691/21

Paper 2

October/November 2013

2 hours

Candidates answer on the Question Paper.

No additional materials are required.

READ THESE INSTRUCTIONS FIRST

Write your Centre number, candidate number and name on all the work you hand in.

Write in dark blue or black pen.

You may use a soft pencil for any diagrams, graphs or rough working.

Do not use staples, paper clips, highlighters, glue or correction fluid.

DO **NOT** WRITE IN ANY BARCODES.

Answer **all** questions.

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.

This document consists of **14** printed pages and **2** blank pages.



- 1 The Computing Department has a problem keeping track of its teaching resources. A student, has been asked to design and program a solution as his computing project. It may be the first large problem he has worked on.

He intends to write one large program that follows the process right through. His teacher tells him to break the problem into smaller parts.

- (a) State and justify **three** of the reasons his teacher could give him for breaking the problem into smaller parts.

Reason 1

.....

.....

Reason 2

.....

.....

Reason 3

.....

..... [6]

Juan decides that the design will include the following modules:

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

(b) (i) Describe **one** diagrammatic method for showing how these modules are related.

.....

.....

.....

..... [2]

(ii) Use your method with the modules above.

[2]

- (c) Name **two** features of a high-level programming language that help with this modular approach.

1
2 [2]

- (d) Juan realises that he will have to pass data from one module to another.

Explain how this is done.

.....
.....
.....
..... [2]

- For
miner's

1

.....

.....

2

.....

.....

3 [6]

- In a high-level programming language:

- Programming language

Code

[4]

- 2 (a) Juan has little programming experience, but has to write code for this program. He has written the following pseudocode statements.

For each statement describe what is wrong and write a correct version.

- (i) IF Index > 5000 OR < 0 THEN OUTPUT "Error"

Description

.....

Correct statement

..... [2]

- (ii) NumberOfCopies[Index] + 1 ← NumberOfCopies[Index]

Description

.....

Correct statement

..... [2]

- (iii) NumberOfCopies[Index] ← "three"

Description

.....

Correct statement

..... [2]

(b) Juan needs to design the code for a part of the program that determines resources are kept. If the resource has:

- Write pseudocode that processes the variable `ResourceID` and outputs where the resource is kept. Use nested IF statements.

[6]

- (c) In a high-level programming language write code that processes the value of `ResourceID` and outputs where the resource is kept. Use a CASE/SELECT statement.

Programming language

Code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[6]

For
Trainer's

- enter the name of the resource
- choose the type of the resource
- select the purchase date

- a new resource ID
- where the resource is kept

Design a graphical user interface (GUI). Pay particular attention to layout and effective use of the controls you would expect to find in a GUI.

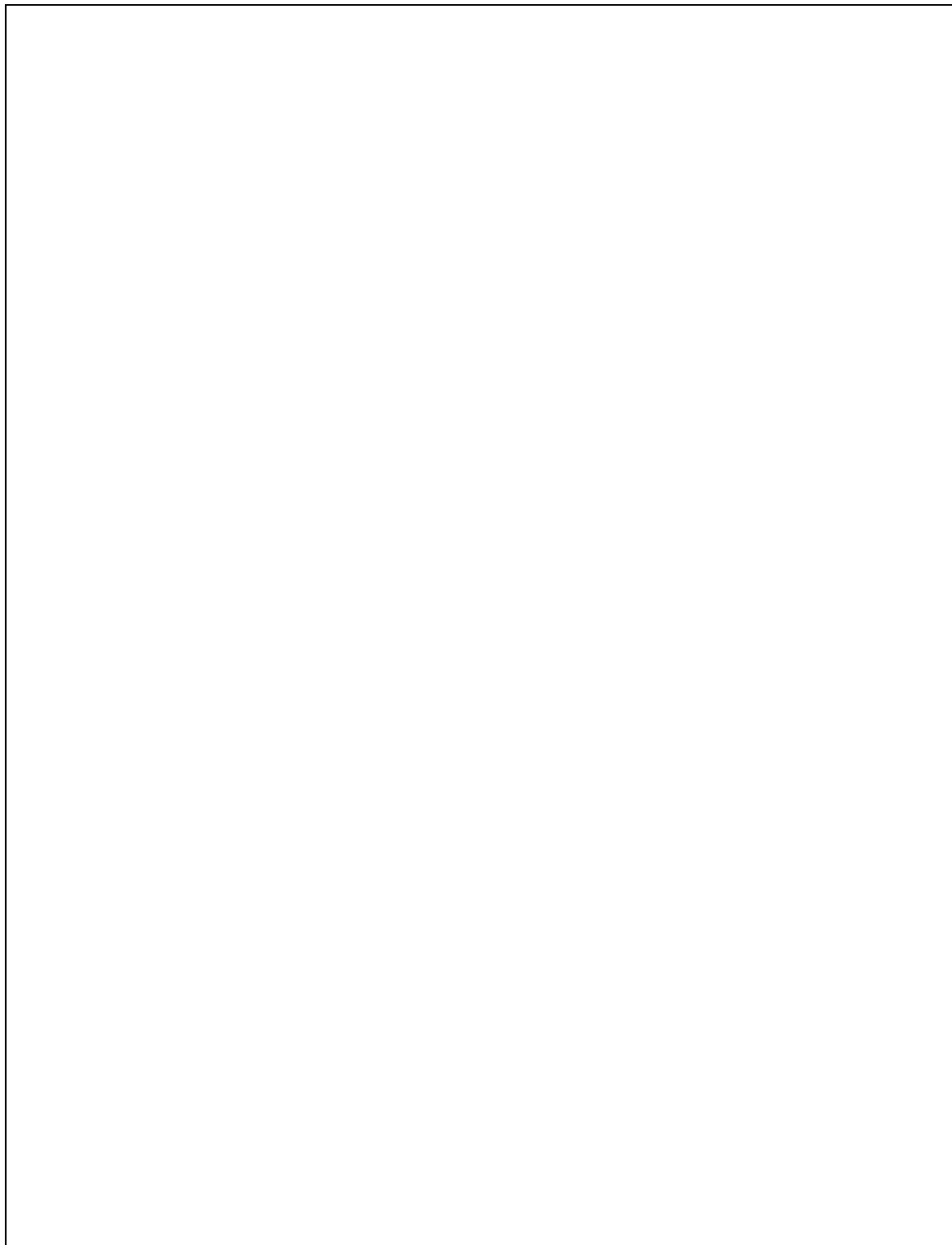
Add new resource(s)

