Data Structures & File I/O

(LAB 6)

Tasks 1 – 3

DT131C

Ebeten Alaga

2021-05-10

1. Introduction

This lab aims to provide practice on data structures and file input and output. The objectives include designing efficient data structures, application of pointers, and file input / output.

2. Design

Using the top-down approach, I broke the problem into a series of smaller problems and wrote a function to solve each small problem.

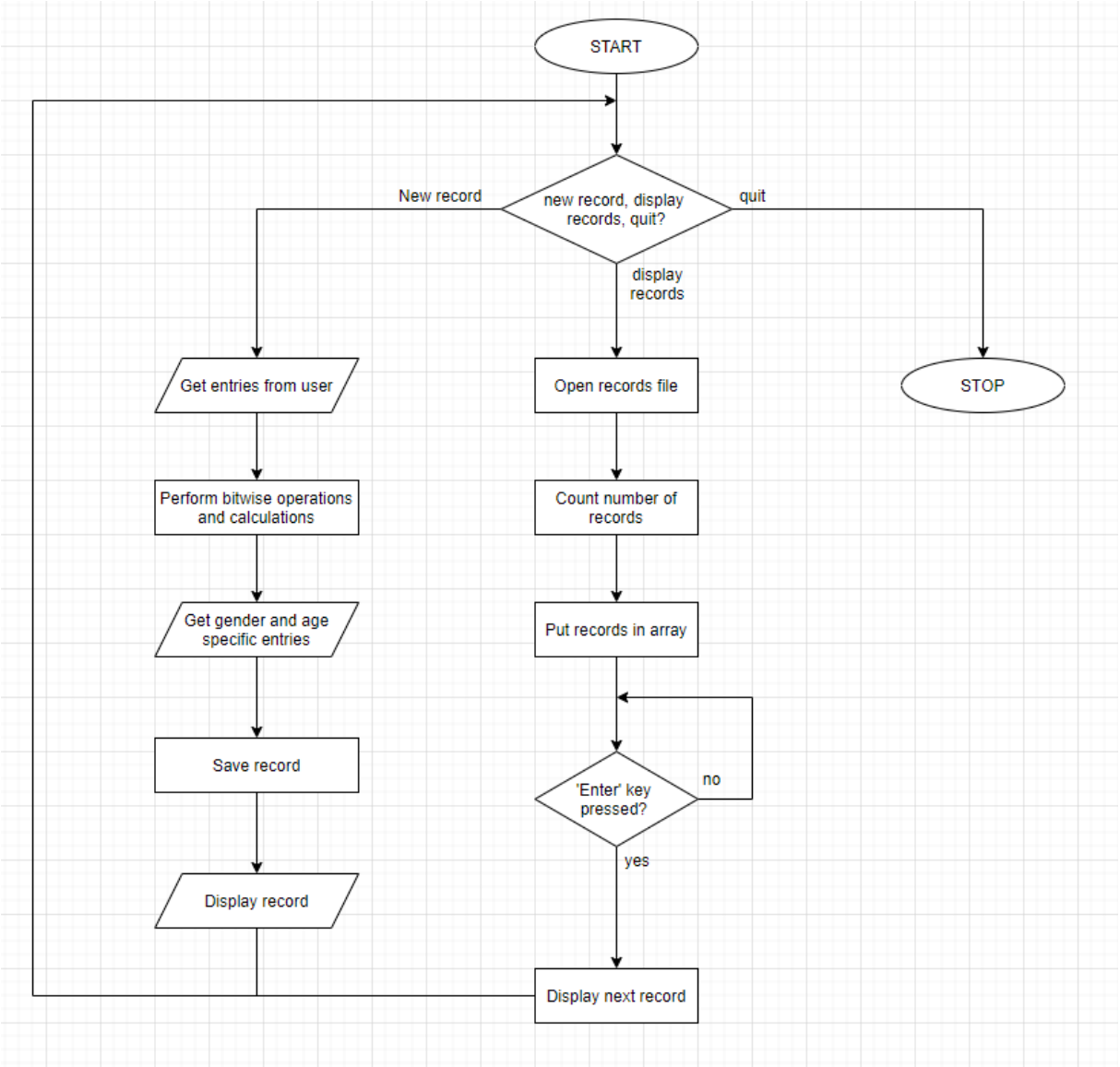
This approach makes it easier to achieve the overall aim and provides some measure of ease to change / improve on something without breaking the entire program.

Capturing New Patient's Information:

- Print questions to screen.
- Read input from keyboard.
- Where necessary, perform bitwise operations or calculation on input (e.g perform bitwise operations on vaccine history or calculate BMI).
- Save input to corresponding field.
- Save record.
- Display record.

Displaying Records Saved to File:

- Open file.
- Count records.
- Read records.
- Put records in array.
- Sort records by BMI in ascending order.
- Display record to screen.
- Show next record on 'Enter' key press.



Sample flowchart

3. Implementation

The program begins with a simple menu to check if the user intends to create a new health record or read records from file.

