

## Tipo B

word.  
map

Queremos gestionar la sala de espera del servicio de urgencias de un hospital muy especial, donde los pacientes suelen mejorar solo por esperar.

La implementación del sistema se realizará mediante un TAD urgencias con las siguientes operaciones:

- nuevo\_paciente(paciente, gravedad): registra a un nuevo paciente (un string), que ya ha sido preevaluado por un médico que le ha asignado cierta gravedad (un int): leve (1), media (2) o grave (3). El paciente pasa a la sala de espera. Si el paciente ya estaba registrado, lanzará una excepción domain\_error con mensaje Paciente repetido. Si la gravedad dada no es un número entre 1 y 3, se lanzará una excepción domain\_error con el mensaje Gravedad incorrecta.
- gravedad\_actual(paciente): devuelve el entero que representa la gravedad actual de paciente. Si el paciente no está en la sala de espera, se lanzará una excepción domain\_error con mensaje Paciente inexistente.
- siguiente(): devuelve el nombre del paciente al que le toca ser atendido, que abandona la sala de espera. Los pacientes se atienden teniendo en cuenta su gravedad: primero los graves, luego los de gravedad media y por último los leves. Dentro de la misma gravedad se tiene en cuenta el orden de llegada (el primero que llega es el más prioritario), o el que provoca la operación mejora de cambio de gravedad, explicada a continuación. Si no hay pacientes se lanzará una excepción domain\_error con mensaje No hay pacientes.
- mejora(paciente): registra el hecho de que un paciente mejora estando en la sala de espera. Si estaba grave pasa a gravedad media y si tenía gravedad media pasa a leve. Para el orden de atención se coloca como el más prioritario de los que tienen la nueva gravedad. Si el paciente estaba leve, entonces se recupera y abandona las urgencias. Si el paciente no existe, se lanzará una excepción domain\_error con mensaje Paciente inexistente.
- recuperados(): devuelve un tipo de datos lineal ordenado alfabéticamente (y sin repeticiones) con los pacientes que han pasado alguna vez por el servicio de urgencias y se han recuperado del todo mientras esperaban.

### Requisitos de implementación.

La implementación de las operaciones debe ser lo más eficiente posible. Por tanto, debes elegir una representación adecuada para el TAD, implementar las operaciones y justificar la complejidad resultante.

Los métodos del TAD no deben mostrar nada por pantalla. El manejo de la entrada y salida de datos se realizará en funciones externas al TAD.

### Entrada

La entrada consta de una serie de casos de prueba. Cada caso está formado por una serie de líneas, en las que se muestran las operaciones a llevar a cabo, una por cada línea: el nombre de la operación seguido de sus argumentos. Los nombres de los pacientes serán cadenas de caracteres sin espacios. La palabra FIN en una línea indica el final de cada caso.

### Salida

Las operaciones nuevo\_paciente y mejora no producen salida, salvo en caso de error. Con respecto a las restantes:

- Tras llamar a gravedad\_actual debe imprimirse una línea con el texto La gravedad de XXX es N, siendo XXX el paciente consultado y N su gravedad (1, 2 o 3).
- Tras llamar a recuperados debe imprimirse una línea con el texto Lista de recuperados: seguida por los nombres de los pacientes recuperados, separados por espacios.

- Tras llamar a `siguiente` debe imprimirse una línea con el texto `Siguiente paciente: XXX`, donde `XXX` es el paciente al que le toca ser atendido.

Cada caso termina con una línea con tres guiones (`---`). Si una operación produce un error, entonces se escribirá una línea con el mensaje `ERROR:`, seguido del error que devuelve la operación, y no se escribirá nada más para esa operación.

### Entrada de ejemplo

```
nuevo_paciente Luis 2
nuevo_paciente Ana 3
nuevo_paciente Eva 3
nuevo_paciente Ivan 2
mejora Ana
siguiente
siguiente
siguiente
gravedad_actual Ivan
mejora Ivan
recuperados
mejora Ivan
nuevo_paciente Luz 1
mejora Luz
recuperados
FIN
nuevo_paciente Pedro 1
nuevo_paciente Pedro 2
mejora Pedro
recuperados
nuevo_paciente Pedro 3
mejora Pedro
gravedad_actual Pedro
siguiente
gravedad_actual Pedro
nuevo_paciente Pedro 4
siguiente
FIN
```

### Salida de ejemplo

```
Siguiente paciente: Eva
Siguiente paciente: Ana
Siguiente paciente: Luis
La gravedad de Ivan es 2
Lista de recuperados:
Lista de recuperados: Ivan Luz
---
ERROR: Paciente repetido
Lista de recuperados: Pedro
La gravedad de Pedro es 2
Siguiente paciente: Pedro
ERROR: Paciente inexistente
ERROR: Gravedad incorrecta
ERROR: No hay pacientes
---
```

**Autor:** Isabel Pita