[6]

- (b) A report will show all the items that have been entered on a particular day.

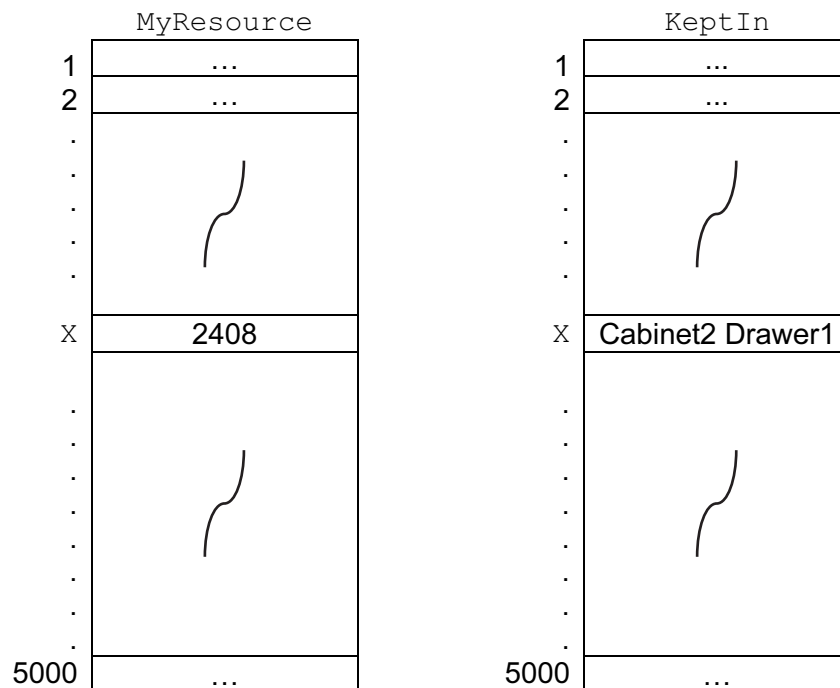
It will show each resource ID and where the resource is kept. The resources will be grouped by type.

Design the report layout.



[5]

An array, `MyResource`, size 5000, data type `INTEGER`, is used to store the resource ID.
 An array, `KeptIn`, size 5000, data type `STRING`, is used to store where a resource is kept.
 A resource with resource ID `MyResource[X]` is kept at `KeptIn[X]`, where `X` is an integer variable.



Juan writes the pseudocode that searches `MyResource` for a given resource ID and outputs where the resource is kept.

```

flag ← 0
INPUT P
FOR X ← 1 TO 5000
  IF myresource[X] = P
  THEN
    OUTPUT keptin[X]
    flag ← 1
  ENDIF
NEXT
IF flag = 0
  THEN
    OUTPUT "Not Found"
  ENDIF

```

For miner's

- | | |
|---|--|
| 1 | |
| 2 | |
| 3 | |
| 4 | |

- [6]

- (e) (i) If Juan were to perform a dry run on the pseudocode using a trace table, state a type of error he might find. Give an example.

Type 1

Example

State another type of error he might find later. Give an example.

Type 2

Example [4]

- (ii) Juan has written his program and one module appears not to work.

Explain how Juan can use a break point and stepping to debug his program.

.....

.....

.....

.....

.....

..... [3]

- 4 Juan is also learning about recursion. He writes the pseudocode for a recursive function.

```

1  FUNCTION Add(N)
2      DECLARE R
3      IF N <= 0
4          THEN
5              R ← 0
6          ELSE
7              R ← N + Add(N - 1)
8          ENDIF
9      RETURN R
10 ENDFUNCTION

```

- (a) What is the scope of the variable R?

..... [1]

- (b) State the line number which shows that this function is recursive.

..... [1]

- (c) List the function calls that are generated by an initial call of Add(3).

.....

 [3]

