

795 Computer Science 3

CAMEROON GENERAL CERTIFICATE OF EDUCATION BOARD General Certificate of Education Examination

JUNE 2013

ADVANCED LEVEL

Subject Title	Computer Science
Paper No.	Paper 3 – Practical
Subject Code No.	795

Three Hours

Carry out ALL the tasks given. For your guidance, the approximate mark for each part of a task is indicated in brackets.

Great importance is attached to the accuracy, layout and labelling of the drawings and computer generated outputs.

You are reminded of the necessity for good English and orderly presentation of your answers.

Write algorithms in the answer booklet provided. Also record in your answer booklet any information requested or that you believe would make it easier to understand how you carried out tasks or answered questions.

When an imperative programming language is required to write program code, either **Standard [ISO]** Pascal or the **[ANSI]** C programming language may be used.

If need be, supervisors will assist you in recording details of intermediate work carried out on the computer.

Turn Over

Exercise (Programming; Applied to Concurrent Systems)

Instructions for Exercise: Analyse the problem in the answer booklets provided. Also record in your answer booklets any other information requested, or that you believe would make it easier to understand the exercises you have carried out. Instructors will assist you in recording intermediate samples of work carried out on the computer.

You are expected to print out a single copy of relevant fragments of your program at different times. **PLEASE NOTIFY THE SUPERVISOR OF ANY REQUIRED PRINTOUT THAT WAS NOT DONE!**

Carry out the instructions/tasks below in order to write a program that coordinates three processes in a system.

Consider a very simplified booking system (BS) at a cashier of a supermarket. The system consists of three processes: the Reader (R), the Controller (C) and the Printer (P). R reads the code for an article and communicates it to the Controller. An article is a sales item such as a shirt. On receiving such data, the Controller transmits the price of the sales item to the Printer which then prints the item's name together with its price on the receipt.

Figure 1 shows the individual processes. Each process can only be in one of two states labelled 0 and 1. Each process moves from one state to another along an arrow (arc) by carrying out the operation adjacent to the arc. Thus, the Reader moves from state 0 to state 1 by executing the *store* operation, and back to state 1 by executing the *scan* operation. The operations are defined as follows:

- (a) *scan*: Read a value from the standard input for eventual storage.
- (b) *store*: Store data for use in eventual printing.
- (c) *p_cmd*: Issue an instruction for stored data to be printed.
- (d) *print*: Output data ready for printing.

Processes function independently of each other, but are synchronised by operations that are invoked in two or more processes – i.e., the *store* and *p_cmd* operations.

The state of the booking system at any given time is the combined state of the processes. It is given as the triple (r, c, p) where r , c , and p correspond to the state of process R, C and P respectively. In Figure 2, boxes indicate the state of processes and arrows indicate the operations carried out. The new state obtained is at the head of the arrow.

Thus, in state $(1, 0, 0)$ a *store* operation is performed to transition to state $(0, 1, 0)$. This is because state r (left state) is 1 and so the *store* in process Reader changes it to 0 as shown in Figure 1(a). Similarly, state c is changed from 0 to 1 by *store* in Controller as shown in Figure 1(b). State p does not change because there is no *store* operation in Printer, as can be seen in Figure 1(c).

States Y1, Y2, Y3 and operations X1, X2 and X3 in Figure 2 have been deliberately concealed for later use.

Task 1 – Problem Understanding

1. Having read the problem description above, copy Figure 2 in your answer booklet.
 - a. Fill in the missing triples labelled Y1, Y2 and Y3.
 - b. Then fill in the missing operations labelled X1, X2 and X3.

(3 marks)

(3 marks)

[HINT: For triples, consider the states that are changed by a given operation in Figure 1. For operations, consider in turn what operation may change each of the process states r , c or p to their respective target state.]

