

June 2017 Paper Three

PROBLEM

Your computer teacher is not a programmer and he wants to use a computer application to record the marks for all his students. He needs this application to:

- Allow him give a student's number and put a mark against it,
- Classify the students in order of merit, and
- Search for a given student and publish his/her mark.

You are to propose database and programming solutions to the teacher's problem as detailed in parts 1 and 2 below,

PART 1: DBMS

Task 1

- (a) Create a database called EXAMS using a DBMS of your choice. **(2 marks)**
- (b) Create a table called MARKRECORD with two fields STUDENTNAME and MARK. Record the marks of ten students on the table. Marks are out of 20. Save and print the table **(4 marks)**

1	Tiko	3.5
2	Tambe	11.5
3	Ngum	17.25
4	Fabian	11.5
5	Ndong	9.75
6	Tata	12
7	Ayuk	13
8	Nchang	9.25
9	Ndongmo	16
10	Ibrahim	14.8

- (c) Give or create a query, called SUCCESSFUL, that displays the names and marks of all students who scored a mark of 10 and above, in order of merit. Execute this query and print its result. Save the query, possibly in a text file. **(4 marks)**
- (d) Give or create another query that displays the name and mark of the student Ayuk. Execute the query and print its result. Save the query, possibly in a text file. **(3 marks)**
- (e) Save the database in your candidate folder. **(1 marks)**

PART 2: PROGRAMMING

Begin

For i=1 to n do

t[i]. mark \leftarrow 0

t[i]. numb \leftarrow -1

end

For i=1 to n-1 do /**i=1 to n in Pascal **/

Prompt user to give number and mark

Read into result.num

Read into result. Mark

For j \leftarrow 1 to j \leftarrow i-1 do

If (result. mark < t[j]. mark) then

Aux \leftarrow t[j]

t[j] \leftarrow result

result \leftarrow aux

endif

endfor

t[i] \leftarrow result

endfor

end

Task 2

To develop a programming solution to the teacher's problem using an imperative programming language, you need some data structures:

- A record (call it **result**) for each student, containing integer and real fields for registration number (num) and mark respectively.
 - An array of these records.
- (f) In your answer booklet, give a declaration of the data structures in either C or PASCAL.
- (g) Write the programming language (PL) procedure, or function, called **insertsort**, that reads a registration number and the corresponding mark into the record **result** and inserts it into an array *t* of such records, in order of merit, for all *n* (*n*=10) students, as given in the algorithm above.
[NOTE: Transform the algorithm above into a suitable PL code. Also, $t[j]$ is the j^{th} entry of array *t*, and $t[j]$ mark is the mark field of the j^{th} entry.]
(12 marks)
- (h) Write another PL procedure (or function) called **search** that searches the array (table) for a given registration number and prints the student's number and mark, if found. However, if the number is not found the statement "**STUDENT NOT FOUND**" is printed. **(6 marks)**
- (i) Write a main program that calls the **insertsort** procedure, and then loops, printing all records in array *t*. afterwards, it prompts for a student registration number and calls the search procedure to look for it. It then prints the entry if found, or a suitable error message. The output of array *t* is in a tabular form with a colon separating each registration number and its mark, For example, we have:
- ```
3: 17.25
7: 13.00
```
- (j) Run the program, capture its terminal output, save it and then print it. Also save and print all source codes.