# SOUTH WEST REGIONAL MOCK EXAMINATION
# GENERAL EDUCATION

**THE TEACHERS' RESOURCE UNIT (TRU)**
Cellule d'appui à l'action Pédagogique

**IN COLLABORATION WITH**
En collaboration avec

**THE REGIONAL INSPECTORATES OF PEDAGOGY AND**

**THE SUBJECT TEACHERS' ASSOCIATIONS (STA)**

THURSDAY, 30TH MARCH 2023 ADVANCED LEVEL

| | |
|---|---|
| Subject Title | **Computer Science** |
| Paper Number | **Paper 3** |
| Subject Code Number | **0795** |

**TWO HOURS**

---

**INSTRUCTIONS TO CANDIDATES:**

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

*Great importance is attached to accuracy, layout and labeling of drawing generated outputs*

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

*Write algorithms in your 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.*

*You are expected to print out a single copy of relevant fragments of your program at different times.* ***Please notify the instructor of any required print out that was not done!***

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

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

*Where information is provided as soft copy, notify the instructors if it is not found in your machine or has not been made available to you*

**SECTION A: DATABASE**

The library attendant of your school uses an exercise book to keep the records of students who borrow books from the library. This manual system is inefficient and the library attendant wants you to computerise the system by implementing a small database that will help her manage the library. An extract from the library attendant's exercise book is given in the table below. Study the table and carry out the tasks below that will help you design and implement a database for the school library.

| StudentID | StudentName | Gender | DOB | Class | BookID | BookTitle | ReturnDate |
|---|---|---|---|---|---|---|---|
| 21LS001 | Alem Joseph | M | 03/01/2006 | LS2 | BK01 | A/L Chemistry | 12/11/2022 |
| 21LA010 | Abianga Moses | M | 24/06/2005 | LA4 | BK02 | A/L History | 01/12/2022 |
| 22LA015 | Bih Anita | F | 22/04/2006 | LA2 | BK02 | A/L History | 05/11/2022 |
| 22LS020 | Enanga Laura | F | 29/08/2007 | LS1 | BK03 | A/L Computer Science | 10/01/2023 |
| 21LA010 | Abianga Moses | M | 24/06/2005 | LA4 | BK04 | A/L Mathematics | 10/01/2023 |

**Table 1**

**Task 1: Normalisation (6 marks)**

i. One functional dependency that exists amongst the fields in **Table 1** is:
{StudentID, BookID} → {ReturnDate}
Write down **all** other functional dependencies that exist amongst the fields in Table 1. **(3 marks)**

ii. Table 1 is not in Second Normal Form (2NF). It can be converted into 2NF by decomposing it into three relations **Students**, **Books** and **Loans** which are of obvious meanings.
In your answer booklet, represent the three relations *Students*, *Books* and *Loans* in 2NF using the format:
*table_name(field1, field2, field3,…), where field1 (underlined) is the primary key*. **(3 marks)**

---

**Task 2: Implementation (14 marks)**

i. In your favourite database management system (DBMS), create a database named **Library** which contains the relations **Students**, **Books** and **Loans** as defined in **Task 1**. Write down the SQL statement you have used to create the database. **(3 marks)**

ii. In your answer booklet, write down an SQL statement to create the Students relation from **Task 1**. **(3 marks)**

iii. Create relationships between the relations in the **Library** database. Show the degree of each relationship.
Do a ***screenshot*** of the relationship diagram, save it as ***Relationship*** and then ***print*** a copy or draw the relationship diagram in your answer booklet. **(2 marks)**

iv. Populate the tables in the **Library** database with the data from **Table 1**. Print each of the tables in Datasheet view. **(3 marks)**

v. In the Library database, write an SQL query that returns for each student ID and name, the number of books he/she has borrowed. Save and print the query (in SQL view) or ***copy*** it in your answer booklet. **(3 marks)**

---

## SECTION B: PROGRAMMING

Modern computers communicate and operate with binary numbers which use only 2 digits, 0 and 1. This means that for computers to manipulate real-world data, the data needs to be converted into binary form of 0s and 1s and when the computer wants to output data, it has to convert the data back into a form understandable by humans.