3

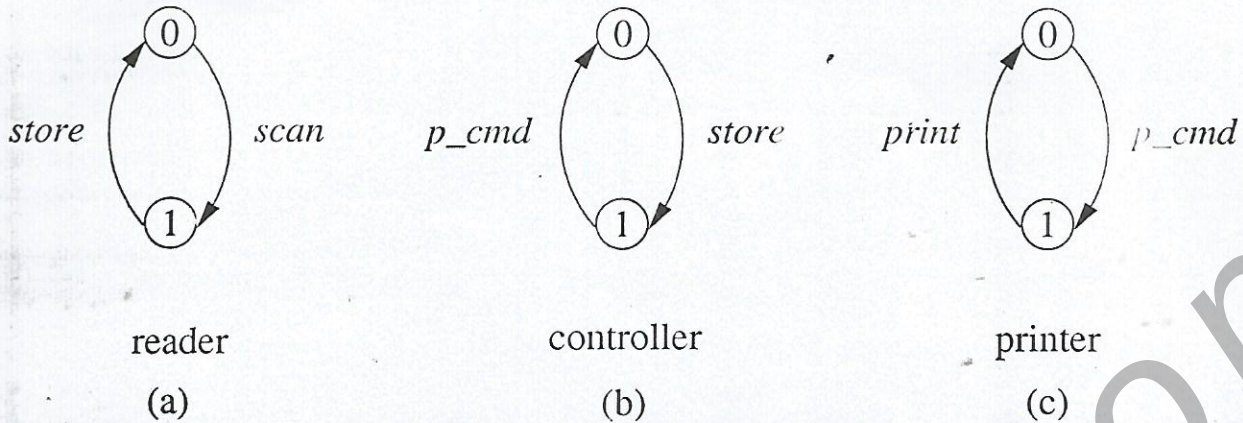


Figure 1 – States in Individual Components of Booking System

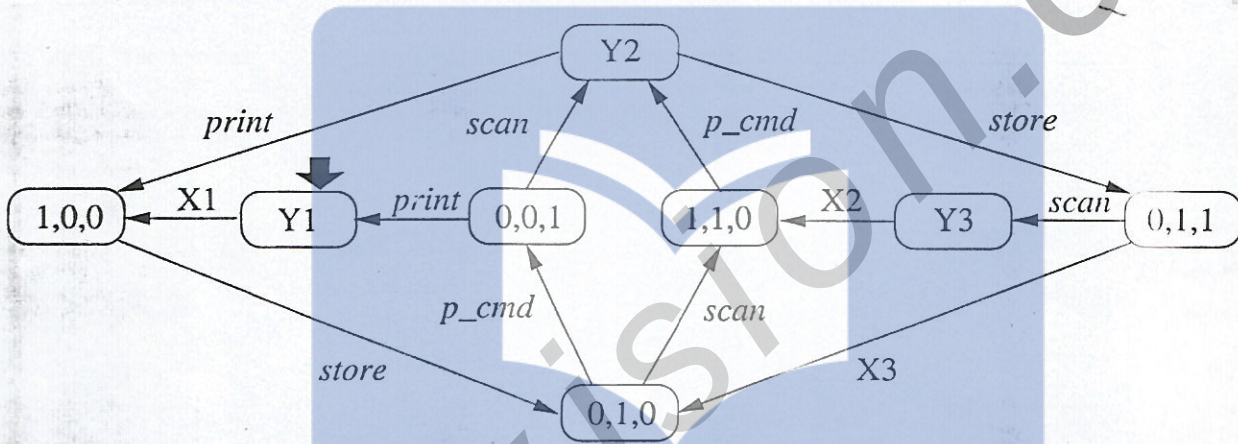


Figure 2 – Representation of State Transitions (changes) of Booking System

Task 2 – Develop Operations of Processes

- Algorithms for operation *store* and *scan* are given in pseudocode in Algorithm 1 below. In your answer booklet, give the algorithms for operations *p_cmd* and *print*. For the current item ID in variable *c_id*, operation *p_cmd* copies its item name and item cost into the string variable *p_name* and the integer variable *p_cost* respectively. Operation *print* does output as explained in the text.

[HINT: The algorithm for operation *print* is similar to the algorithm for operation *scan*, except that it considers input. The algorithm for operation *p_cmd* is similar to the algorithm for operation *store* in that they consider the states of two processes.]