When entering new records, the user follows the on-screen prompts and supplies the required information. Care has been taken to make the data entry process as simple as possible, with the only requirements being that the user enters data in specified formats – such as entering weight in kilograms, height in meters and year in YYYY.

The user's involvement in displaying records on-screen is limited to scrolling and pressing 'Enter' to display the next record, so there is less room for error while displaying records.

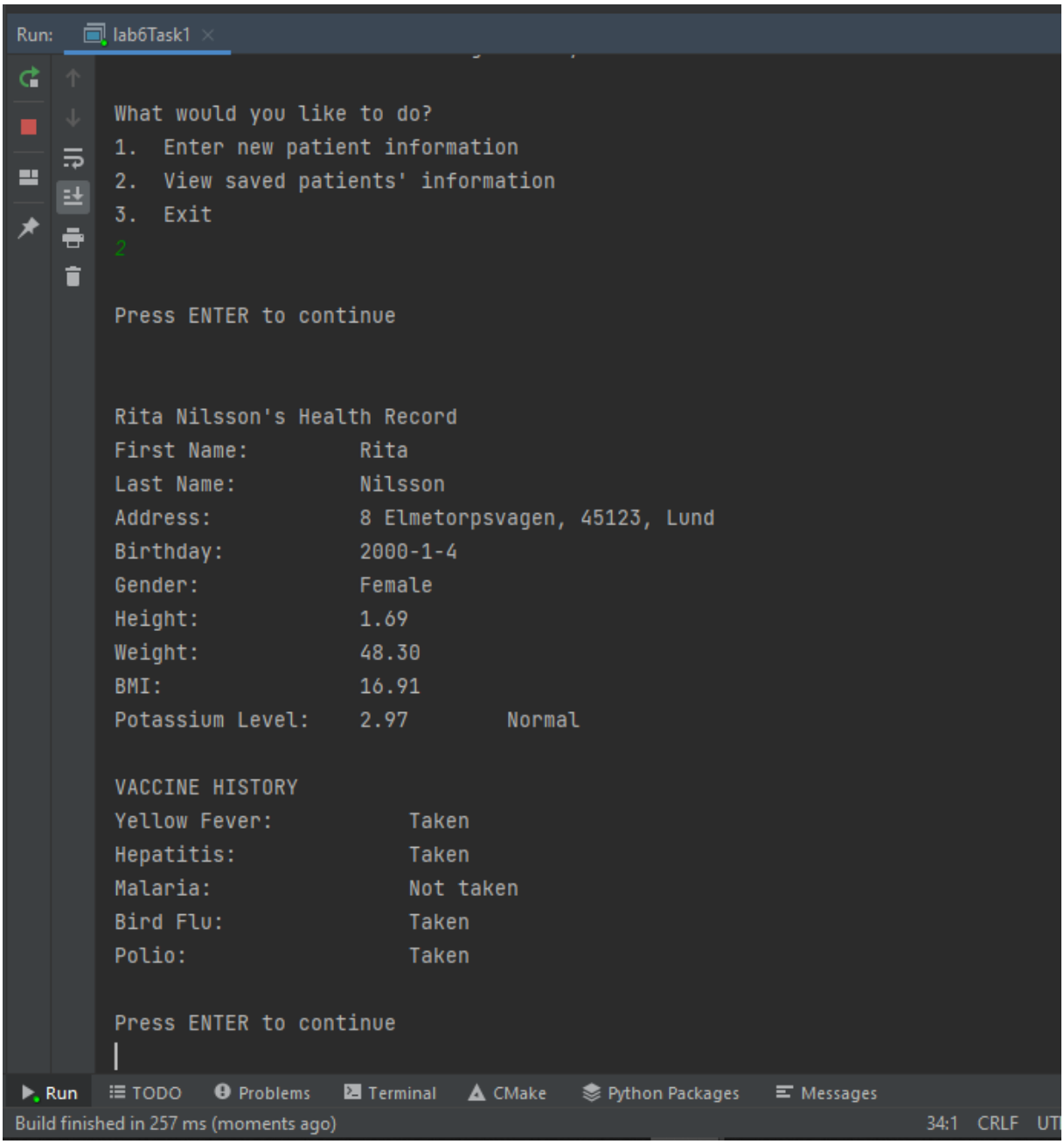
The program is also broken into several smaller functions, with each function doing the most basic of jobs to make it easier to understand.

```
74      //Function Prototypes
75      float calculateBMI(float weight, float height);
76      void getPatientProfile(healthRecord *patient);
77      void savePatientProfile(healthRecord *record);
78      void showPatientProfile(healthRecord *patient);
79      void readPatientProfile();
80      void setVaccineStatus(int *add, int vaccine);
81      school setEducation();
82      void menu();
83      void levelPrinter(int i);
84      void vaccinePrinter(int vac);
85      void sortRecords();
86      void swap(healthRecord *one, healthRecord *two);
87      void displayTen();
88      void exit(int x);
89
90      #endif //LAB6TASK1_LAB6TASK1_H
91
```

