OCR
RECOGNISING ACHIEVEMENT

ADVANCED SUBSIDIARY GCE

COMPUTING

Programming Techniques and Logical Methods

F452

Candidates answer on the question paper.

OCR supplied materials:
None

Other materials required:
None

Friday 21 January 2011
Morning

Duration: 1 hour 30 minutes

Candidate forename		Candidate surname	

Centre number						Candidate number					

INSTRUCTIONS TO CANDIDATES

* Write your name, centre number and candidate number in the boxes above. Please write clearly and in capital letters.
* Use black ink. Pencil may be used for graphs and diagrams only.
* Read each question carefully. Make sure you know what you have to do before starting your answer.
* Write your answer to each question in the space provided. Additional paper may be used if necessary but you must clearly show your candidate number, centre number and question number(s).
* Answer **all** the questions.
* Do **not** write in the bar codes.

INFORMATION FOR CANDIDATES

* The number of marks is given in brackets **[ ]** at the end of each question or part question.
* The total number of marks for this paper is **100**.
* This document consists of **20** pages. Any blank pages are indicated.

OCR is an exempt Charity

Turn over

1 The Anytown Bus Company uses a computer program to calculate the bus fares of journeys and to print bus tickets.

The program uses an array called BusStop to store the names of all the stops in the order of travel. The beginning of the array for a particular route is shown below.

High St
Station Rd
Green La
Avenue
New St
Kingsway

(a) The value of BusStop(1) is High St.

State the value of BusStop(4).

.... [1]

(b) When the program is used, the driver enters the name of a stop. The program needs a function which will return the position of that stop in the array BusStop.

(i) State the position which should be returned if the name entered is Kingsway.

.... [1]

(ii) Describe how a serial search can be used to determine the position when the name of a stop is entered.

....

....

....

....

....

....

....

....

....

.... [5]

(c) The program uses variables to store the following data.

For each variable, state the most suitable data type and give a reason for your choice.

  (i) RouteNumber (the number of the bus route. e.g. 65, 5A or N93)

Data Type ....

Reason ....

.... **[2]**

  (ii) Fare (the cost of the journey in pounds e.g. 0.40)

Data Type ....

Reason ....

.... **[2]**

  (iii) TicketType (the word CHILD, ADULT or PENSIONER)

Data Type ....

Reason ....

.... **[2]**

  (iv) Destination (the position of the destination of the journey in the array BusStop)

Data Type ....

Reason ....

.... **[2]**

4

(d) To calculate the fare, the program uses the following function.

```
01  FUNCTION CalculateFare(Start, Destination, Type)
02      CONSTANT PensionerMax = 0.50
03      Distance = Destination – Start
04      Fare = Distance * 0.20
05      IF  Type = "CHILD" THEN
06          Fare = Fare / 2
07      END IF
08      IF Type = "PENSIONER" AND Fare > PensionerMax
09          Fare = PensionerMax
10      END IF
11      RETURN Fare
12  END FUNCTION
```

Use the function to calculate the fare for the following journeys.

You must show your working.

(i) An adult going from High St to Avenue.
(Start = 1, Destination = 4, Type = ADULT)

....

....

....

....

....

.... **[3]**

(ii) A child going from New St to Kingsway.
(Start = 5, Destination = 6, Type = CHILD)

....

....

....

....

....

....

....

.... **[4]**

(iii) A pensioner going from Station Rd to New St.
(Start = 2, Destination = 5, Type = PENSIONER)

....

....

....

....

....

....

....

....**[4]**

(e) The function declares and uses the constant PensionerMax.

(i) State **two** advantages of declaring and using this constant.

....

....

....

....**[2]**

(ii) Identify **one** other value in the code for which a constant could be used, and state a suitable name for this constant.

Value ....

Suitable name ....**[2]**

6

(f) All tickets have three lines of 15 characters. Each line is formatted into a single string and then printed.

An example ticket is shown below.

R	O	U	T	E		N	9	3						
A	V	E	N	U	E		–	K	I	N	G	S	W	A
A	D	U	L	T					£	0	.	4	0	

- **(i)** The first line contains the word ROUTE followed by a space followed by the value of RouteNumber.

