

## **AEDD- EXAMEN PARCIAL 2 - 2/12/2020**

### ***Ejercicio de Práctica en Papel - Comisión C***

Con el objetivo de estudiar el avance de la enfermedad del SARAMPION, se ha realizado un estudio sobre  $P=1500$  personas pertenecientes a la ciudad de Santa Fe. Para esto, se posee un vector *santafe* de  $P$  elementos que mantiene almacenada la información personal de cada uno de los ciudadanos bajo estudio. Para cada ciudadano se almacena: *edad* (valor entero), *barrio* (cadena de caracteres) y *contactos* (valor entero que indica la cantidad de personas con la que la persona convive y/o tiene contacto estrecho).

Por otra parte, se dispone de una lista de internaciones denominada *listado* (puede almacenarse en un vector). Por cada internación se registra: *id\_paciente* (valor entero que corresponde al subíndice  $i$  del vector *santafe* donde se encuentran los datos personales del paciente internado), *dias* (valor entero que indica la cantidad de días de internación) y un valor booleano *sarampionPositivo* (true si la persona dió positivo al test de SARAMPION, false en caso contrario). Como máximo, existirán  $TF=50$  internaciones en *listado*. En principio, se considera que *listado* se encuentra vacío (Tamaño Virtual=0).

Se solicita:

1. Definir todas las estructuras de datos que considere necesarias para dar respuesta al problema enunciado.
2. Implementar la función *cargarInternación()* que agregue una nueva *internacion* a *listado*. Los datos de la internación deben solicitarse por teclado. Como resultado de la función, *listado* debe quedar actualizado.
3. Implementar la función *InternadoNorte()* que determina si se ha internado alguna persona de barrio Norte (*barrio* = "Norte"). En caso afirmativo, se muestra por pantalla los datos personales de la primera persona que se ha encontrado que cumple con esta condición. En caso contrario, se muestra el mensaje "No existe".
4. Implementar la función *existeInternadoPlus()* que determina si existe alguna persona que ha sido internada más de una vez. En caso afirmativo, devuelve true. En caso contrario, devuelve false.
5. Implementar la función *sinContactos()* que retorne la cantidad de personas que forman parte del estudio y no tienen contactos estrechos (*contactos* = 0).

Para la resolución del problema el alumno puede codificar todas las funciones que considere necesarias. No es necesario codificar el proceso de carga de datos.