|  |  |
| --- | --- |
| Ex No : 8  Date: 22.04.2019 | **MINI PROJECT** |
| Title | **Hospital Management System** |
| Team Lead | **185001138 Satya Bharathi K** |
| Team Members | **185001159 Sowmya R**  **185001142 Sharmi R**  **185001140 Shanmuga Priya M G** |

1. **Abstract**

Hospitals are the essential part of our lives, providing best medical facilities to people suffering from various ailments. It is necessary for the hospitals to keep track of its day-to-day activities and records. But keeping track of all the activities and their records on paper is very cumbersome and error prone. Thus, keeping the working of the manual system as the basis of our project we have developed and automated version of the manual system, named as “Hospital Management System”.

A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves; patient personal information and medical history, room and ward scheduling, pharmacy stock details, etc.

The main aim of our project is to provide a paper-less hospital up to 70%. It also aims at providing low-cost reliable automation to the existing systems. Furthermore, it aims at standardising data, consolidating data ensuring data integrity and reducing inconsistencies.

**2.Description of the Project:**

The hospital management system is basically designed to handle and manage the medical, financial, accounting and the corresponding processing of services that takes place in a hospital.

This project includes various functionalities like

-Storing the patient details right from admission to discharge

-Storing the number of available units of blood in the blood bank and modifying them when necessary.

-Storing the ward details for every inpatient along with details of the Doctor In charge

-Processing the stock details in Pharmacy

-Billing during discharge that combines all the facilities and records to calculate the final bill

-Allows patient to enter the symptoms and know about the possible ailments based on the symptoms given.

**3.Salient Features:**

The salient features of this project include

* Symptom-Ailment matching

This is an efficient feature that allows the patient to input their symptoms and know about the possible ailments they could have. This feature could

also be used by doctors who are in need of external help/ consultation

of sorts

* Blood donor finder

A suitable match is found by traversing though the patient records by considering their RBC count.

* Inpatient and Outpatient billing

These are done separately since a vast majority of charges will not be applicable for Outpatients. The Inpatient billing takes into account the ward charges according to the type of room booked.

**4.Pseudocode:**

* **Function: main()**

1. **Functions Used**

pat(),ward(),pharmacy(),blood\_bank(),house(),billing()

* **Function:pat()**

1. **Structures used**
   * + - struct patient - to store the details of the patient like the name of patient, guardian’s name,age,address,blood group,contact number etc.
       - struct medicalrecords - to store the medical details of the patient such as blood pressure count, RBC count and blood sugar count.
       - struct address- to store the address details like house number,street,city and state.

**2- Functions used**

menu() and pat()

**3.Files created: pat.DAP,temp.DAT**

* 1. Display the mainmenu using menu() and choose the required option to add,modify,search for or delete the patient records.
  2. Store the details in a file named pat.DAT for future use
  3. Display Menu
* add,
* edit
* search
* delete
  1. When addition of records is chosen,
* Read the inputs for the various patient details like the name, age, address, locality, bloodgroup etc from the user using the structure patient.
  1. When edit is chosen,
* Read UID number to modify
* Replace the new details into the original data of the file pat.Dat
  1. When search is chosen,

* Read the name of patient whose detail has to be searched for.
* Display the details of the patient
  1. When delete is chosen

* Create a temporary file temp.DAT
* Read the UID number of the patient to be deleted
* Write the remaining record details into the temporary file and rename it.
* Close file

* **Function:ward()**

1. **Structures used**
   1. struct ward -to store ward details like patient name, doctor name, room type,etc.
   2. struct checkin\_date,struct checkout\_date – to store the respective dates. They will be nested within ward ( )
2. **Functions used-**type(),check\_in(),check\_out(),display\_ward(),days\_calculate

1.Display Menu to choose Check in,Check out or diplay ward details.

1.check\_in()

-call type() {to allot AC rooms and Non AC rooms}

-Read room type, patient details

-Write it onto the file “ward.dat”

2.check\_out

-Read check\_out date

-Write the modified record onto the file “ward.dat”

3.display\_ward()

-Read and Print records (if status of room is occupied)

2.Press ‘n’ to leave ward system

* **Function:blood\_bank ()**

1. **Structures used:**
   * 1. struct bloodbank-stores group and corresponding units of blood
2. **Functions used:**

add\_bloodgrooup(),display\_units(),update\_units(),donor\_search()

1.Display menu to add blood group ,update ,display units or search for a potential donor.

1.add\_bloodgroup()

-Read blood group,units

-write it onto the file “bloodbank.dat”

2.update\_units()

-Read group name to be modified

-Get the number of units to be incremented or decremented and modify the structure member units

-Write the modified data onto the file.

3.display\_units()

-Read and Print each record

4.donor\_search()

-Read blood group to be searched

-Traverse the patient details to look for patients with the same blood group,check if they are of age.

-Check their platelet count to see if the patient is an eligible donor and then display name and address of such donors in a list.

* **Function:pharmacy()**

**1. Structures used:**

**a.** typedef struct med: stores the stock of medicines and their

details

**2. Functions used:**

read(), out(), search(), change(), display()

1. Display menu to add new arrivals (read()), display stock (out()), search for a certain medicine details (search() and display()), update stock quantities (change() and display()) and exit.

1. read()

- Read the medicine details

- write it in a file named pharm.dat

2. out()

-open file in read mode

- display details of the file in tabular alignment

3. search()

-read the name of medicine to be searched for

-return the pointer where the match is found to display the details

4. change()

