Cambridge
O Level

Cambridge International Examinations
Cambridge Ordinary Level

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

COMPUTER SCIENCE **2210/22**

Paper 2 Problem-solving and Programming **May/June 2016**

1 hour 45 minutes

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.

DO NOT ATTEMPT TASKS 1, 2 AND 3 in the pre-release material; these are for information only.

You are advised to spend no more than **40 minutes** on **Section A** (Question 1).

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 50.

DC (LM/SW) 124612

Section A

You are advised to spend no longer than 40 minutes answering this section.

Here is a copy of the pre-release material.

DO NOT attempt Tasks 1, 2 and 3 now.
Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release Material

The manager of a building materials delivery service needs a program to check the contents and weight of sacks to ensure that correct orders are made up for delivery. A price for the order will be calculated.

Write and test a program for the manager.

- Your program must include appropriate prompts for the entry of data.
- Error messages and other output need to be set out clearly.
- All variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

TASK 1 – Check the contents and weight of a single sack

Each sack must obey the following rules to be accepted:

- contain cement, gravel or sand, with a letter on the side for easy identification
  - C - cement
  - G - gravel
  - S - sand
- sand or gravel must weigh over 49.9 and under 50.1 kilograms
- cement must weigh over 24.9 and under 25.1 kilograms

Input and store the weight and contents for one sack. The contents must be checked and an incorrect sack rejected. The weight must be validated on entry and an overweight or underweight sack rejected.

Output the contents and weight of an accepted sack. If a sack is rejected, output the reason(s).

TASK 2 – Check a customer's order for delivery

Input and store the number of sacks of each type required for the order. Use TASK 1 to check the contents and weight of each sack. Ensure that the delivery contains the correct number and type of sacks for the order.

Output the total weight of the order.
Output the number of sacks rejected from the order.

TASK 3 – Calculate the price for a customer's order

Extend TASK 2 to calculate a price for an order. Prices for the sacks are as follows:

- regular price for each sack
  - cement, $3
  - gravel, $2
  - sand, $2
- discount price for a special pack containing 1 sack of cement, 2 sacks of sand and 2 sacks of gravel, $10

Calculate and output the regular price for the order. Check how many special packs are in the order. If a discount price applies then output the new price for the order and the amount saved.

1 **(a)** All variables, constants and other identifiers should have meaningful names.

(i) For **three** of the variables that you have used in **Task 2**, state the name, type and its use.

Variable 1 name .....

Type .....

Use .....

Variable 2 name .....

Type .....

Use .....

Variable 3 name .....

Type .....

Use .....[3]

(ii) State **three** constants that you could have used for **Task 1**. Give the value that would be assigned to each one.

Constant 1 name .....

Value 1 .....

Constant 2 name .....

Value 2 .....

Constant 3 name .....

Value 3 .....[3]

(b) Write an algorithm to complete **Task 2**, using **either** pseudocode, programming statements **or** a flowchart. You can assume that the weight and contents have already been checked and accepted in Task 1. You do not need to output the number of sacks rejected.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[5]

(c) **(i)** Give **two** different data values that could be used to check your validation rules for sand in **Task 1**. Explain why you chose each value.

Sand data value 1 .....

Reason for choice .....

.....

Sand data value 2 .....

Reason for choice .....

.....[2]

(ii) Give **two** different data values that could be used to check your validation rules for cement in **Task 1**. Explain why you chose each value.

Cement data value 1 .....

Reason for choice .....

.....

Cement data value 2 .....

Reason for choice .....

.....[2]

(d) Explain how your program calculates the price for an order (**Task 3**). You may include programming statements as part of your explanation.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[5]

Section B

2 Read this section of program code that inputs 10 positive numbers and then outputs the total.

```
1   Total = 0
2   Counter = 0
3   REPEAT
4     INPUT Num
5     Total = Total + Num
6     PRINT Total
7     Counter = Counter + 1
8   UNTIL Counter = 10
```

This code works, but it is inefficient.

(i) Suggest **three** improvements that could be made.

1.....

.....

2.....

.....

3.....

.....[3]

(ii) Rewrite the program code with your improvements.

.....

.....

.....

.....

.....

.....