You are required to write a program that will enable a computing device to perform conversion of whole numbers from decimal to binary and vice versa. When the program is run, it should prompt the user to enter a base which should be 2 or 10. If the base entered is valid, the program prompts the user to enter a number in the chosen base. Otherwise, the program displays an error message and prompts the user to enter a valid base. If the number entered by the user is a base 10 number, the program converts the number to binary and then displays the binary number obtained. If the number is a base 2 number, the program converts it to decimal and then displays the decimal number obtained. Below are sample outputs for different expected scenarios.

```
Enter a base: 10
Enter a base 10 number: 99

Decimal: 99
Binary: 1100011
```

```
Enter a base: 2
Enter a base 2 number: 11001

Binary: 11001
Decimal: 25
```

```
Enter a base: 2
Enter a base 2 number: 123
Not a binary number.

Enter a base 2 number:
```

**Task 3: Program Design.** ***Answer all the questions in this task in your answer booklet.*** **(10 marks)**

To convert a decimal number into binary, the following steps can be used:

*Step 1*: divide the decimal number by 2 and then retain the remainder

*Step 2*: divide the quotient from the previous division by 2 and then retain the remainder

*Step 3*: repeat step 2 until the quotient becomes 0

*Step 4*: print the remainders in reverse order

Write a pseudo code algorithm that gets a positive integer, *num*, and then converts it to binary as described above. In your algorithm, use a loop to implement the steps described above, an array **R** to hold the remainders, a variable **count** to keep a count of the number of remainders (or bits), the **mod** operator which returns the remainder of a division and the **div** operator for integer division. Array indexing should start from 0. **(4 marks)**

The decimal number 25 has been converted following the procedure above and the remainders have been stored in the array R as shown below.

| | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [...] |
|---|---|---|---|---|---|---|---|---|---|
| R = | 1 | 0 | 0 | 1 | 1 | | | | ... |

Modify the pseudo code fragment below so that it can be used to correctly display the binary equivalent of any decimal number after conversion. *The variable* **count** *is obtained from* ***Task 3 (i)*** *above.* **(2 marks)**

```
for(i ← 0 to count)do
  print(R[i])
next i
```

i. When the user enters a base, it needs to be checked and an error message displayed if the base is not valid. The process is repeated until a valid base is entered.

(a) When is a base valid? **(1 mark)**

(b) The pseudo code fragment below can be used to implement the process of inputting and validating a base.

Copy and complete the pseudo code fragment by filling the spaces labeled **1**, **2** and **3**. **(3 marks)**

```
valid_base ← 0
do
  print("Enter a base:");
  get(base)
  if(1.....)then
    valid_base ← 1
  else
    2.....
  endif
while(3.....)
```

**Task 4: Implementation (20 marks)**

***Open your favourite Integrated Development Environment (IDE) and carry out all the activities below.***

i. Convert the pseudo code in **Task 3(i)** above into a programming language (PL) procedure called **`decBin()`** that accepts a positive integer, **`deci`**, as parameter and then converts it to binary. The procedure **`decBin()`** should display the binary number after conversion. *Make sure you declare every variable used.* **(3 marks)**

ii. Write a PL procedure called **`binDec()`** that accepts as parameter, a binary number **`bin`** and then converts it to decimal. To convert a binary number to decimal, use the following pseudo code fragment in your procedure.

```
while(bin ≠ 0)do
  rem = bin mod 10
  bin = bin div 10
  sum = sum + rem *(2^p)
  p = p + 1
endwhile
```

***Hint**: `2^p` means 2 raised to the power p.*

***For C programmers**, use the function pow() in the math.h library. So, `2^p` becomes pow(2,p).*

***For Pascal programmers**, use the function power() in the math unit. So, `2^p` becomes power(2,p).*

Make sure all the variables you use are properly initialized. The procedure **`binDec()`** should display the decimal number after conversion. **(3 marks)**

iii. In your main program,

(a) Convert the pseudo code fragment in **Task 3(iii)(b)** into PL statements. **(3 marks)**

*Insert a label, **`L0:`**, before you continue with the next activity. You can call on the supervisor/invigilators to assist you with this.*

(b) Prompt the user to enter a number in the base chosen. Read the number and store it in a variable called `num`. **(2 marks)**

(c) Call the procedures **`decBin()`** and **`binDec()`** appropriately in order to display the number inputted in Task 4(iii)(b) in decimal or binary. A selection statement may be needed to display the output as required. **(3 marks)**

iv. When a binary number is entered by the user, it needs to be checked to make sure it is a valid number in binary. If it is not a binary number, an error message is displayed.

Convert the pseudo code fragment below into PL statements and insert where appropriate in your program. **(3 marks)**

```
Set a variable check to the binary number entered
while(check ≠ 0)do
  if(check mod 10 > 1)then
    print("Not a binary number!")
    goto L0:
  else
    check ← check div 10
  endif
endwhile
```

v. Run you program for the following conversions and print the outputs.

- Convert the binary number 11101101 to binary
- Convert the decimal number 999 to binary **(2 marks)**

vi. Print your source code. **(1 mark)**