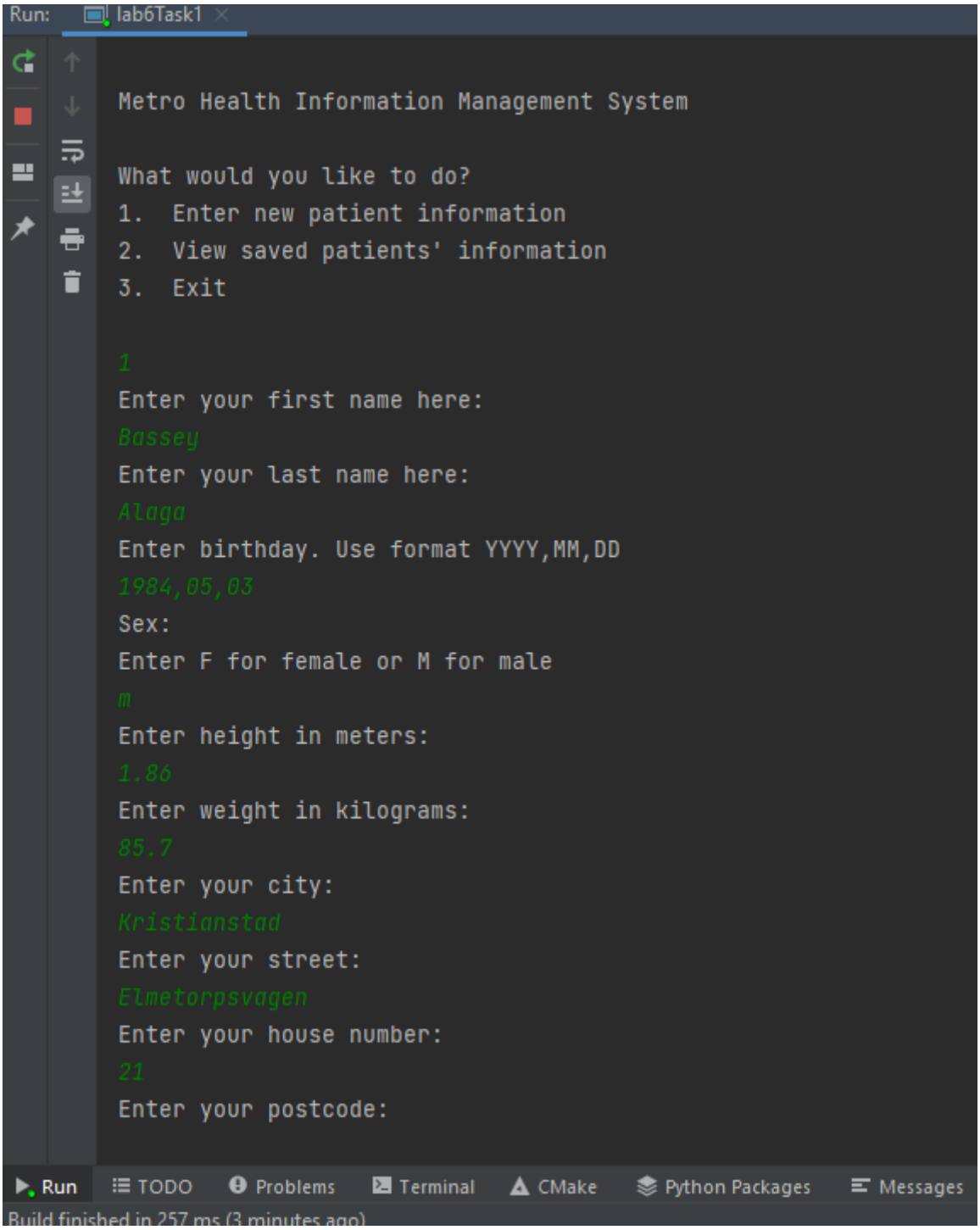
To avoid passing health records around these functions, pointers are used.

4. Test and evaluation

Use the test cases developed in the design stage to test your algorithms and implementation; present the results of the work performed using neatly organized and completely labeled tables and/or graphs whenever possible.



Sample records display screen



Sample record entry screen

5. Conclusion and what I have learnt

In this final section of the report, you should briefly bring everything together and reflect upon the results of the lab.

It would be good to improve your learning outcomes if you include what problems you have encountered and how you solve them, as well as what you have learnt from this lab.

When counting the size of the health record by hand, it was smaller than the result using the sizeof() function. I think this is like having a collection of files in a folder, when the size of the folder makes it slightly larger. I would liken the struct to a folder that holds the individual data elements that make up the health record.

In accomplishing this task, I have learnt more about how to use pointers, as well as how to use linked lists and memory allocation. I eventually discontinued the use of linked lists because the size of the struct and the number of records required to save as a test case for the program meant I would use up more memory by using linked lists. It was however a good opportunity to learn about them, and how they can be used.

One problem I faced was reading from the text file using “r” as read mode. It always read only one record, while “rb” reads all the records from the file even without using a loop.

6. References

No references were used in this report.