

CANDIDATE
NAME

CENTRE
NUMBER

| | | | | |
|--|--|--|--|--|
| | | | | |
|--|--|--|--|--|

CANDIDATE
NUMBER

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

COMPUTING

9691/21

Paper 2

May/June 2016

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 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.

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 Zara wants to write a program to print a set of name labels for a group of students.

Each label will show the name surrounded by a border. There is one gap line above and below the name line. This gap line only has the border symbol at the start and end. There are two spaces between the border symbol and the name.

Here is a label example:

```
@@@@@@@@@@@@
@           @
@  FRED   @
@           @
@@@@@@@@@@@@
```

The user will choose the border symbol for the set of labels.

Zara writes pseudocode for her program:

```
INPUT Symbol
REPEAT
  INPUT Name
  LabelWidth ← LENGTH(Name) + 6
  CALL PrintTop2Lines(Symbol, LabelWidth)
  CALL PrintNameLine(Symbol, Name)
  CALL PrintBottom2Lines(Symbol, LabelWidth)
UNTIL Name = "NoName"
```

(a) Write **program code** for the procedure

PrintNameLine(Symbol, Name)

Programming language
Code
.....
.....
.....
.....
.....
.....
.....
..... [4]

Zara uses top-down design and refines her solution:

```
PROCEDURE PrintTop2Lines(Symbol : CHAR, LabelWidth : INTEGER)
  CALL PrintGapLine(Symbol, LabelWidth)
  CALL PrintSymbolLine(Symbol, LabelWidth)
ENDPROCEDURE
```

```
PROCEDURE PrintBottom2Lines(Symbol : CHAR, LabelWidth : INTEGER)
  CALL PrintSymbolLine(Symbol, LabelWidth)
  CALL PrintGapLine(Symbol, LabelWidth)
ENDPROCEDURE
```

(b) Write **program code** for the procedures below.

Programming language

```
PrintSymbolLine(Symbol : CHAR, LabelWidth : INTEGER)
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

```
PrintGapLine(Symbol : CHAR, LabelWidth : INTEGER)
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[6]

(ii) Describe what happens when a name consists of 18 or more characters.

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

(iii) Suggest a solution to the problem that you described in **part (c)(ii)**.

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

(d) Zara used top-down design to develop her modular solution.

Give **two** benefits of a modular solution.

1
.....
2
..... [2]

(e) The pseudocode Zara wrote in **part (c)** has some features that make it easier to read and understand.

State **three** such features.

1
.....
2
.....
3
..... [3]

- 2 A city has a metro line with five stations. The transport department wants to introduce a ticket price calculator for its website. The transport department gives the following specification to a programmer.

Tickets cost 25 cents for each kilometre (km) travelled.

The user chooses the origin and destination of their journey from a list:

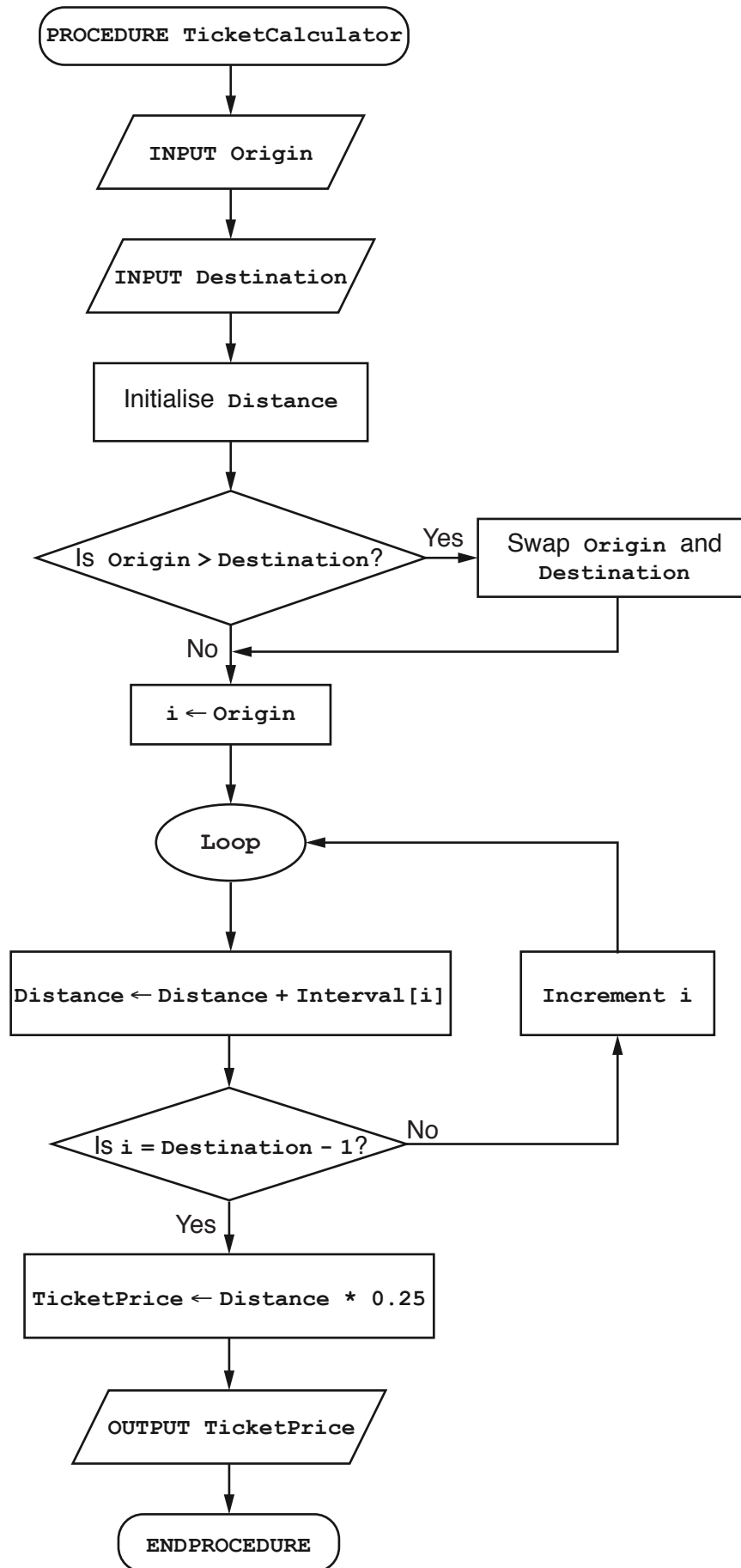
| Station number | Station name |
|----------------|-------------------|
| 1 | Airport |
| 2 | Industrial Centre |
| 3 | City Centre |
| 4 | University |
| 5 | Sports Stadium |

