REGISTRATION CENTRE NUMBER	CENTRE NAME	
CANDIDATE'S FULL NAMES		
CANDIDATE IDENTIFICATION NUMBER	SUBJECT CODE 0795	PAPER NUMBER 3

FOLD HERE

FOR OFFICIAL USE ONLY (Candidate Random Code): ▶	

GENERAL CERTIFICATE OF EDUCATION BOARD

General Certificate of Education Examination

ADVANCED LEVEL

SUBJECT TITLE	SUBJECT CODE	PAPER NUMBER
COMPUTER SCIENCE	0795	3
	EXAMINATION DATE:	JUNE 2026

Duration: TWO HOURS

Enter the information required in the shaded boxes above.

Do not write in pencil.

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 drawings and computer generated outputs.

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

Record all your answers in the spaces provided in this question booklet. Also record in your question booklet any information requested or that you believe would make it easier to understand how you carried out tasks or answered questions. Blank spaces have been provided at the end of this booklet in case you need additional space for your answers or rough work.

Make sure all your answers, including printed works are submitted with your question booklet.

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.

FOR EXAMINERS' USE ONLY	*SCORE*
Marked by:.....	
Signature: ..... *Date:*	
Checked by:.....	
Signature: ..... *Date:*	

Turn Over

SECTION A: DATABASE (Nina's Shop) **(20 marks)**

The relation R below keeps records of customers and the items they buy from Nina's shop.

R ≡

CustomerID	CustomerName	ItemCode	ItemName	UnitPrice	Quantity	TotalPrice	TDate
C001	Ebai Paul	IT001	Rice	200	3	600	09/09/2021
C001	Ebai Paul	IT002	Beans	250	2	500	09/09/2021
C001	Ebai Paul	IT003	Corn	175	2	350	10/10/2021
C002	Eyere Esther	IT003	Corn	175	5	875	11/12/2021

Relation R has the following attributes with names of obvious meanings; the primary key field is underlined:

R(CustomerID, CustomerName, ItemCode, ItemName, UnitPrice, Quantity, TotalPrice, TDate)

Task1: Database Design **(6 marks)**

(i) In a fully normalised relational database the data in the relation R is stored in the relations: Customer, Item, and Customer_Item. Customer_Item holds details about items bought by customers. Complete each of these relations making sure that their primary key attribute(s) is /are underlined.

Customer(.....

.....) **(1 mark)**

Item(.....

.....) **(1mark)**

Customer_Item(.....

.....) **(2 marks)**

(ii) Complete the entity-relationship diagram below for the entities Customer, Item and Customer_Item

Customer

Item

Customer_Item

Task 2: Database Implementation **(6 marks)**

In your favourite Relational Database Management System (RDBMS), use SQL statements to implement the following.

(i) Create a database called NinashopDB. Write down the SQL statement used. **(1 mark)**

.....

.....

(ii) In NinashopDB, create the tables Customer, Item, and Customer_Item. Write down the SQL statement used to create the Customer_Item table. **(3 marks)**

(iii) Use a suitable SQL command to rename the attribute *Quantity* to *Qty*. Write down the SQL statement used. **(2 marks)**

Task 3: Database Manipulation **(8 marks)**

(i) Use the SQL **Insert** command to populate the three tables in `NinashopDB` with data from **R**. Write down the SQL statement(s) used to insert data into the table `Customer`. **(2 marks)**

(ii) The price of corn just increased to 200. Use the SQL **Update** command that effects this change in the database. Write down the SQL statement used. **(2 marks)**

(iii) Print the tables `Customer`, `Item`, and `Customer_Item`. **(1 mark)**

(iv) Write down SQL statements to retrieve all customers who bought beans, showing the quantity and total price. Print a copy of your result. **(3 marks)**

.....

.....

.....

.....

.....

.....

.....

.....

SECTION B: PROGRAMMING (Population Growth Program) **(30 marks)**

You are to write a program that determines how long it takes for a population to reach a particular size.

Suppose we have a population of n cows. Each year, $n/3$ new cows are born, and $n/4$ cows pass away. So, for example, if we have $n = 1200$ cows, then in the first year, $1200/3 = 400$ new cows are born and $1200/4 = 300$ cows pass away. Therefore, at the end of that year, we have $1200 + 400 - 300 = 1300$ cows. Repeat the process with the new population size until the desired target population is reached.

Fractional cows in intermediate computations are discarded. So, for example, if there were 1001 cows initially, the number of new cows would be $1001/3 = 333.67$ cows rounded down to 333 cows.

Task 4: Problem Understanding **(6 marks)**

(i) Determine how long it will take for the population to grow from:

(a) 1200 to 1300 cows. **(1 mark)**

.....

.....

.....

.....

(b) 27 to at least 37 cows. **(3 marks)**

.....

.....

.....

.....

(ii) How many years will it take for a population of 4 cows to become zero, if no new cows are born? **(2 marks)**

.....

.....

.....

.....

.....

Task 5: Design **(6 marks)**

Write the following algorithms in the spaces provided.

(i) Write a pseudocode function called `InputStartSize()` that prompts the user for a starting population size and returns the value input. A valid starting population is at least 9. Otherwise, the function repeatedly prompts for a valid input. **(3 marks)**

(ii) Write a pseudocode function called `CalculateYears(start_size, end_size)` that calculates and returns an integer number of years required for a population of `start_size` to reach at least `end_size`.
***[Hint**: Repeatedly use the newly calculated population size to compute the new population until at least the target population is reached.]* **(3 marks)**

Task 6: Program Implementation **(15 marks)**

In your favourite integrated development environment (IDE), carry out all the activities below. Use either C or Pascal programming language (PL).

(i) Convert the algorithms in Task 5 (i) and (ii) above to the PL functions `InputStartSize()` and `CalculateYears(start_size,end_size)` respectively. **(6 marks)**

(ii) Write a PL function called `InputEndSize(current_size)` that prompts the user for a target population size and returns the value input. A valid input is at least the current population size. Otherwise, the function repeatedly prompts for a valid input. **(4 marks)**

(iii) Write a main program that appropriately calls the subprograms in (i) and (ii) above in order to calculate the number of years (`noOfYears`) it takes for a population of size `start_size` to reach the size `end_size`. Prefix `n`, the number of years calculated and output, with a suitable label. So, for example, one can have for `n` the labeled output: "`Years: n.`" **(4 marks)**

(iv) Save your program as ***population.c*** or ***population.pa***s based on your choice of PL and then print a copy. **(1 mark)**

Task 7: Sample Output **(4 marks)**

Run your program for the following sequence of inputs. Make a screen capture of the output in each case, save it and then copy. [*Note*: *The outputs of multiple input values can be captured in the same screenshot.*]

(i) Start size: -5.
Start size: 3.
Start size: 9.
End size: 5.
End size: 18. **(2 marks)**

(ii) Start size: 20.
End size: 1.
End size: 10.
End size: 100. **(2 marks)**

STOP

GO BACK AND CHECK YOUR WORK