- read the name of stock to be appended

- read the changes in the quantity of the stock

- change the value in the file

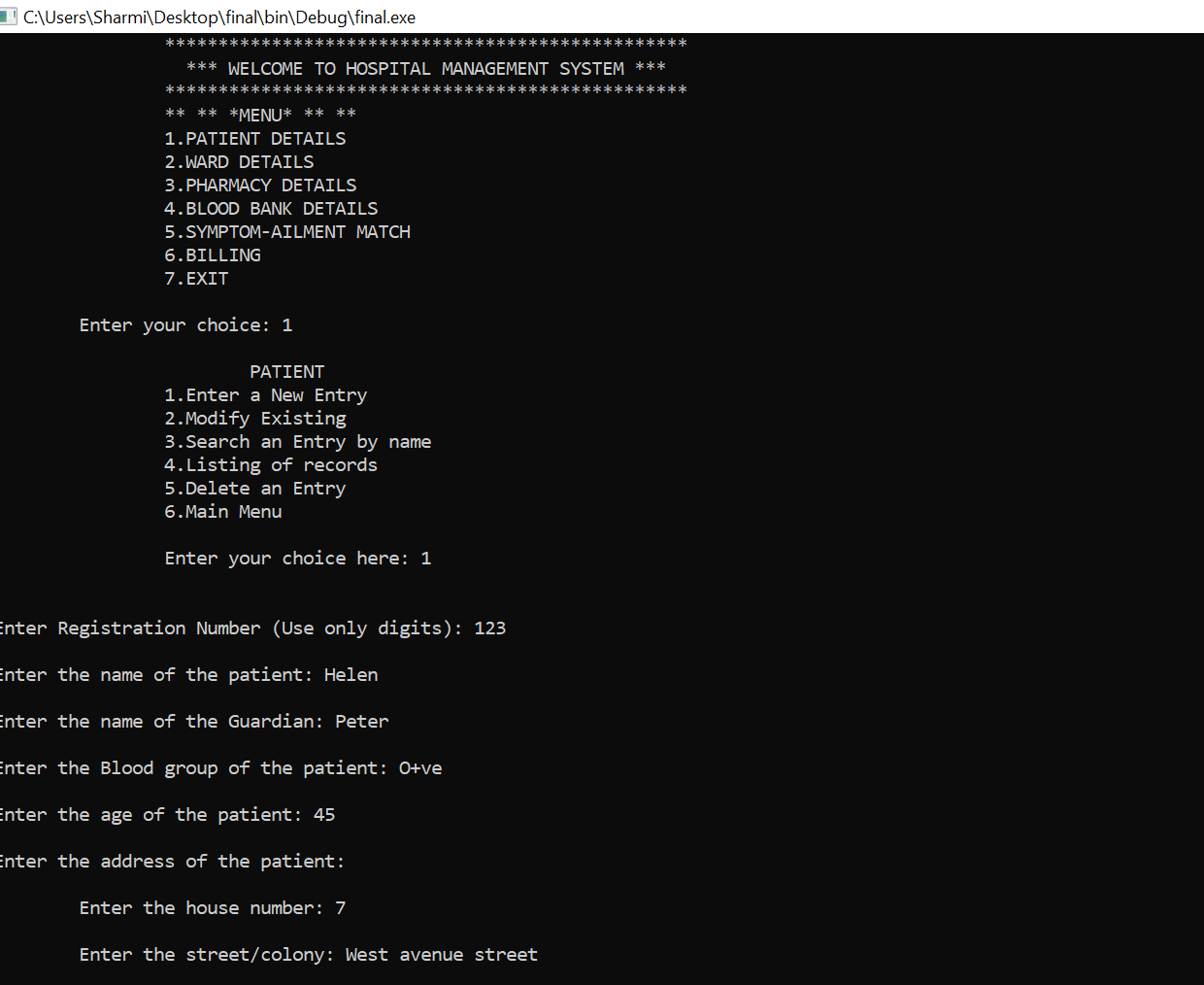
5. display()

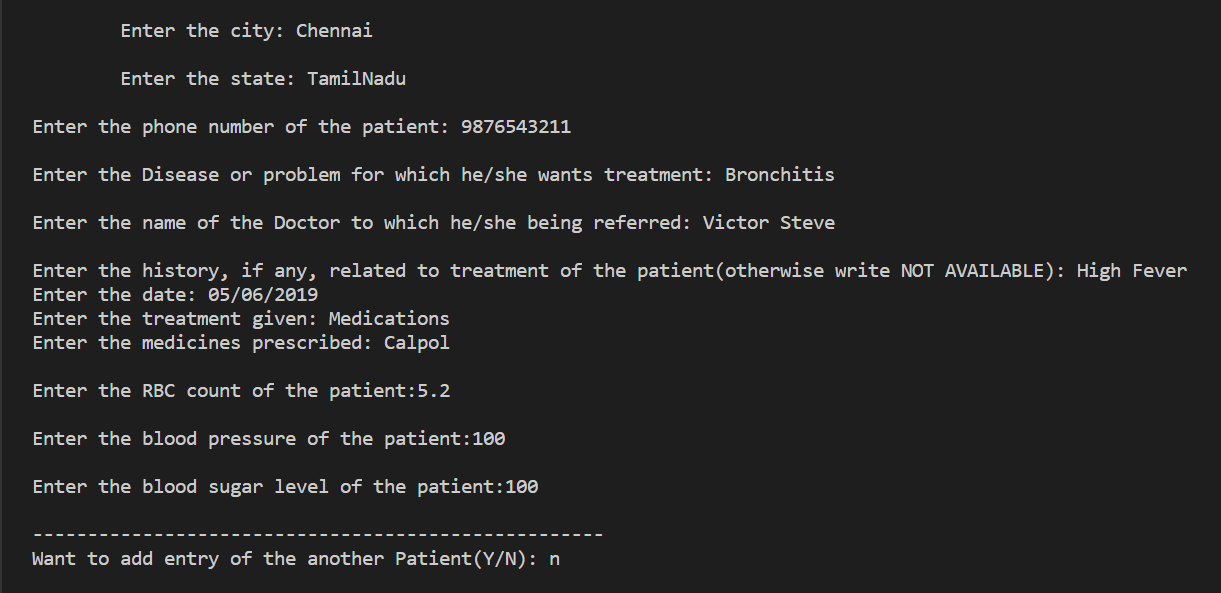
- open the file in read mode

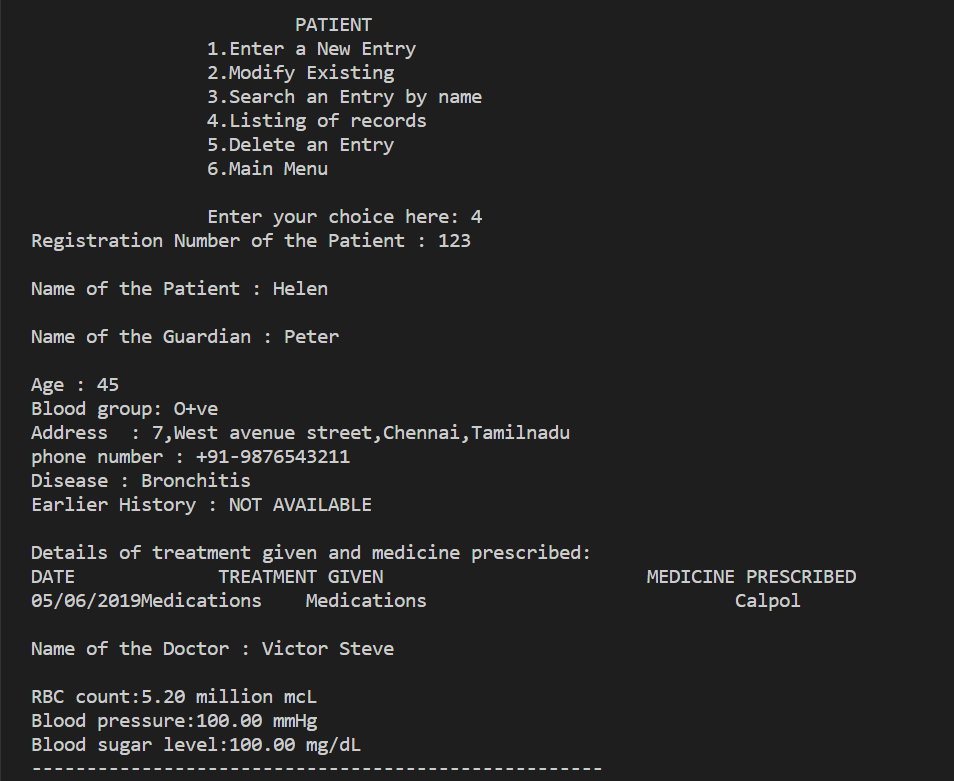
- display the required medicine and its details stored in the structure