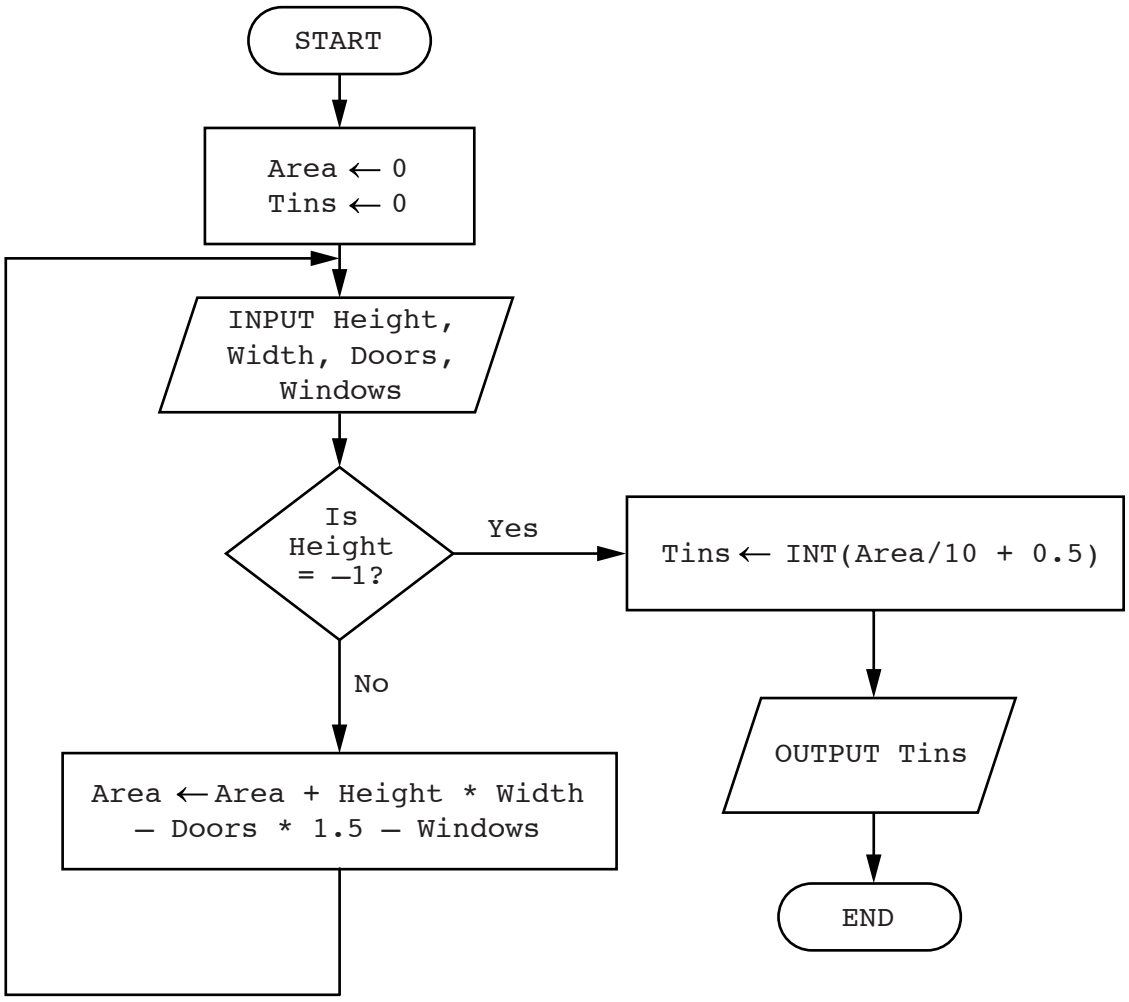
.....

.....

.....

.....[3]

3 The flowchart below calculates the number of tins of paint required to paint walls. The flowchart inputs the height and width of a wall in metres, the number of doors and the number of windows. A value of –1 for the height stops the input.



Complete the trace table for the input data:

3, 5, 1, 0, 3, 7, 0, 0, 3, 5, 0, 3, 3, 7, 1, 1, –1, 0, 0, 0

Area	Tins	Height	Width	Doors	Windows

[4]

4 Four statement types and four examples are shown below.

Draw a line to connect each statement type to the correct example.

Statement type	Example
Assignment	`FOR X ← 1 TO 10`
Iteration	`READ X`
Input	`PRINT X`
Output	`X ← Y + Z`

[3]

5 A programmer writes a program to store a patient's temperature every hour for a day.

State the data structure that would be most suitable to use and give the reason for your choice.

Data structure .....

Reason.....

.....[2]

6 Identify **two** different selection statements that you can use when writing pseudocode.

1 .....

.....

2 .....

.....[2]

Question 7 begins on page 10.

7 A database, SOFASELECT, was set up to show the prices of suites, sofas and chairs for sale fro... an online furniture warehouse. Part of the database is shown below.

Description	Brochure Number	Number of Seats	Number of Pieces	Material	Colour	Price in $
Sofa	SF17	2	1	Leather	Red	950
Sofa	SF19	3	1	Vinyl	Black	1,000
Suite	SU10	4	3	Velvet	Green	1,500
Suite	SU23	5	3	Leather	Brown	950
Recliner chair	RC01	1	1	Leather	Cream	600
Chair	CH16	1	1	Vinyl	Red	250
Recliner sofa	RS23	4	1	Leather	Cream	1,200
Chair	CH10	1	1	Velvet	Red	175

(a) How many fields are in each record?

.....[1]

(b) State which field you would choose for the primary key.

.....

Give a reason for choosing this field.

.....

.....[2]

(c) State the data type you would choose for each of the following fields.

Number of Seats .....

Price in $ .....[2]

(d) The query-by-example grid below selects all the furniture in cream leather.

Field:	Description	Material	Colour	Price in $	Brochure Number
Table:	SOFASELECT	SOFASELECT	SOFASELECT	SOFASELECT	SOFASELECT
Sort:				Descending	
Show:	☑	☐	☐	☑	☑
Criteria:		= 'Leather'	= 'Cream'		
or:					

Show the output from the query-by-example.

.....

.....[3]

(e) Complete the query-by-example grid below to select and show the brochure number, material, colour and price of all the furniture with 3 or more seats.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

BLANK PAGE