

# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

PRIMERA EVALUACIÓN  
I TÉRMINO 2010 – 2011

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

## TEMA 1 (20 puntos)

Implemente la función **long rusa(long n1, long n2)** que retorna el producto de n1 por n2 utilizando el método de multiplicación rusa. Para hallar el producto de n1 por n2, hay que sumar el valor de n1 cuando su respectivo **n2** es impar. Luego **n1** se duplica a la vez que **n2** se reduce a su mitad entera. El algoritmo continúa hasta que **n2** se convierta en 1.

**Por ejemplo:** Para multiplicar 35 por 23 se tiene la siguiente prueba de escritorio.

<u>n1</u>	<u>n2</u>	<u>suma</u>
35	23	35
70	11	105
140	5	245
280	2	245 → n2 es par, no se suma el 280
560	1	<b>805 → el resultado es 805</b>

**Nota:** Asuma que n1 y n2 son mayores a cero.

## TEMA 2 (40 puntos)

Implemente la función **bool cedulaValida ( int ced[ ] )**; donde el arreglo **ced** es de tamaño 10 y contiene en cada posición un dígito de la cédula. Esta función retorna *true* si la cédula es válida sino retorna *false*.

Para que una cédula sea válida debe cumplir TODAS las condiciones siguientes:

- Las 2 primeras posiciones corresponden a la provincia donde fue expedida. Estos dos primeros dígitos forman un número que no será mayor que 24 ni menor a 1.
- El tercer dígito es menor que 6.
- El último dígito (verificador) debe ser igual a **result** (número obtenido aplicando la fórmula del registro civil).

### Fórmula del registro civil

1. Usa un arreglo de coeficientes **A**.

**A**

2	1	2	1	2	1	2	1	2	0
---	---	---	---	---	---	---	---	---	---

2. Multiplica las posiciones del arreglo **A**, por las correspondientes posiciones del arreglo **ced**. Si el resultado de alguna multiplicación es *mayor que 9* se suman sus dígitos.

**A**

2	1	2	1	2	1	2	1	2	0
---	---	---	---	---	---	---	---	---	---

**ced**

1	7	1	0	0	3	4	0	6	5
---	---	---	---	---	---	---	---	---	---

**A\*ced**

2	7	2	0	0	3	8	0	12	0
								3	

Si el resultado de la multiplicación es *mayor que 9* se suman sus dígitos.

3. Se suman los resultados de las multiplicaciones parciales. A este resultado le llamaremos **sum**. Para el ejemplo:  $2+7+2+0+0+3+8+0+3+0=25$ .
4. Se obtiene el residuo de dividir **sum** para 10. A este valor le llamaremos **residuo**. Para el ejemplo, **residuo** es 5.
5. Finalmente, se obtiene **result**. Si **residuo** es 0, **result** es 0; caso contrario **result** es  $10 - \text{residuo}$ .

### **TEMA 3 (40 puntos)**

Una compañía desea implementar la versión electrónica del JUEGO DE LA OCA para 2 jugadores. El juego de la oca se realiza utilizando un tablero con 63 posiciones y 2 dados. Este juego se realiza por turnos. Inicia el jugador 1.

Las posiciones del tablero están numeradas del 1 al 63 y en algunas de ellas hay un dibujo.

**Oca:** en las posiciones múltiplo de 7 aparece el dibujo de una oca. Cuando se cae en una oca, se debe avanzar hasta la siguiente posición en la que hay una oca.

**Laberinto:** si el jugador cae en la posición 43, está obligado a retroceder a la posición 30 donde está dibujado un laberinto.

#### **En cada turno se debe:**

- Mostrar por pantalla el mensaje “**De oca a oca y tiro porque me toca**” para mantener la tradición del juego original.
- Lanzar los dados y avanzar sobre el tablero el número de posiciones que sea igual a la suma de los dados. Esto se realiza para el jugador que tiene el turno.
- Mostrar por pantalla la posición actual del jugador.
- Una vez que el jugador ha caído en una posición, realizar lo indicado de acuerdo al dibujo del tablero, su turno termina y pasa el turno al siguiente jugador.

La partida continúa con los jugadores avanzando en el tablero, como se ha descrito, hasta que un jugador llega a la última posición o al lanzar los dados su ubicación sobrepasa la última posición; en esta situación se dice que ese jugador llegó al “Jardín de la Oca ” con lo cual ese jugador gana la partida. **Al finalizar se muestra el ganador y para cada jugador: las posiciones en el tablero y el número de lanzamientos realizados.**

#### **Ud. deberá implementar:**

- a) La función **int juega (int posJugador)**; recibe la posición actual y retorna la nueva posición del jugador según las reglas.
- b) El procedimiento **posicionJugador(int posJugador)**; el cual muestra en pantalla un mensaje que indica la posición en la que se encuentra el jugador. Si la posición en el tablero tiene un dibujo (oca, laberinto) el mensaje también lo indicará.
- c) La función **int sigteOca(int posicion)**; la cual retorna el número de la posición de la siguiente oca del tablero. Si no hay másocas retorna -1.
- d) El **programa principal** que simule el juego de la oca. **Use las funciones y procedimientos anteriores.**