(6 marks)

3. Implementing Operations of processes:

- Use a text editor or your favourite program development environment to type in your program text. Add the code fragments for constant, variable and procedure declarations that have been given you by instructors in the appropriate places in your program code. (1 mark)
- From the algorithms in Algorithm 1 and your answer in (a), write programming Language (PL) procedures (and/or functions) for the FOUR operations *store*, *scan*, *p_cmd* and *print*. Ensure your procedures can at least compile successfully. (10 marks)
- Save a copy of your answer from (b) in the location indicated by instructors, and then print your PL procedures. (1 mark)

Turn Over

Variables

```
(* Some variables used in code fragments given to candidates *)
in_id ≡ input variable of scan for item ID
c_id ≡ Controller variable for item ID
p_cost ≡ cost of sales item
p_name ≡ name of sales item
tbl1 ≡ table of sales item costs
tbl2 ≡ table of sales item names
r_s ≡ integer variable for state r in triple (r,c,p)
c_s ≡ integer variable for state c in triple (r,c,p)
p_s ≡ integer variable for state p in triple (r,c,p)
```

Algorithms:

```
(* Read sales item ID from Standard input *)
scan()≡
  IF Reader in state 0 THEN
    Read product ID into variable in_id
    Change state of Reader process to state 1
  END
END (* scan *)

(* store sales item ID for Controller use *)
store()≡
  IF Reader in state 1 AND Controller in state 0 THEN
    Store item ID in variable c_id.
    Change state of Reader process to state 0.
    Change state of Controller process to state 1.
  END
END (* store *)
```

Algorithm 1 - Algorithms for Operations *scan* and *store* in Pseudocode**Algorithm:**

```
(* Variables xch and choice are globally defined. *)
(* Randomise the variable choice *)

xch ← (xch + 2*choice) mod 91
choice ← xch mod 3

(* perform one of actions 1, 2 or 3 if variable choice *)
(* is 0, 1 or 2 respectively *)
CASE choice IS
  0: perform action 1
  1: perform action 2
  2: perform action 3
DEFAULT: Issue a suitable error message
END
```

Algorithm 2 - Algorithm to Randomise procedure calls

Task 3 – Develop Algorithms of Processes

4. In your answer booklet, give three separate algorithms called Reader, *Controller* and Printer respectively that execute the various operations invoked by the processes *Reader*, *Controller* and *Printer* respectively. Give the algorithms in pseudocode. (6 marks)
5. Implement Process algorithms by doing the following in your program from Task 2: (6 marks)
- a. Implement the algorithms as separate PL procedures (or functions). (3 marks)
 - b. Implement a PL procedure called `execOpn` that calls your procedures Printer, Reader, Controller in that order. Call `execOpn` from your main program. (2 marks)
 - c. After making sure your program works, save your work and then print it. (1 mark)

Task 4 – Implement Top-level Algorithms

6. In your main program do the following:
- a. Invoke the procedure `initialise` to initialise all variables. This procedure sets the initial state to the triple (0,0,0). This state is indicated by the thick arrow in Figure 2. It also initialises a table of sales items and their prices. (1 mark)
 - b. Modify your driver program from Task 3 to repeatedly invoke `execOpn` until a negative number is input (via the scan operation). (2 marks)

Task 5 – Sample Output Results

7. Run your program with the following input data items and print your output. Also store a copy of your output data in the location indicated by instructors. The input data is:

1 2 4 0 3 10 5 8 5 9 6 -1 2 -9

NOTE: Your program need not use all the input data given above.

(2 marks)

8. Algorithm 2 is an algorithm that can be used to perform procedure calls in a random order. Use it to modify procedure `execOpn` from (6) so that it executes PL procedure Printer, Controller and Reader only when variable choice is 0, 1 or 2 respectively. (4 marks)

[NOTE: Variables `choice` and `xch` are globally defined and initialised in the sample code given to you.]

9. Run the program to make sure the operations work correctly. Save your program where indicated by instructors. (1 mark)
10. Run your modified program, with the same input data from (7). Save and print your output. (2 marks)
11. Is your output the same or different? Explain why. (2 marks)