  Using string manipulation operations in a high level language you have studied, show how the line can be formatted into a single string.

  ....

  ....

  ....

  .... **[2]**

- **(ii)** In the second line the names of the stops are printed using 7 characters each. If a name is shorter than 7 characters, spaces are added. If a name is longer than 7 characters, any extra characters are removed.

  Show how string manipulation functions can be used to format **the name of a stop** for printing.

  You do not need to format the whole line.

  ....

  ....

  ....

  ....

  ....

  .... **[3]**

© OCR 2011

(iii) In the third line the ticket type is printed on the left and the fare (preceded by £) is printed on the right.

Explain how string manipulation functions can be used to format the values of TicketType and Fare into a correct 15-character string for printing.

....

....

....

....

....

....

....

....

....

.... **[5]**
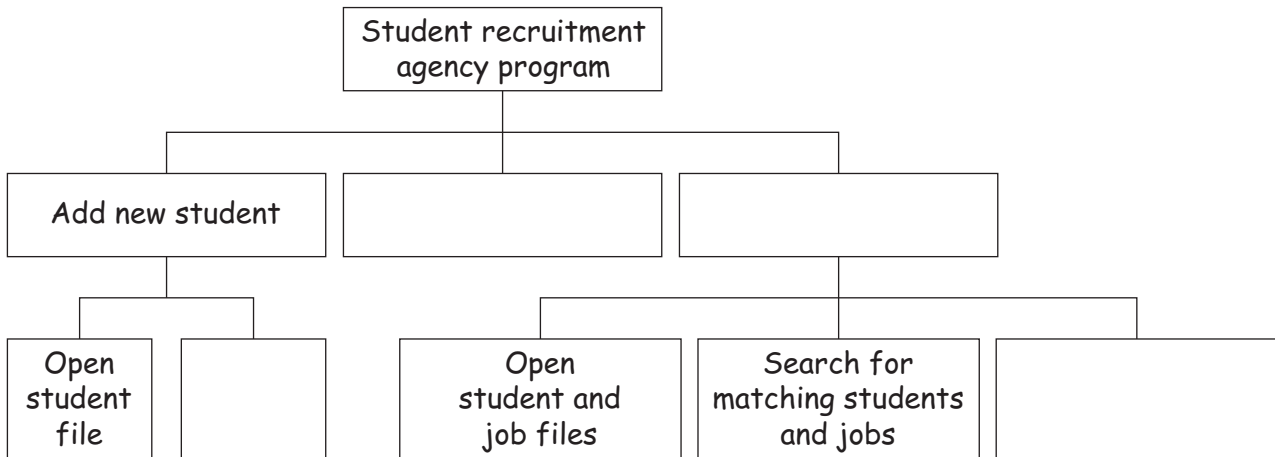
2 Sanchez is a student doing his A-Level Computing project. He is working with a recruiting agency which specialises in local part-time jobs for students.

(a) The agency wants a program which will allow the user to

- add details of a student to a file,
- add details of a job to another file,
- match students with jobs.

Sanchez decides to produce a top-down design for the program. Part of this design is shown in the structure diagram below.

```
                        ┌─────────────────────┐
                        │ Student recruitment │
                        │   agency program    │
                        └─────────────────────┘
           ┌────────────────────┼────────────────────┐
  ┌─────────────────┐  ┌─────────────────┐  ┌─────────────────┐
  │ Add new student │  │                 │  │                 │
  └─────────────────┘  └─────────────────┘  └─────────────────┘
      ┌──────┴──────┐          ┌───────────────┼───────────────┐
 ┌─────────┐ ┌─────────┐  ┌──────────┐ ┌──────────────┐ ┌──────────┐
 │  Open   │ │         │  │   Open   │ │  Search for  │ │          │
 │ student │ │         │  │ student  │ │   matching   │ │          │
 │  file   │ │         │  │ and job  │ │  students    │ │          │
 │         │ │         │  │  files   │ │  and jobs    │ │          │
 └─────────┘ └─────────┘  └──────────┘ └──────────────┘ └──────────┘
```

