LENGUAJES DE PROGRAMACIÓN II (ITIG)

Práctica 2

Estructuras de datos en C (de 10 a 12 horas)

APARTADO A

Dadas las siguientes estructuras:

```
typedef struct Pac
{
    int num_SS;
    int precio;
    struct Pac *sig;
} Paciente;

typedef struct
{
    Paciente *cap;
    Paciente *cua;
} Cola;
```

Escribir las siguientes funciones:

1. Una función que inicialice una cola poniendo sus punteros a NULL.
2. Una función a la que se le pasa como parámetros una cola, un número de seguridad social y un precio e inserta un paciente con esos datos en la cola.
3. Una función que elimina el paciente más antiguo de la cola y devuelve sus datos.
4. Una función a la que se le pasa una cola y un número de SS y elimina dicho paciente de la cola.
5. Una función que vacíe el contenido de una cola.
6. Una función a la que se le pase una cola y devuelva la suma del campo precio de todos los pacientes.
7. Una función que recorra una cola mostrando los datos de todos los pacientes que hay en ella.

Escribir un programa en C que permita manejar las anteriores cinco funciones sobre una cola.

APARTADO B.

Reutilizando el código del apartado anterior, queremos realizar un programa para gestión diaria de pacientes en una consulta médica. En dicha consulta hay una enfermera que toma los datos y dos médicos que pasan consulta. Los pacientes son atendidos por el médico que esté libre y en estricto orden de llegada. Una vez reconocido el paciente, el médico indica a la enfermera la cantidad que debe cobrarle al enfermo. Los datos de los enfermos atendidos se almacenan para luego poder hacer un recuento de las ganancias de cada médico.

Todo esto se gestionará mediante tres colas de pacientes. La primera almacena la lista de pacientes en espera, y las otras dos los pacientes que ya han sido atendidos, una cola para cada médico. Cada vez que llega un paciente, éste se inserta en la cola de la primera lista. Cada vez que se atiende un paciente, se elimina su nodo de la cabeza de la primera lista y se apunta en alguna variable su número de SS hasta que el paciente salga de la consulta. Entonces se inserta el paciente en la cola del médico correspondiente con el precio que ha pagado por la visita.

El programa debe contar con las siguientes opciones:

1. *Nuevo paciente*. Inserta un paciente en espera.
2. *Atender paciente por el médico 1*. Saca un el paciente más antiguo de la lista de espera y anota su número de SS en alguna variable.
3. *Atender paciente por el médico 2*. Saca el paciente más antiguo de la lista de espera y guarda su número de SS en una variable.
4. *Cobrar al paciente del médico 1*. Pone el precio e inserta el nodo en la lista del médico 1.
5. *Cobrar al paciente del médico 2*. Lo mismo que la opción anterior, pero en la cola del médico 2.
6. *Ganancias*. Esta opción recorre las colas de pacientes atendidos contando las ganancias de cada doctor, y sobre el total de ambos calcula el 10%, que es el sueldo de la enfermera. Después elimina todos los nodos de estas listas.
7. *Eliminar paciente en espera*. Algunas veces los pacientes fallecen en la sala de espera, o simplemente se cansan y se van. En ese caso se debe eliminar su nodo de la lista de espera.
8. *Eliminar paciente de la cola de atendidos*. ¡Si no queda satisfecho le devolvemos su dinero! Se elimina el paciente para no contabilizarlo luego al calcular las ganancias.
9. *Salir del programa*. Se manda a todos los pacientes a casa y se vacía la lista 1. A continuación se espera a que salgan los pacientes que están siendo atendidos para poderles cobrar, y luego se calculan las ganancias tal y como indica la opción 6.

Además, el programa en todo momento mostrará el estado de los médicos (si están ocupados o están libres).

Nota aclaratoria: En la cola de pacientes en espera carece de utilidad el campo precio de las estructuras *Paciente*, pues este valor sólo se sabe después de que el enfermo haya sido atendido. No obstante, con el objeto de que la realización de la práctica sea más sencilla, se ha preferido que la estructura de datos sea la misma para las tres colas.