Journeys can start and end at any of the following stations. Travel can be in either direction.

The distances between adjacent stations are:

| Between stations | Distance (km) |
|------------------|---------------|
| 1 and 2 | 4 |
| 2 and 3 | 2 |
| 3 and 4 | 1 |
| 4 and 5 | 3 |

To use the ticket price calculator, a user clicks on the 'Ticket Price Calculator' button. This calls the procedure `TicketCalculator`. The user inputs the numbers of their chosen start and end stations. The procedure calculates and displays the ticket price.



- 3 Alessio has been asked to write a computerised table booking system for a restaurant. The restaurant has 12 tables, each seating between 2 and 8 people.

The restaurant manager wants the following facilities:

- take a new booking
- cancel a booking
- report on which table(s) are available

As a first attempt, Alessio designs his program to take bookings for one evening only. Each table is only booked once per evening.

He writes the pseudocode for his main program:

```

CALL Initialisation
EndProgram ← FALSE
REPEAT
    CALL DisplayOptions
    INPUT Option
    CASE OF Option
        1: CALL TakeBooking
        2: CALL CancelBooking
        3: CALL AvailableTablesReport
        4: EndProgram ← TRUE
        OTHERWISE OUTPUT "Error - invalid input"
    ENDCASE
UNTIL EndProgram = TRUE

```

Each booking is for a number of customers (size of the group). Each table has a number of seats. The procedure Initialisation will set up arrays TableSeats, Booked and GroupSize as follows:

| TableNumber | TableSeats | Booked | GroupSize |
|-------------|------------|--------|-----------|
| 1 | 2 | FALSE | 0 |
| 2 | 2 | FALSE | 0 |
| 3 | 2 | FALSE | 0 |
| 4 | 2 | FALSE | 0 |
| 5 | 4 | FALSE | 0 |
| 6 | 4 | FALSE | 0 |
| 7 | 4 | FALSE | 0 |
| 8 | 4 | FALSE | 0 |
| 9 | 6 | FALSE | 0 |
| 10 | 6 | FALSE | 0 |
| 11 | 8 | FALSE | 0 |
| 12 | 8 | FALSE | 0 |

(a) When a booking enquiry is made, the number of customers is given.

Complete the pseudocode for the procedure `TakeBooking`:

```

PROCEDURE TakeBooking

    DECLARE ..... : INTEGER

    DECLARE ..... : BOOLEAN

    INPUT NumberOfCustomers

    // initialise search for a suitable table

    Found ← FALSE

    TableNumber = .....

    REPEAT          // find a table with enough seats

        TableNumber ← TableNumber + 1

        IF TableSeats[.....] .....

            AND Booked[.....] .....

        THEN

            Found ← TRUE

        ENDIF

    UNTIL ..... OR .....

    IF Found = FALSE

        THEN // no tables left with enough seats

            .....

        ELSE // make the booking

            Booked[.....] .....

            GroupSize[.....] .....

            OUTPUT "Table number booked: ", TableNumber

        ENDIF

    ENDPROCEDURE

```

[11]

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.