Write the letters A, B, C and D in the blank boxes to show the most appropriate location for the following modules.
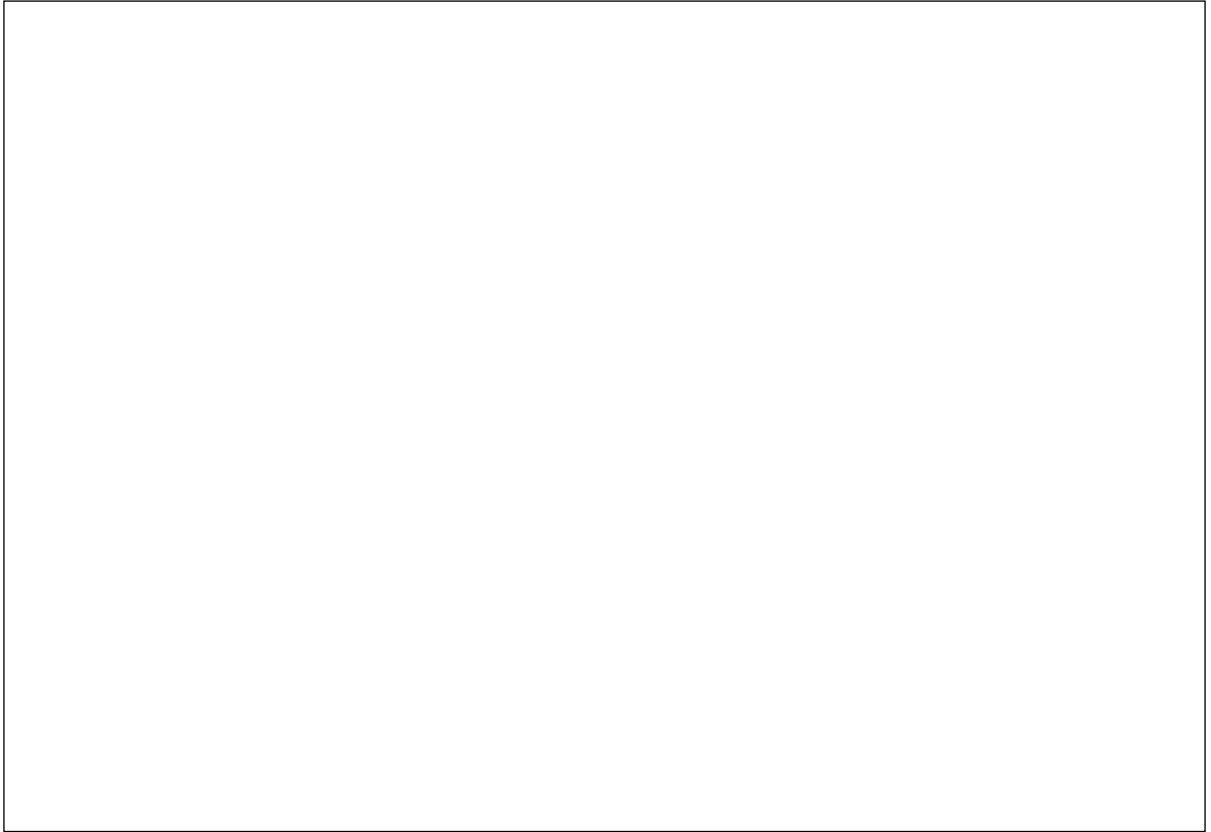
A = Add new job
B = Append student record
C = Match students to jobs
D = Print report of matches

[4]

(b) When a student registers with the agency for part-time work, they must fill in a form giving the following information.

- Name
- Date of birth
- Mobile telephone number
- Whether they have a full driving licence

In the space below, design the layout of a data capture form the student would complete.

[8]

(c) Using the student recruitment agency program as well as other examples, discuss the importance of a good design of the interface used to input data into a computer program.

(The quality of written communication will be assessed in your answer to this question.)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[8]**

11

BLANK PAGE

PLEASE TURN OVER FOR THE NEXT QUESTION

PLEASE DO NOT WRITE ON THIS PAGE

3 The design for a computer program contains the following algorithm shown in pseudo-code.

```
01  INPUT A
02  INPUT B
03  C = 0
04  IF A = B THEN
05      B = 1
06  ELSE
07      WHILE B > A
08          B = B - A
09          C = C + 1
10      END WHILE
11      A = B
12  END IF
```

(a) Define the following terms. For each, give an example from the algorithm.

