# Simple Electronic Health Record

Objectives

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

Task 1 – Design an Efficient Data Structure

Electronic Health Record (EHR) is nowadays widely used to manage patient health information. Keeping track of personal health information is key to getting the right care. The patient health profile allows patients or caregivers to keep track of important health information. If you want to know more about it, [read it at Wikipedia (Links to an external site.)](https://en.wikipedia.org/wiki/Electronic_health_record).

Some pieces of a patient’s health information are given below:

* First Name
* Last Name
* Date of birth
* Gender
* Height in cm
* Weight in kg to the nearest tenth
* BMI accurate to 2 decimal places. (Set it to 0.0 initially)
* Address consists of:
* City,
* Street name,
* Street number,
* Postcode (such as 29188)
* Vaccination History: Each of the following (Yes or No)
  + Yellow Fever
  + Hepatitis
  + Malaria
  + Bird Flue
  + Polio
* Additionally, the following information is stored depending on the patient type
* Sodium level (normal range 135-145) for adult man
* Potassium level (normal range 2.5 - 3.5) for adult woman
* School (preschool, preschool class, comprehensive school) for a child (A child is anyone below the age of 16)

In this task, you need to complete the followings:

1. The efficiency of the storage is very important in resource limited systems. Design a storage efficient structure named **healthRecord**to store patient health profile data. Make sure that your design makes your data structure **readable** and **understandable** to others.
2. Count by hand the number of bytes used to represent one healthRecord. Then compare it with the result found by C program using sizeof( ) operator. Discuss the difference and your understanding.

Task 2 – Design an Efficient Data Input and Output Functions

After the data structure is designed, it is time to design and implement functions to get data from the user (via keyboard input), show the patient profile data (on screen) and read/save the data from/to disk files. The functions should be designed to be efficient in execution time.

In this task, complete the following functions:

1. Write a support function called calculateBMI() to calculate the body mass index (BMI) for given weight and height. You can find the formula for BMI on the Internet at [https://en.wikipedia.org/wiki/Body\_mass\_index (Links to an external site.)](https://en.wikipedia.org/wiki/Body_mass_index).
2. Design and implement an input function named getPatientProfile() to read a patient profile data from the standard input file (keyboard by default).
3. Design and implement an output function named savePatientProfile( ) to store the data in a disk file.
4. Design and implement an output function named showPatientProfile( ) to display the data in the standard output file (screen by default).
5. Design and implement an input function name readPatientProfile( ) to read data from a disk file.

 Task 3 – Test and Evaluation

Write a driver program to test the functions you have implemented in the last task:

1. Get input of patient health records of 3 adult men, 3 adult women, and 4 children from the standard input file (keyboard), and save them to a disk file.
2. Read 10 saved profile data records from the disk file, sort them into increasing order based on their BMI values.
3. Display the sorted records on the standard output file (screen), one record a time, controlled by pressing Enter key.