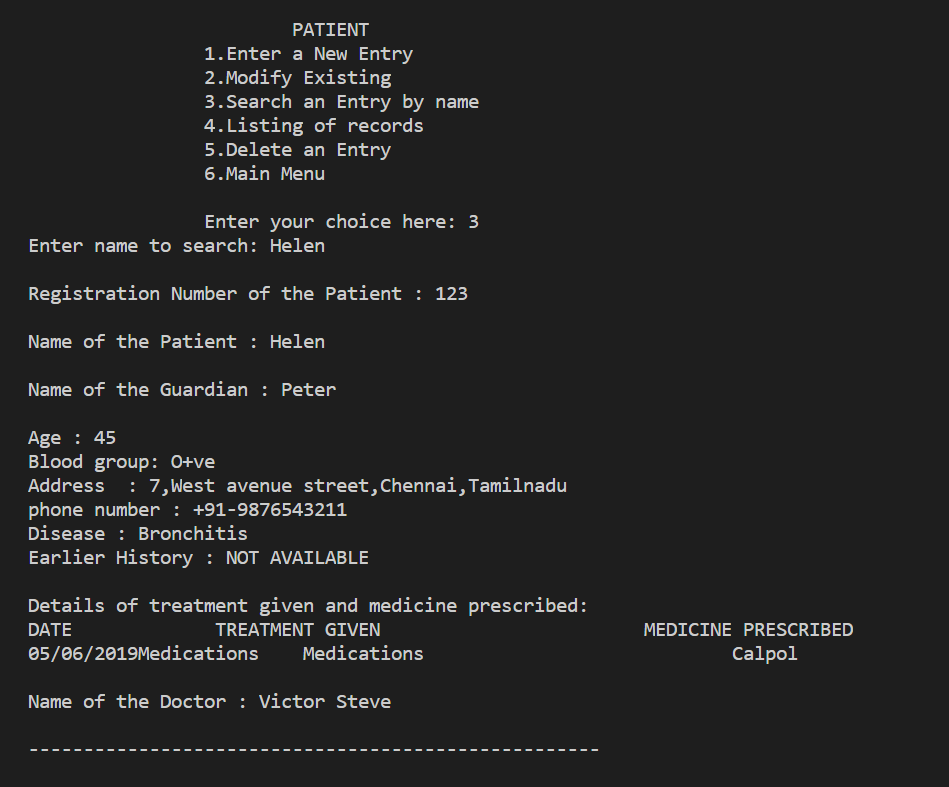
* **Function : house()**
  + **Structures used:** 
    - diagnose element- disease and score.
  + **Headers used:**
    - #include <collectc/hashtable.h>
      * a header file for dictionary
    - #include <collectc/array.h>
      * a header file for dynamic array
  + **Functions used:**
    - load()
      * takes a file with diseases and list of symptoms
      * makes a dictionary where symptoms are keys and an array of their corresponding diseases as values
    - getallsymptoms()
      * reads all the keys from the dictionary and displays it to the user
    - getusersymptoms()
      * it gets the users symptoms
      * it also reads all possible symptoms and searchess the dictionary for user entered symptoms
      * it gets all diseases and updates score by calling on maintain\_score()
    - maintain\_score()
      * it makes an array of structure diagnose\_element
      * it updates the score based on number of times it is encountered when searching
    - sortdiseases()
      * it sorts the array
      * uses temporary structure pointers
      * sorts them according to virtue of their score

