1. Nov/2021/Paper_21/No.1(d)

(d) Write an algorithm for **Task 2**. You do **not** need to display updated court availability. Use either pseudocode, programming statements or a flowchart. Assume that **Task 1** has been completed.

[6]

2. Nov/2021/Paper_21/No.2(b)

(b) The pseudocode for this algorithm could be shortened by the use of a `FOR ... NEXT` loop.

Rewrite the algorithm using a `FOR ... NEXT` loop.

.....

.....

.....

..... [3]

3. Nov/2021/Paper_22/No.1(c),(d)

(c) Write an algorithm to show how your program carries out **Task 3**, using either pseudocode, programming statements or a flowchart. Assume **Tasks 1** and **2** have already been completed.

[6]

(d) Explain how your program could be changed to count and store the number of bookings made by each passenger. Then, after ten bookings have been made by a passenger, apply an additional 10% discount to every future booking. Any programming statements used in your answer must be fully explained.

.... [4]

4. Nov/2021/Paper_22/No.4(b)

(b) The algorithm needs changing to allow only the numbers 1, 2, 3, or 4 to be entered for the input variable `Operator`.

Write the pseudocode to perform this task and state where in the algorithm it would be located.

Pseudocode .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Location in algorithm .....

.....

.....

[5]

5. Nov/2021/Paper_23/No.1(d)

(d) Part of **Task 2** is to check if a table is available. Write an algorithm to do this, using either pseudocode, programming statements or a flowchart. Assume that **Task 1** has been completed.

[6]

6. Nov/2021/Paper_23/No.2(b)

(b) The pseudocode for this algorithm could be shortened by the use of a `FOR ... NEXT` loop.

Rewrite the algorithm using a `FOR ... NEXT` loop.

.....

.....

.....

..... [3]

7. March/2021/Paper_22/No.1(c)

(c) Write an algorithm for **Task 1**, using **either** pseudocode, programming statements **or** a flowchart.

[6]

8. June/2021/Paper_21/No.1(c)

(c) Write an algorithm using pseudocode, programming statements **or** a flowchart to show how your program completes these parts of **Task 2**:

- Allows students to enter their unique voter number before casting their vote.
- Checks whether the student has already voted:
  - if so, supplies a suitable message and does **not** allow them to vote.
  - if not, stores the unique voter number, but **not** their vote, in a suitable data structure.

It is **not** necessary to show parts completed in **Task 1**, including counting of votes for each candidate.

[5]

9. June/2021/Paper_21/No.4(b)

(b) Write the pseudocode to output the contents of the arrays `Score[]` and `Grade[]` along with suitable messages.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

10. June/2021/Paper_22/No.1(c)

(c) Write an algorithm for the part of **Task 2** that inputs the tickets required, calculates the total price for the ticket(s) purchased by a passenger including a group discount (if applicable) and updates the data structures used for the totals.
Use **either** pseudocode, programming statements **or** a flowchart.

[6]

11. June/2021/Paper_22/No.2

(a) Write an algorithm in pseudocode to input 500 positive whole numbers. Each number input must be less than 1000. Find and output the largest number input, the smallest number input and the range (difference between the largest number and smallest number).

.....

.....

.....

..... [6]

(b) Describe how the algorithm could be changed to make testing less time-consuming.

.....

.....

.....

..... [2]

12. June/2021/Paper_22/No.5

A one-dimensional array `dataArray[1:20]` needs each element set to zero.

(a) Write a pseudocode routine that sets each element to zero. Use the most suitable loop structure.

.....

.....

.....

.....

.....

.....

.....

..... [3]

(b) Explain why you chose this loop structure.

.....

..... [1]

13. June/2021/Paper_23/No.1(b)

(b) Write an algorithm for **Task 2**, using pseudocode, programming statements **or** a flowchart. It is not necessary to show details completed in **Task 1**.

[6]