

|                                                           |                              |                           |  |
|-----------------------------------------------------------|------------------------------|---------------------------|--|
| <b>REGISTRATION CENTRE NUMBER</b>                         |                              | <b>CENTRE NAME</b>        |  |
| <b>CANDIDATE'S FULL NAMES</b>                             |                              |                           |  |
| <b>CANDIDATE IDENTIFICATION<br/>NUMBER</b>                | <b>SUBJECT CODE<br/>0795</b> | <b>PAPER NUMBER<br/>3</b> |  |
|                                                           |                              |                           |  |
| <b>FOR OFFICIAL USE ONLY<br/>(Candidate Random CODE):</b> |                              |                           |  |
| <b>HOLY INFANT HIGH SCHOOL – MELEN</b>                    |                              |                           |  |
| <b>ADVANCED LEVEL MOCK EXAMINATION</b>                    |                              |                           |  |
| <b>SUBJECT TITLE<br/>COMPUTER SCIENCE</b>                 | <b>SUBJECT CODE<br/>0795</b> | <b>PAPER NUMBER<br/>3</b> |  |
|                                                           | <b>EXAMINATION DATE:</b>     | <b>MARCH 2026</b>         |  |
| <b>FOR EXAMINERS' USE ONLY</b>                            |                              |                           |  |
| Marked By: .....                                          |                              | <b><u>SCORE</u></b>       |  |
| Signature: ..... Date: .....                              |                              |                           |  |
| Checked By: .....                                         |                              |                           |  |
| Signature: ..... Date: .....                              |                              |                           |  |

**SET 2  
TWO HOURS**

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

Great importance is attached to the accuracy, layout and labeling of diagrams and computer-generated outputs.

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

You are expected to print a single copy of relevant fragments of your work at different times. Please notify the supervisor of any required printout that was not done!

When an imperative programming language is required to write program code, either standard (ISO) Pascal or the Standard (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.

**PROGRAMMING: A COMPUTER PROGRAM FOR YOUNGER STUDENTS****(35 Marks)**

A team of four high-school students decided to create a computer program for younger students to help them learn to count and recognize whole numbers from 1 to 25.

A table partially completed with numbers from 1 to 25 is given to a younger student. The table consists of five rows and five columns (see **Figure 1**). The younger student must enter the remaining numbers. Each number from 1 to 25 should be entered only once.

|    |    |    |    |    |
|----|----|----|----|----|
|    | 24 | 1  |    |    |
|    |    | 7  | 14 |    |
| 4  | 6  |    |    | 22 |
| 10 |    |    | 21 | 3  |
| 11 |    | 25 |    | 9  |

**Figure 1: Example table partially filled with numbers**

Each of the four students is responsible for creating several sub-programs that will be included in the program. The sub-program *inputInRange()* should return a valid input number. This will ensure that the younger student's input is a number in the range 1–25 inclusive. If the input number is not in this range, an appropriate message is output. The process will be repeated until a valid number is input. For example:  
For the sub-program call  $N = \text{inputInRange}()$ , if the following appears on the screen:

Enter a number:

The number is not in the range from 1 to 25. Please try again.

Enter a number:

The number is not in the range from 1 to 25. Please try again.

Enter a number:

Then the value of N would be 15 after execution.

- (1) Write a programming language (PL) sub-program *inputInRange()*. **(5 marks)**
- (2) Write a programming language (PL) subprogram *InsertValues(MAT)* that inserts values to the array *MAT* after verifying if each number falls in the range (1-25) inclusive by calling on the subprogram *inputInRange()*. **(5 marks)**
- (3) Write a programming language (PL) subprogram *Initialise()* that initialises the one-dimensional array, *FLAGS*, with zero (0) values. **(3 marks)**  
**NB: *FLAGS* and N are global variables.**

In the program, the table is stored as a static two-dimensional array.

|     | [0] | [1] | [2] | [3] | [4] |
|-----|-----|-----|-----|-----|-----|
| [0] | 2   | 24  | 1   | 8   | 15  |
| [1] | 23  | 5   | 7   | 14  | 16  |
| [2] | 4   | 6   | 13  | 20  | 22  |
| [3] | 10  | 12  | 19  | 21  | 3   |
| [4] | 11  | 18  | 25  | 17  | 9   |

**Figure 2a:** The two-dimensional array is correctly completed because each of the numbers from 1 to 25 appears only once

|     | [0] | [1] | [2] | [3] | [4] |
|-----|-----|-----|-----|-----|-----|
| [0] | 11  | 24  | 1   | 8   | 15  |
| [1] | 23  | 5   | 7   | 14  | 16  |
| [2] | 4   | 6   | 13  | 20  | 22  |
| [3] | 10  | 2   | 19  | 21  | 3   |
| [4] | 11  | 18  | 25  | 2   | 9   |

**Figure 2b:** The two-dimensional array is not correctly completed because the number 17 and 12 are missing and the numbers 11 and 2 appear twice.

**Figure 2:** Two examples of data

The sub-program *isCorrectlyCompleted*(MAT) accepts the two-dimensional array MAT and outputs an appropriate message to indicate if it has been correctly filled.

For example:

- If MAT is the two-dimensional array shown in Figure 2a, then *isCorrectlyCompleted*(MAT) should output that the table has been correctly completed.
- If MAT is the two-dimensional array shown in Figure 2b, then *isCorrectlyCompleted*(MAT) should output that the table has not been correctly completed.