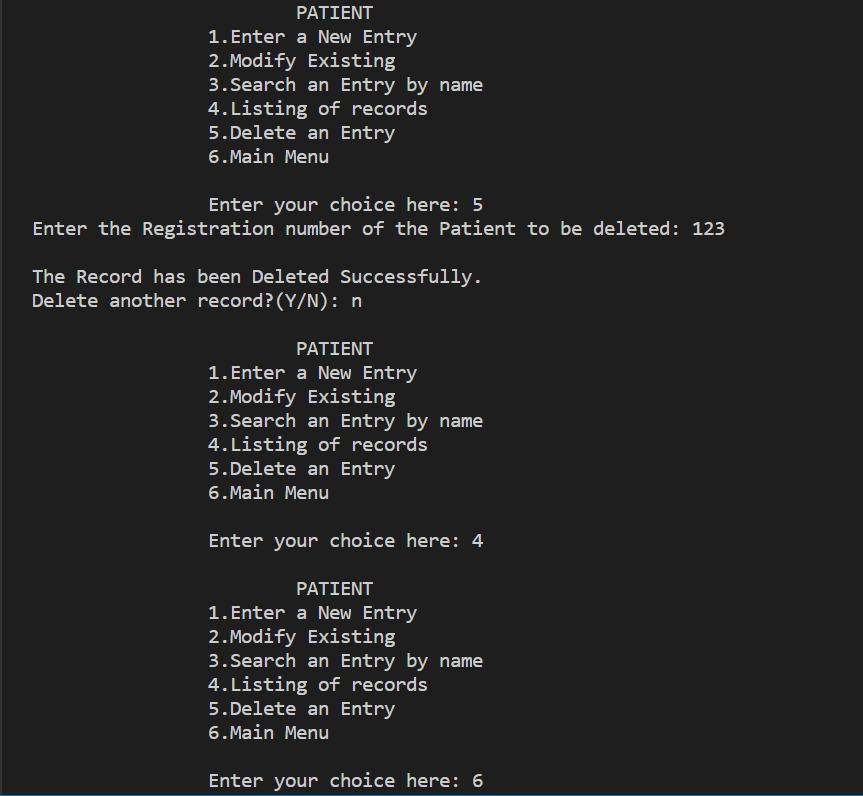
**5.Snapshots of Output:**

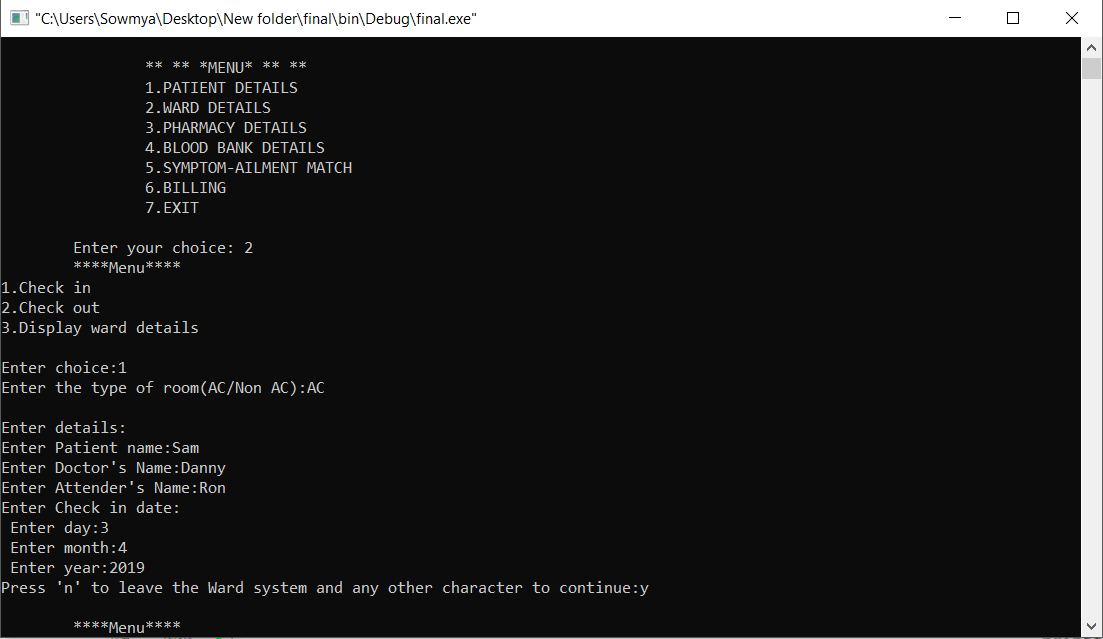


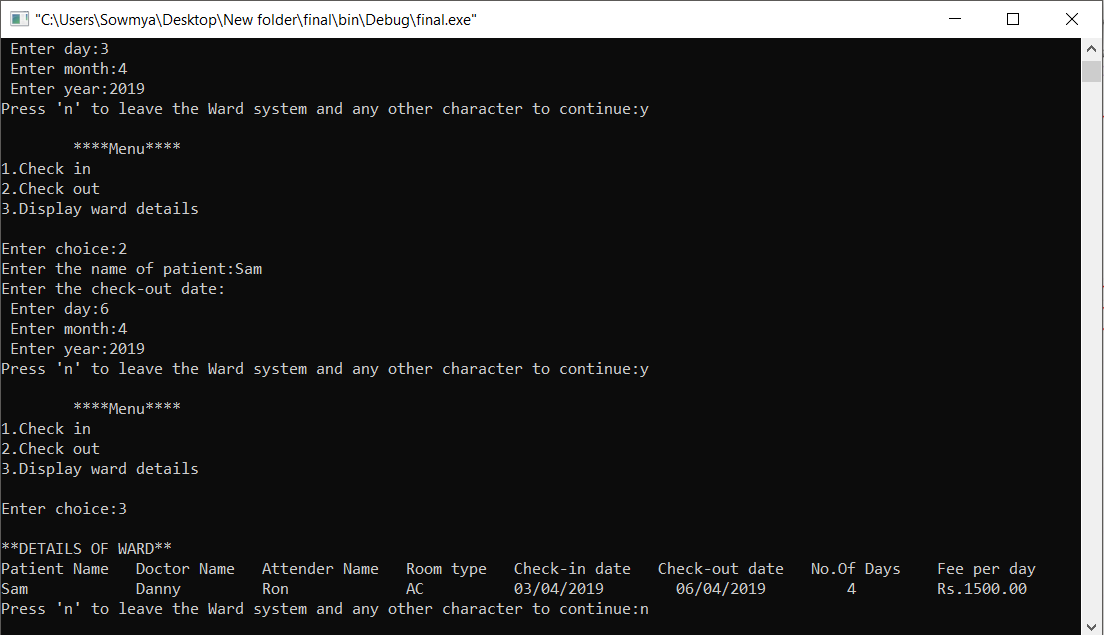


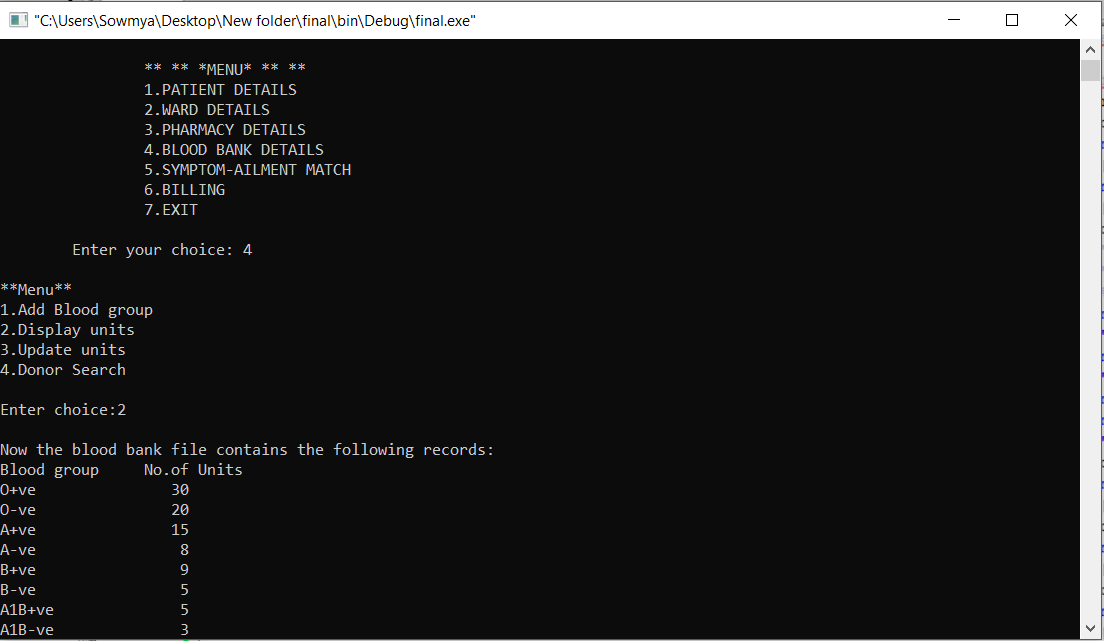


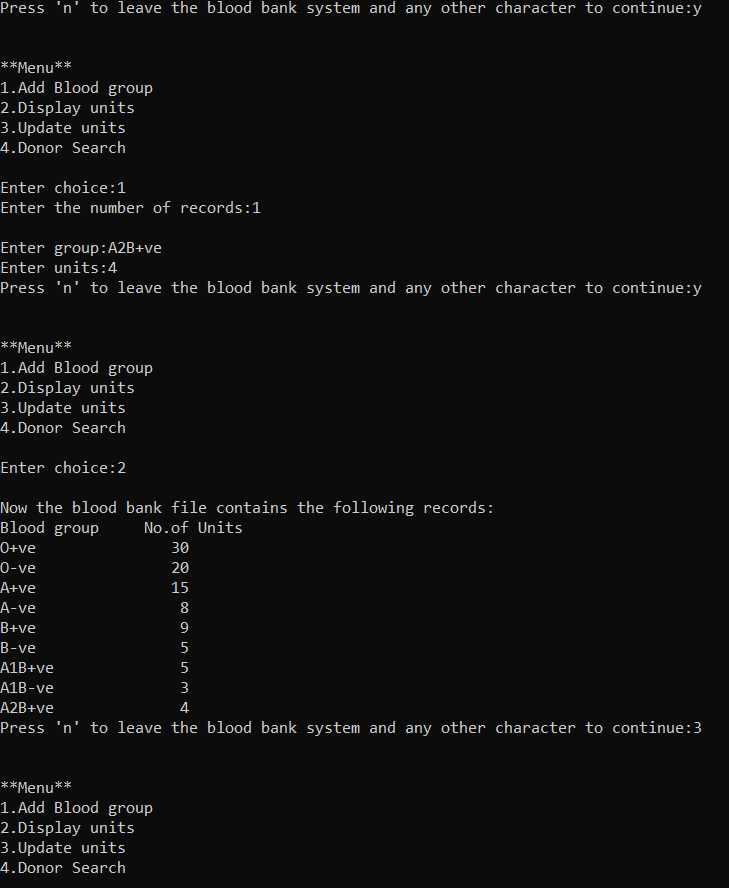


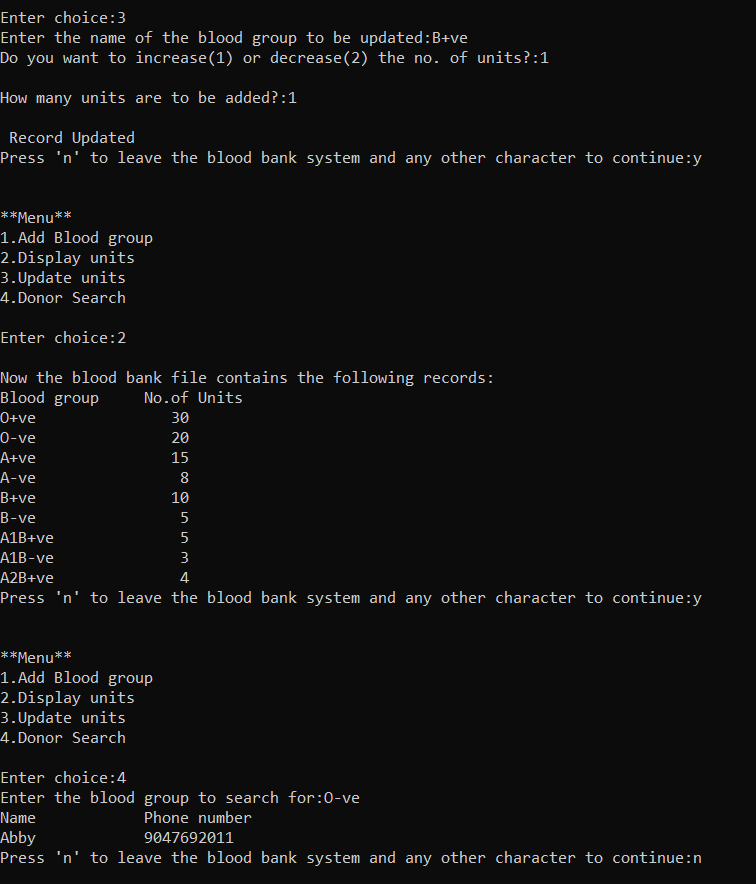


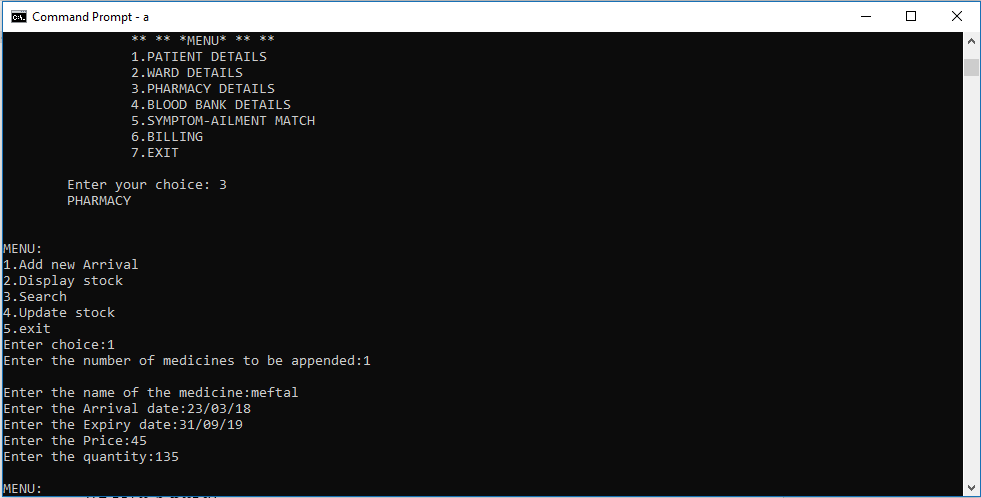


