Cambridge
O Level

**Cambridge International Examinations**
Cambridge Ordinary Level

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

*2765557817*

**COMPUTER SCIENCE** **2210/21**

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 (LK/SW) 124306/1
© UCLES 2016

## 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 parcel delivery service needs a program to check the size and weight of parcels to determine which parcels can be accepted for delivery.

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 size and weight of a single parcel

Each parcel must obey the following rules to be accepted for delivery:
- each dimension must be no more than 80 cm
- the sum of the three dimensions must be no more than 200 cm
- the weight of the parcel must be between one and ten kilograms inclusive

Input and store the weight and dimensions for one parcel. All the dimensions and the weight must be validated on entry and an unsuitable parcel rejected.

Output if the parcel is accepted or rejected. If rejected, output **all** the reasons why the parcel was rejected.

TASK 2 – Check a customer's consignment of parcels

Input and store the number of parcels in the consignment. Calculate the number of parcels accepted and the total weight of the parcels accepted. For each parcel that was rejected, output **all** the reasons why that parcel was rejected.

Output the number of parcels accepted and the total weight of parcels accepted.
Output the number of parcels rejected.

TASK 3 – Calculate the price for a customer's consignment of parcels

Extend TASK 2 to also calculate the price for each parcel, using the following rules:
- 1 kg to 5 kg inclusive costs $10
- Each 100 grams over 5 kg, up to 10 kg, costs an extra $0.10

Your output should also include the price for each parcel accepted and the total price of the consignment.

© UCLES 2016

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

**(i)** For each of the variables that you have used to record dimensions of the parcels in **Task 1**, 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 **two** 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 .....[2]

**(b)** Write an algorithm to complete **Task 1**, using **either** pseudocode, programming statements **or** a flowchart.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[5]

**(c)** Give **three** different data sets that could be used to check your validation rules for **Task 1**. Explain why you chose each data set.

Data set 1 .....

Reason for choice.....

.....

Data set 2 .....

Reason for choice.....

.....

Data set 3 .....

Reason for choice.....

.....[6]

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

## Section B

**2** Read this section of program code that inputs 10 positive numbers and then outputs the smallest number input.

```
1   Small = 1000
2   Counter = 0
3   REPEAT
4     INPUT Num
5     IF Num < Small THEN Small = Num
6     Counter = Counter + 1
7   UNTIL Counter = 10
8   PRINT Small
```

**(i)** Identify **three** changes you would need to make to find the largest number input instead of the smallest number.

1 .....

.....

2 .....

.....

3 .....

.....[3]

**(ii)** Rewrite the program code with your changes.

.....

.....

.....

.....

.....

.....

.....

.....[3]

**3** A program will be written to store information about members of a swimming club.

The following membership details will be recorded:

- Name
- Gender
- Status:
  - Senior
  - Junior
- Fee
- Team member (Yes or No)

**(i)** Choose a suitable data type for each of the membership details to be recorded.

| Membership details | Data type |
|---|---|
| Name | |
| Gender | |
| Status | |
| Fee | |
| Team member | |

[5]

