UNIVERSIDAD DE MÁLAGA
E.T.S.I. TELECOMUNICACIÓN

Simón dice ...
Laboratorio de Programación 2
(1º de Ingeniería de Telecomunicación)

Simón dice ...

Se quiere realizar el juego de *Simón dice ...* En este juego el ordenador genera una secuencia de colores que el jugador debe repetir. Si el jugador acierta a repetir la secuencia completa, en el siguiente turno el ordenador añade un nuevo color al final de la secuencia y la repite entera. Si el jugador no es capaz de repetir la secuencia completa, pierde.

La representación de la secuencia

Son varias las operaciones que hay que realizar sobre la secuencia de colores, como añadir un elemento por el final y consultar el valor de cada uno de sus elementos para escribirlos en pantalla y compararlos con los valores introducidos por el usuario.

La secuencia se puede, por tanto, representar con una estructura lineal en la que tengamos acceso directo a todos los elementos. La *lista posicional* cumple esos requisitos.

Su interfaz es el siguiente:

```
class ListaPos
{
    public:
        enum TColor {rojo, amarillo, azul, verde};

        typedef TBase TColor;

    private:
        struct Nodo* PNodo;

        struct Nodo
        {
            TBase cl;
            PNodo sig;
            PNodo ant;
        };

        PNodo primero;
        PNodo ultimo;

    public:

        ListaPos();
        // Crea un lista posicional vacia
        ~ListaPos();
        // libera los recursos usados por el objeto

        void Insertar(const unsigned int pos, const TBase el);
        // agrega un nuevo elemento a la lista en la posicion indicada
```

```
        // por pos. El rango valido es 1<= pos <= Longitud + 1.
        void Eliminar(const unsigned int pos);
        // borra de la lista el elemento que esta en la posicion indicada
        // por pos. El rango valido es 1<= pos <= Longitud.
        TBase Consultar(const unsigned int pos);
        // devuelve el elemento de la lista que esta en la posicion indicada
        // por pos. El rango valido es 1<= pos <= Longitud.

        unsigned int Longitud() const;
        // devuelve el numero de elementos que hay en la lista
        bool Esta_Vacia() const;
        bool Esta_Llena() const;
};
```

Como se ve en el interfaz, se ha escogido una representación dinámica para implementar la lista. En este caso, los elementos que se van a insertar en la cola son de un tipo enumerado `TColor`.

El bucle de control del juego

El cuerpo principal del juego consistirá en un bucle en el que en cada iteración se hace una jugada. En primer lugar, el ordenador genera un nuevo elemento de la secuencia. Después, muestra toda la secuencia por pantalla. Una vez mostrada, el ordenador irá leyendo la respuesta del jugador hasta que complete la secuencia o se equivoque.

El uso de números aleatorios nos permitirá evitar que siempre se obtenga la misma secuencia. El siguiente código genera una secuencia aleatoria de colores.

```
#include <cstdlib> // para srand y rand
#include <ctime>   // para clock

...
srand(clock()); // inicia la serie en funcion de la hora actual

while (!fin)
{
   nuevo = TColor( rand() % MAX_COLOR); // genera un nuevo color
}
...
```