

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**  
**FUNDAMENTOS DE PROGRAMACIÓN**  
**PRIMERA EVALUACIÓN**  
**I TÉRMINO 2012-2013**

*Nombre:* \_\_\_\_\_ *Matrícula:* \_\_\_\_\_

**TEMA 1 (25 puntos)**

- La función **CESPED** dibuja un pequeño césped de 9 puntas.  
 - La función **TALLO** dibuja el tallo completo de una flor.  
 -La función **CUADRADO** recibe como parámetro el diámetro del cuadrado que la tortuga debe dibujar.

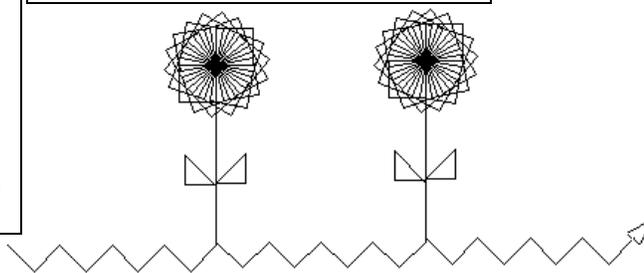
1. Complete el procedimiento llamado **FLORCUADRADA :LADO** que dibuja cuadrados consecutivos. La función cada vez que empieza a dibujar un cuadrado nuevo lo hace desviándose un cierto ángulo respecto al anterior, para llegar a formar una figura con aspecto de flor como la de la ilustración siguiente. Para este dibujo en particular dibuja 18 cuadrados. Use la función **CUADRADO**.  
 2. Implemente el código de la función **JARDIN** que dibuje la siguiente figura en Logo. **USE** las funciones definidas anteriormente.

```
TO CESPED
rt 45
REPEAT 4[RIGHT 90 FORWARD 25 LEFT 90 FORWARD 25]
END
```

```
TO TALLO
FD 40 LT 45 FD 30 lt 135 fd 20 lt 90 fd 20 lt 45
fd 30 rt 135 fd 20 rt 90 fd 20 rt 90 fd 80
FND
```

```
TO CUADRADO :LADO
REPEAT 4 [ FORWARD LADO RIGHT 90 ]
END
```

```
TO FLORCUADRADA:LADO
END
```



**TEMA 2 (30 puntos)**

Una ferretería tiene una banda sin fin por donde se desplazan tornillos asimétricos y al final de la banda existe un recolector con algunas perforaciones en donde se clasifican los tornillos dependiendo del tamaño.

La banda es un arreglo de 20 celdas que contiene números entre 1 y 7, que representan los diferentes tamaños de tornillos. El recolector es un arreglo de 7 celdas que contiene para cada tipo de tornillo, cuántos existen de cada uno. Ud. Debe implementar en Scilab:

1. La función [AT]= **generarArreglo**( ); que devuelve el arreglo que representa la banda (AT) lleno con números aleatorios entre 1 y 7.
2. La función [AR]= **clasificarTornillos**(AT); que recibe el arreglo de tornillos y devuelve el arreglo recolector lleno con las cantidades existentes de cada tipo de tornillo.

Por ejemplo, con este arreglo de tornillos:

1	1	3	4	5	4	6	7	3	2	1	3	5	5	2	2	4	4	7	7
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

La salida de la función **clasificarTornillos** es la siguiente:

3	3	3	4	3	1	3
1	2	3	4	5	6	7

### TEMA 3 (45 puntos)

Una compañía desea implementar la versión electrónica del juego de naipes “Quita Montón”. El juego consiste en obtener cartas de un mazo de 52 cartas (13 cartas por cada palo) y se rige de acuerdo a las siguientes reglas:

1. Al iniciar el juego se baraja el naipe y se obtiene una carta del mazo. Esta carta será la base del montón.
2. El primer turno siempre lo tiene el usuario y el siguiente turno la computadora.
  - a. El usuario tiene opción de sacar una carta o pasar su turno.
  - b. La computadora siempre saca carta.
3. Al sacar una carta del mazo se la coloca sobre el montón.
4. Si la carta que se está colocando es de igual valor a la que se encuentra sobre el montón, el jugador que la coloca se *lleva todas las cartas (Llevarse todas las cartas significa acumular un punto por cada carta del montón y dejar el montón con 0 cartas)*. A continuación la computadora obtiene una carta del mazo, la misma que será la nueva base del montón.
5. El juego continúa y el turno lo tiene el siguiente jugador.
6. El juego se repite a partir del paso 3 y termina cuando no hay más cartas en el mazo.
7. Gana el juego el jugador que más puntos tenga.
8. Al finalizar el juego, el programa debe mostrar que jugador ganó y el número de puntos acumulados por dicho jugador.

#### Usted debe implementar:

1. La función **[topeMonton, nCartasMonton, puntos] = colocarCarta(cartas, topeMonton, nCartasMonton)**, que recibe: La carta a colocar, la carta que se encuentra en el tope del montón y cuantas cartas se encuentran en el montón. Retorna: la carta que se encuentra en el nuevo montón, el número de cartas existentes en el nuevo montón, la cantidad de puntos resultantes de la jugada.

#### 2. Un programa que implemente el juego “Quita montón”

- Utilizando la función implementada anteriormente y las disponibles en **Nota\***.
- En cada turno del jugador, se deberá mostrar por pantalla:
  - La carta que se encuentra sobre el montón.
  - La cantidad de cartas que se encuentran actualmente en el montón.
  - La cantidad de cartas en el mazo.
  - Los puntos de los dos jugadores.

**Nota:** Asuma que posee las siguientes funciones, si usted desea implementarlas cada una tendrá un valor de 5 puntos extra:

- **[carta] = obtenerCarta(mazo)**, que recibe el mazo y remueve la carta que se encuentra sobre el mazo. En el caso de que el mazo esté vacío, retorna 0.
- **[mazo] = barajar(mazo)**, que recibe un mazo de cartas y lo desordena.