**(ii)** The swimming club has 50 members.

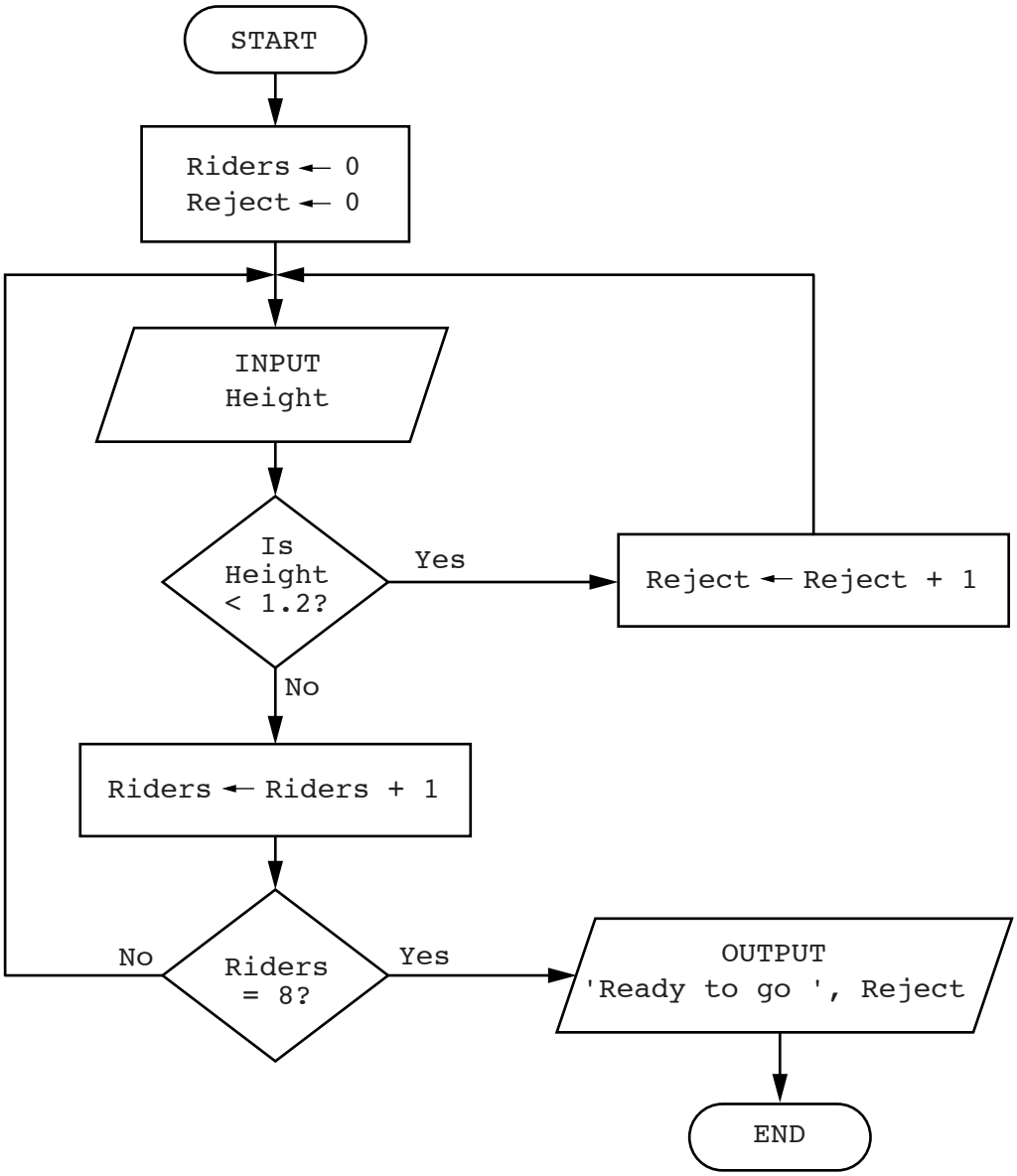
State the data structure that would be most suitable to use and give a reason for your choice.

Data structure.....

Reason.....

.....[2]

**4** The flowchart below inputs the height of children who want to ride on a rollercoaster. Children under 1.2 metres are rejected. The ride starts when eight children have been accepted.



Complete the trace table for the input data:

1.4, 1.3, 1.1, 1.3, 1.0, 1.5, 1.2, 1.3, 1.4, 1.3, 0.9, 1.5, 1.6, 1.0

| Riders | Reject | Height | OUTPUT |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

[4]

**5** `REPEAT ... UNTIL` is one type of loop structure.

Identify and describe **two** other types of loop structure that you could use when writing pseudocode.

Loop structure 1.....

Description.....

.....

Loop structure 2.....

Description.....

.....[4]

**6** A database, STAFFPHONE, was set up to show the telephone extension numbers for members of staff working in a department store.

| Name | Department | Extension number |
|---|---|---|
| Jane Smith | Toys | 129 |
| Sue Wong | Books | 124 |
| David Chow | Toys | 129 |
| Amy Tang | Household | 123 |
| Joe Higgs | Books | 124 |
| Jane Smith | Shoes | 125 |
| Adel Abur | Shoes | 125 |
| Peter Patel | Toys | 129 |

**(a)** Explain why none of the fields in the database can be used as a primary key.

.....

.....

.....[2]

**(b)** State a field that could be added as a primary key.

.....

Give a reason for choosing this field.

.....

.....[2]

**(c)** Use the query-by-example grid below to provide a list of all members of staff, in alphabetical order, grouped by department.

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

[5]

© UCLES 2016

**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.

© UCLES 2016