(i) Statement

....

....

....

.... **[2]**

(ii) Selection

....

....

....

.... **[2]**

(b) Explain how nesting has been used in the algorithm above. You should refer to line numbers in your answer.

....

....

....

....

....

.... **[3]**

(c) Good program writing techniques make pseudo-code easier to follow.

   (i) Explain **one** technique used in the pseudo-code on the previous page to make it easier to follow.

   ....

   ....

   ....

   .... **[2]**

   (ii) Explain **one** technique **not used** in the pseudo-code on the previous page, which could make it easier to follow.

   ....

   ....

   ....

   .... **[2]**

(d) Explain the difference between the use of A=B on line 4 and line 11, by referring to the type of operation.

   ....

   ....

   ....

   ....

   ....

   .... **[4]**

(e) Explain why line 03 is needed in this algorithm.

   ....

   ....

   ....

   .... **[2]**

(f) Explain the difference between white box testing and a black box testing.

....

....

....

....

....

....

....

.... **[4]**

15

BLANK PAGE

PLEASE TURN OVER FOR THE NEXT QUESTION

PLEASE DO NOT WRITE ON THIS PAGE

Here is a copy of the algorithm shown at the beginning of this question.

```
01  INPUT A
02  INPUT B
03  C = 0
04  IF A = B THEN
05      B = 1
06  ELSE
07      WHILE B > A
08          B = B - A
09          C = C + 1
10      END WHILE
11      A = B
12  END IF
```

The algorithm is tested with the inputs 2, 5.

The path of execution is:

    01, 02, 03, 04(FALSE), 06, 07(TRUE), 08, 09, 10,
    07(TRUE), 08, 09, 10, 07(FALSE), 11, 12

The final values of the variables A, B and C are:

    A =1   B = 1   C = 2

(g) For the following sets of inputs state the path of execution. If line 04 or line 07 is executed, you should state whether the condition is TRUE or FALSE. Also state the final values of A, B and C.

(i) 10, 10

Path of execution:

....

....

....

Final values:

A = ...   B = ...   C = ...

[4]

(ii) 4, 6

Path of execution:

....

....

....

....

Final values:

A = ...   B = ...   C = ...

[5]

(h) Describe the set of input data which will produce the following path of execution using an example.

01, 02, 03, 04(FALSE), 06, 07(FALSE), 11, 12

Description: ....

....

Example: .... **[2]**

4 The gears of a bicycle contain up to 8 rings with decreasing numbers of teeth.
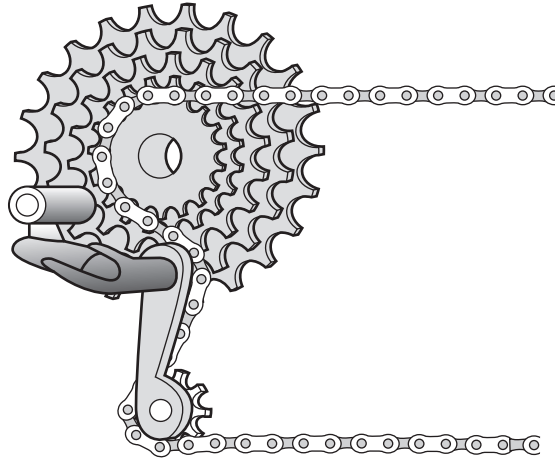
A computer program in a bicycle repair shop allows the user to input the number of teeth on each ring into an array called Ring using the method described below.

- The user inputs the number of rings between 1 and 8.
- The user then inputs the number of teeth on each ring, starting with the largest.
- The program checks that each number of teeth input is smaller than the previous number.
- The program stores the number of teeth on the first ring into the array Ring as Ring(1), the number of teeth on the second ring as Ring(2) and so on.
- If there are fewer than 8 rings, any unused elements of the array Ring are set to 0.

Write an algorithm for the routine to input the number of teeth on each ring as described above.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[8]**

PLEASE DO NOT WRITE ON THIS PAGE