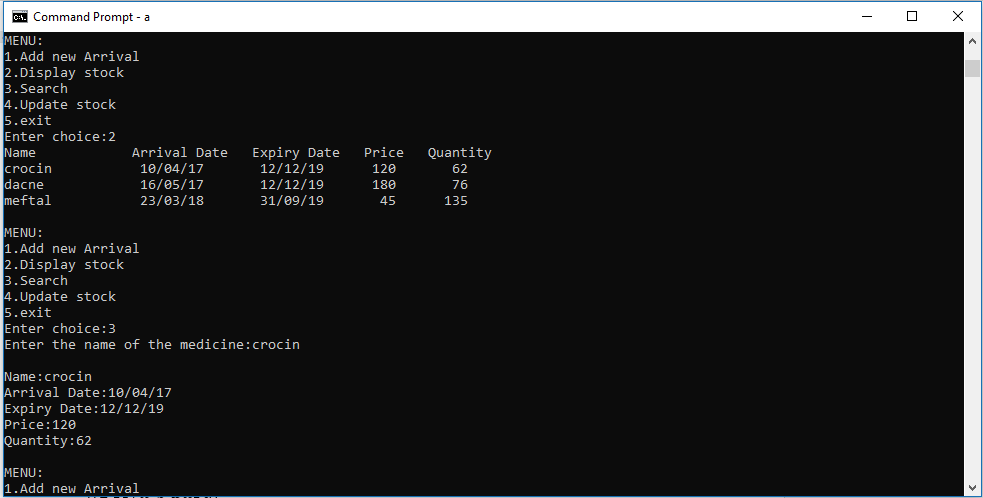


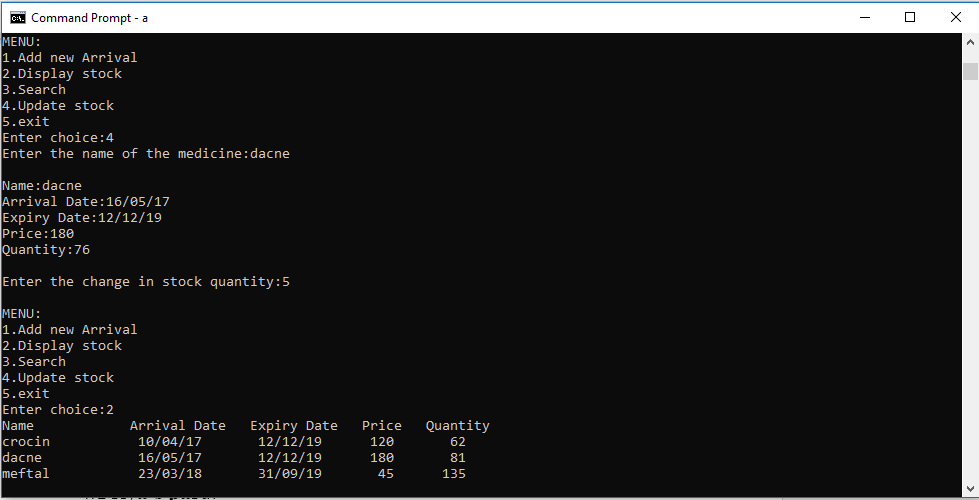


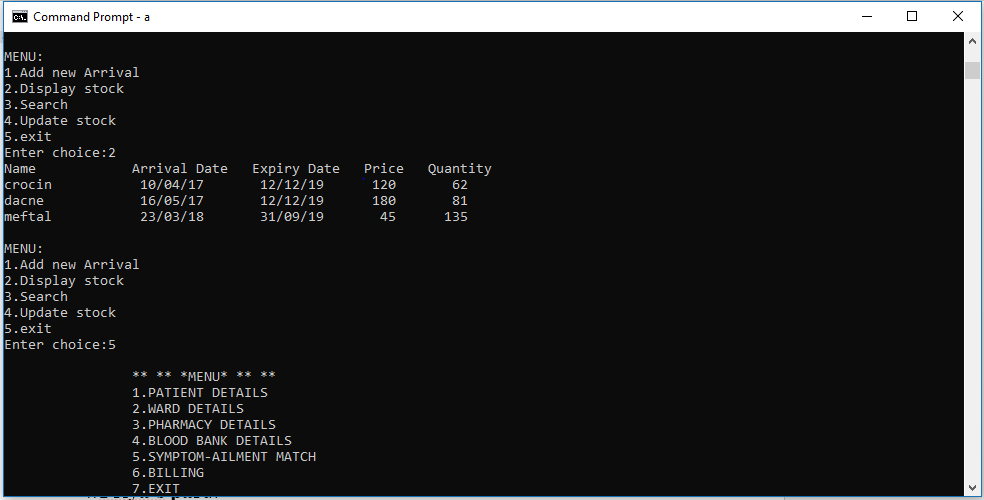


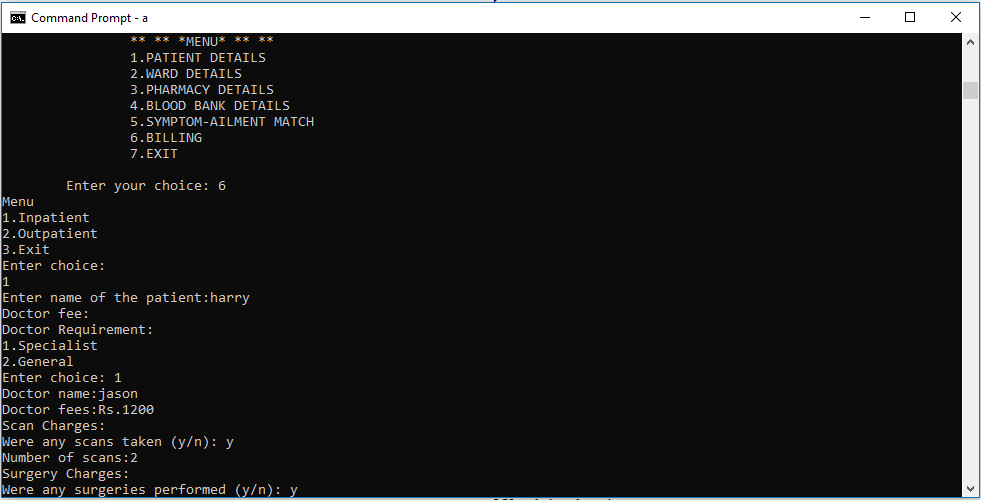


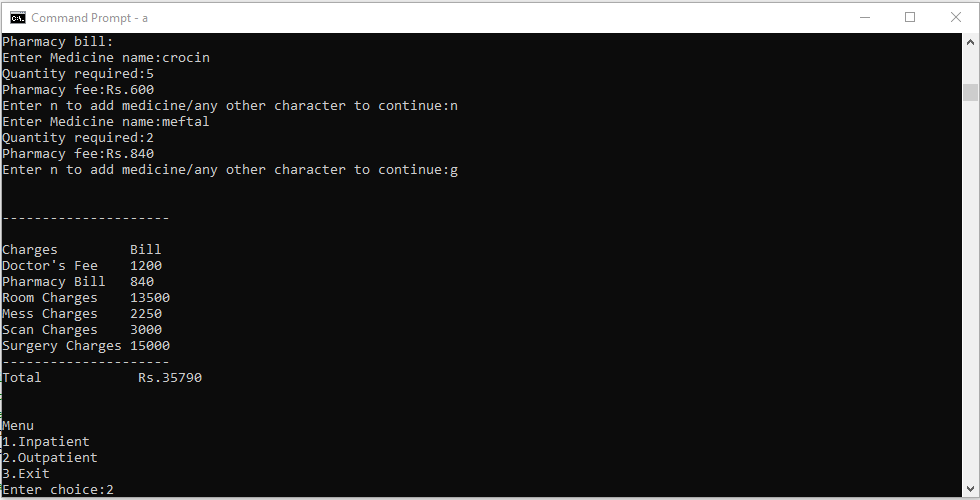
****

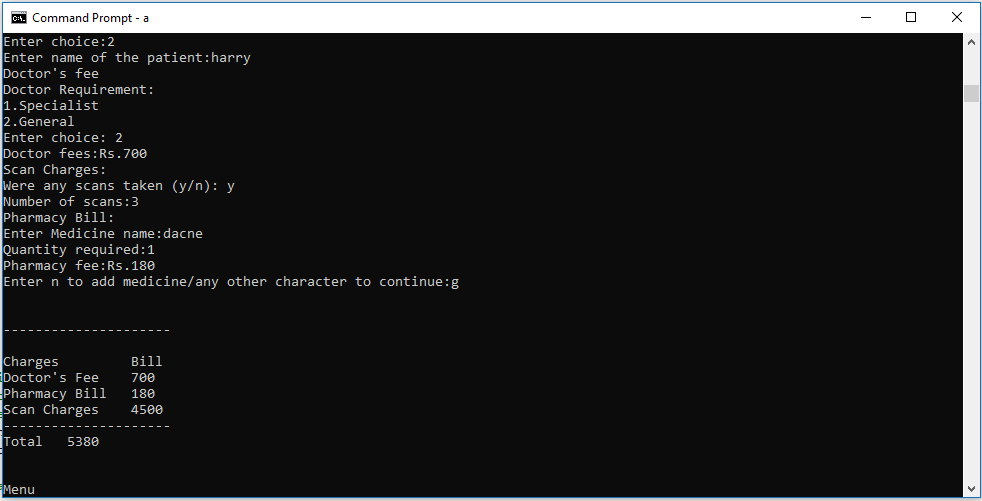
****

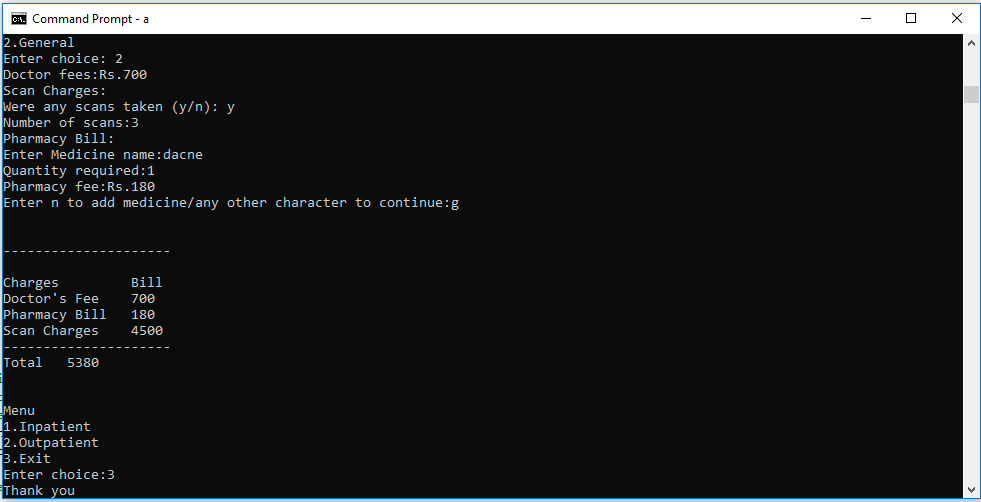
