Specification for the program “Emergency Manager”

Name

Emergency Manager – Functions for managing emergencies in a hospital according to the priority of each patient.

Usage

```
emergency_manager <input-file>
```

Description

“Emergency manager” is a software for managing priority emergencies of patients in a hospital according to the severity of each one. This manager of emergencies should be able to set a new emergency by creating a new patient with the priority of emergency (CRITICAL, MODERATE, GENERAL), and the respective area; The system must also keep the date of entry into the system and the date of departure of the patient in order to have a history of the patient.
The order of turns of each patient depends of the priority and the people with the same priority are on a first come, first served basis. In case of severity of any patient is also possible to change the priority of emergency and then, the rest of patients should move backward/forward depending of the case. When a patient is treated, it is necessary to update the date of departure. To create/modify a new patient, the program should read a text file named with the National Identity Number of the patient.

To summarize, the program should be able to do the following actions:

- Create/Update patient
- Discharge patients
- Get a historical of a patient
- Filter patients by date and area
- Change the priority of a patient.
- Search patient

Options

The following commands are used to represent the listed functions above.

- **help**

  help command with no arguments to display the usage of the program.

- **enter_patient**

  With the file that is specified as program entry, a new patient is created in the system with the data provided in the file, as well as the date of entry.

- **next_patient <ARGUMENT1>**

  Return the next patient who must be tended by a doctor in the area provided as an argument.

- **finalize_treatment <ARGUMENT1>**

The program must discharge the patient that is passed as an argument

- **get_historical <ARGUMENT1>**

  The system must provide a patient's historical with all the information of each income. If the argument corresponds to a patient who does not exist in the database, display a message with not found user.

- **search_patient <ARGUMENT>**

  Search in the whole list of patients the patient that is passed as the argument only if the patient has not been discharged. Show the last input data of the historical.

- **change_priority <ARGUMENT1> <ARGUMENT2>**

  Change the status (Argument2) for any patient (Argument1). Argument2 corresponds to the new priority (CRITICAL, MODERATE, GENERAL).

- **patients_by_day <ARGUMENT1>**

  It shows a list with all the patients that have entered in the system on the date that is passed as an argument.

- **patients_by_area <ARGUMENT1>**

  List all the patients inside an area provided as argument. It will select the patients that are not been discharged in the system.

- **number_patients_area <ARGUMENT1>**

  It returns the number of patients that are in the provided area

- **patients_priority_area <ARGUMENT1> <ARGUMENT2>**

  List all the patients that are enrolled in the area provided as argument 1 and have the priority provided as argument 2

Input data

The program should read a text file name with the National Identity Number of the patient. For example for an identity number 4324324CY, the text file should be named as 4324324CY.txt.

Each file should contain the following information separated by an enter character by each user:

Name
Priority
Area