

1. Hacer un programa que simule el lanzamiento de un dado 10.000 veces. Almacenar en un vector de 6 posiciones el número de veces que salió cada tirada, escribiendo a continuación el número de veces que salió cada cara del dado.
2. El DNI. La letra final del DNI se usa para comprobar la corrección de un número de DNI. La regla de cálculo de la letra es sencilla: se halla el resto de dividir el DNI entre 23, y se asigna una letra en función del resto según la tabla de abajo. Hacer un programa que solicite un número de DNI por teclado e imprima en pantalla su letra. Pone un ejemplo: se introduce 33678223 y el programa indica "Su DNI completo es 33678223-J".

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

3. Programa que genere N números aleatorios no repetidos en el rango 0 a 100, siendo  $N < 100$ . Obtener la media de los números generados.
4. La **Lotería Primitiva** es un juego de azar regulado por la Organización Nacional de Loterías y Apuestas del Estado (LAE) que consiste en elegir 6 números diferentes entre 1 y 49. Para acertar la Combinación Ganadora en el sorteo de **Lotería Primitiva** correspondiente, se extraen 6 bolas de un bombo. Además, se extrae una bola extra como número complementario, y otra bola de un bombo aparte, entre el 0 y el 9, que hace de número de «reintegro».

Hacer un programa que genere 5 combinaciones de la lotería primitiva. Tiene que visualizar:

Combinación 1: 4, 2, 7, 23, 45, 34

Complementario:13

Reintegro: 5

5. Implementar un método que reciba el valor de una moneda, una lista con el valor de las monedas y otra con la cuantía de cada moneda simulando los contenedores de una máquina de cambio. El algoritmo deberá dar el cambio de la moneda con el menor número de monedas.

En la clase Random con C# tenemos tres métodos que nos generan un número aleatorio, pero primero de todo tenemos que crear una instancia de la clase Random:

```
Random r = new Random  
// Creamos el objeto Random, que tiene un constructor por defecto  
// O para los más exigentes, añadimos un plus de  
// aleatoriedad, algo así:  
Random r = new Random(DateTime.Now.Millisecond);
```

Y una vez tenemos el objeto, utilizaremos uno de esos tres métodos.

1.- El método Next Tiene tres sobrecargas:

```
int aleatorio1 = r.Next();  
  
// Le imponemos un máximo  
int aleatorio2 = r.Next(100);  
  
// Le imponemos un rango, por ejemplo queremos  
// un número de 3 cifras  
int aleatorio3 = r.Next(100, 999);  
  
// Queremos un número entre el 3 y el 12,  
int aleatorio4 = r.Next(3, 12);
```

2.- El método NextDouble

```
// Devuelve un double entre 0 y 1  
int aleatorio5 = r.NextDouble();
```

3.- El método NextBytes

```
byte[] buffer = new byte[100];  
r.NextBytes(buffer);
```

Ejemplo:

```
//Inicializamos la clase Random  
Random r = new Random();  
  
//Creamos un array que va a contener cantidad  
//aleatoria de elementos menor que 50.  
int [] numeros = new int [r.Next(0,50)];  
  
//Recorremos el array y vamos asignando a cada  
//posición un número aleatorio  
for(int i=0; i<numeros.Length; i++)  
    numeros[i] = r.Next(0,50);
```