The pseudocode of the algorithm for the subprogram *isCorrectlyCompleted*(MAT) is given below:

```

SUBPROGRAM isCorrectlyCompleted(MAT) ≡
    CALL Initialise()
    //Visit all elements of the MAT array
    FOR i FROM 0 TO 5 DO
        FOR j FROM 0 TO 5 DO
            Num = MAT[i][j]
            //Assign one (1) to the FLAGS array at that index
            FLAGS[Num-1] = 1;
        ENDFOR
    ENDFOR
    //Inspect elements of the FLAGS array
    FOR i FROM 0 TO 25 DO
        IF (FLAGS[i] = 0)
            isComp = FALSE //isComp is a global variable
            BREAK
        ENDIF
    ENDFOR
ENDPROGRAM
    
```

For example, the *FLAGS* array for the *MAT* array shown in **Figure 2a** is:

|     |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | [12] | [13] | [14] | [15] | [16] | [17] | [18] | [19] | [20] | [21] | [22] | [23] | [24] |
| 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    |

And the *FLAGS* array for the *MAT* array shown in **Figure 2b** is:

|     |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | [12] | [13] | [14] | [15] | [16] | [17] | [18] | [19] | [20] | [21] | [22] | [23] | [24] |
| 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1    | 0    | 1    | 1    | 1    | 1    | 0    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    |

- (4) Write a programming language (PL) for the *isCorrectlyCompleted (MAT)* sub-program as described. **(7 marks)**
- (5) Write a programming language (PL) subprogram *MissingNumbers ()* which displays the missing values of the array *MAT* by referring to array *FLAGS*. **(3 marks)**  
 For Example:  
 The missing numbers are:  
 17  
 12
- (6) Write a programming language (PL) subprogram *displayGrid (MAT)* which displays the values inserted Grid *MAT*. **(4 marks)**
- (7) Write a main program that does the following: **(6 marks)**
- Prompt the user to insert values of the grid *MAT*.
  - Display the array *MAT*.
  - Verify if array *MAT* is correctly completed or not.
  - If not complete, output a message saying that “the table has not been completed correctly”; otherwise, output a message saying that “the table has not been completed correctly” and displays the missing numbers.
- Save your program as *Task 2*.
- (8) Do a screenshot of your output when you enter the data in **Figure 2a** and **Figure 2b** respectively. Save the file as *screenshot2*. **(2 marks)**

**DATABASE: HOTEL MANAGEMENT SYSTEM**

**(25 Marks)**

Given a bookstore **MrBooks**, you are asked to design a database such that in this bookstore:

- A customer may borrow a book or more.
- For every book, there is just one author.
- An author could have written many books.
- When borrowing the book, the code of the book, the borrower's code, the borrow date, and the return date must be recorded as well as the state of the book.

Answer the following questions given that the entities Author, Book and Customer have attributes as shown below:

- *Author* (Aut-code, Name)
- *Book* (B-code, title, edition)
- *Customer* (Reg-No, Name, address)

- (1) Using the same notation as that for the entities given above, derive the relational schema for all relations you have identified from the description given. Make sure you indicate the primary key(s), foreign keys and all other attributes for each relation in third normal form (3NF). **(4 marks)**

---

---

---

---

- (2) Construct an entity-relationship diagram (ERD) for the bookstore's database that exactly represents the information in (1) above. **(3 marks)**

- (3) Using the Database Management System of your choice, create a database named **MrBooks**. **(1 mark)**

---

- (4) Create all the relations mentioned in (1) above in your database **(3 marks)**

(5) Populate your tables with data gotten from the flat-file below:

(5 marks)

| Reg-No | Name     | Address       | Aut-code | Author     | B-code | Title                 | Edition         | BorrowDate | BookState |
|--------|----------|---------------|----------|------------|--------|-----------------------|-----------------|------------|-----------|
| CUS01  | John P.  | 456 Ave Rd    | AUT01    | Jane A.    | BK01   | Pride and Prejudice   | 3 <sup>rd</sup> | 2026-01-10 | Fair      |
| CUS01  | John P.  | 456 Ave Rd    | AUT01    | Jane A.    | BK02   | Sense and Sensibility | 2 <sup>nd</sup> | 2026-01-12 | Excellent |
| CUS02  | Epie E.  | 322 UB Jun.   | AUT02    | Charles D. | BK03   | Great Expectations    | 1 <sup>st</sup> | 2026-01-15 | Good      |
| CUS02  | Epie E.  | 322 UB Jun.   | AUT01    | Jane A.    | BK01   | Pride and Prejudice   | 3 <sup>rd</sup> | 2026-01-08 | Excellent |
| CUS03  | Marie D. | 796 Bonas.    | AUT02    | Charles D. | BK03   | Great Expectations    | 1 <sup>st</sup> | 2026-01-14 | Good      |
| CUS04  | Ekuh A.  | 232 Cnt Hand. | AUT02    | Charles D. | BK03   | Great Expectations    | 1 <sup>st</sup> | 2026-01-05 | Good      |

(6) It has been noticed that data wasn't provided for the Borrower's return date. Write an SQL statement to add the attribute *ReturnDate* in its appropriate table. (2 marks)

---

(7) Write SQL statement that displays the names of customers that have borrowed '*Pride and Prejudice*'? Show customer name, book title, and Borrow date. (3 marks)

---



---



---



---

(8) Write an SQL statement that list the books borrowed most at least twice (show count). (4 marks)

---



---



---



---

*Print all table contents and Relational Diagram*