****

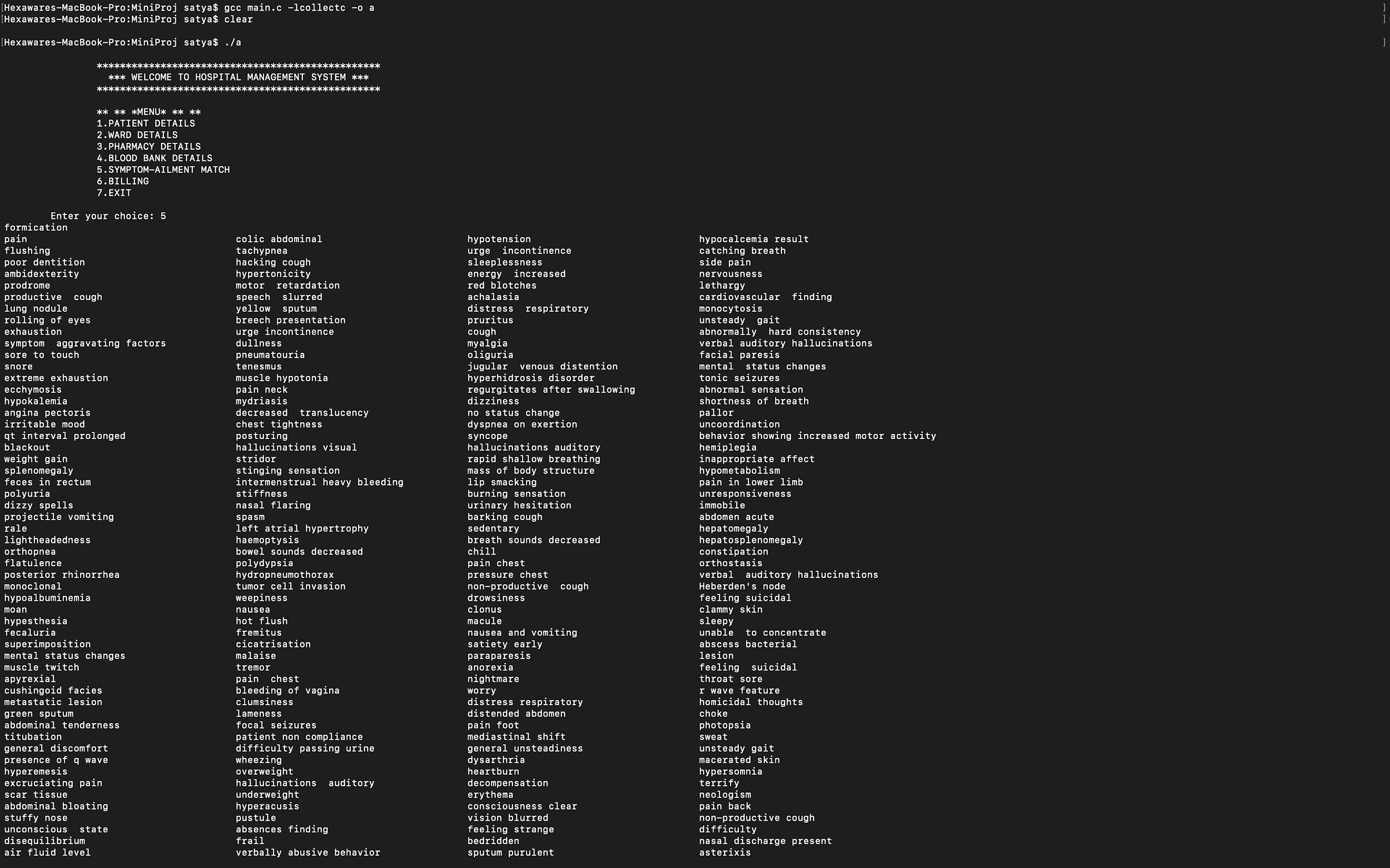
****

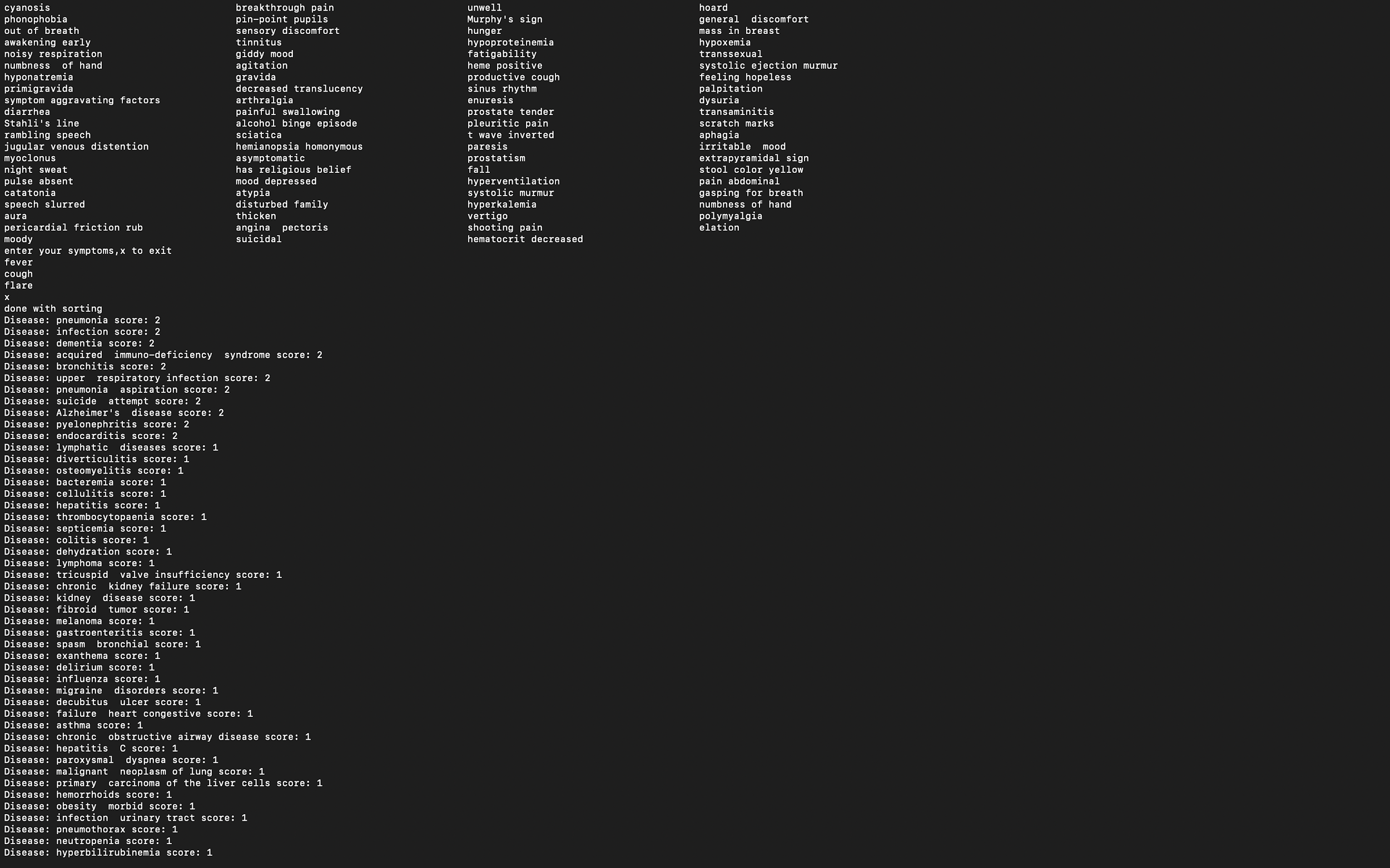
****

****

****

****

****

****

**6.Learning Experience:**

* The project enabled us to learn in detail about all the key operations that can done with the C PROGRAMMING language.
* It has aided our understanding of real-world applications by helping us to analyse technical problems as related to customer requirements.
* It made us understand the functionalities of C programming better.
* Working on this Hospital management system has helped us to learn how to integrate multiple functionalities by using separate functions to increase the efficiency and modularity of the code.
* It also enabled us to understand the Incremental program development and the use of Functions for better programming experience.
* Furthermore, it has helped us to design, correctly implement and document solutions to significant computational problems.