

AEDD- EXAMEN FINAL - 7/10/2020

Ejercicio de Práctica en Papel -

Un hospital mantiene la información de sus 875 pacientes en un arreglo de registros que contiene la siguiente información: *nombre y apellido* (tipo string), *dni* (tipo entero), *edad* (tipo entero), *obra social* (tipo string) y *fecha de ingreso al sistema* (tipo fecha). El código de paciente es un valor entero que varía en el rango 0-874.

Al mismo tiempo, mantiene la información de sus 350 médicos en un arreglo de registros que contiene la siguiente información: *nombre y apellido* (tipo string), *especialidad* (tipo carácter según la siguiente codificación G = generalista, T = traumatólogo, O = oculista, P = pediatra), *cantidad de pacientes a cargo* (tipo entero) y *activo* (tipo bool). El código de médico es un valor entero que varía en el rango 0-349.

Cada vez que se produce el ingreso de un paciente, el hospital actualiza una lista de internados a fin de asociar un paciente con un doctor por medio de sus códigos internos. Además, el ingreso de un paciente como internado involucra su ubicación en una *habitación* y *piso* (valores enteros).

a) Asumiendo que en un momento dado no pueden existir más de 500 pacientes internados en el hospital, defina los tipos de datos y las estructuras necesarias para representar los pacientes, los médicos y los internados.

b) Defina la función *internarPaciente* que recibe un código de paciente y una especialidad y asigna el primer médico disponible (es decir, que se encuentre *activo*) a la atención del paciente. Como resultado, si existe médico disponible, la lista de internaciones y la cantidad de pacientes a cargo del médico son actualizadas. La habitación y el piso en los cuales se interna el paciente deben solicitarse por teclado.

c) Defina la función *mayoresDeEdad* que recibe la lista de pacientes y devuelve la cantidad de pacientes mayores de edad (edad mayor o igual a 18).

d) Defina la función *existeInternacionGeneralista* que recibe la lista de internaciones y devuelve *verdadero* si existe algún paciente internado por un médico generalista. En caso que no exista ningún paciente, la función devuelve *falso*.

e) Defina la función *ordenarPorEdad* que devuelve una copia del listado de pacientes ordenada según su edad.

f) Defina la función *pacienteMasJoven* que imprime en pantalla el nombre y apellido del paciente (no necesariamente internado) que tiene menor edad.

g) Defina la función *dejarInactivo* que recibe el código de un médico y actualiza el listado de todos sus pacientes internados asignando estos pacientes de forma equitativa a todos los médicos activos que corresponden a su misma especialidad. Como resultado, el médico indicado quedará inactivo y sin pacientes a cargo, el listado de internaciones será actualizado al igual que los médicos que reemplazan al médico saliente.

Importante: Para la resolución del problema el alumno puede codificar todas las funciones que considere necesarias. El puntaje final obtenido tendrá en cuenta la eficiencia de la estrategia de resolución elegida.