

## RELACIÓN DE PROBLEMAS 8

### ARRAYS I

1. Realizar un programa que solicite diez números enteros y cuando se termine de introducir el último los muestre en el orden inverso al que se introdujeron.
2. Realiza un programa que cargue un vector con de 10 números enteros. El programa debe imprimir los elementos del vector y después desplazarlos una posición, de tal forma que el último pase a la primera posición, el primero a la segunda, el segundo a la tercera, y así sucesivamente.

```
Introduce el número en la posición (1/10): 5
Introduce el número en la posición (2/10): 10
...
Introduce el número en la posición (10/10): 9
```

```
Vector original:          5 10 2 30 4 15 6 88 3 9
Vector después de desplazar: 9 5 10 2 30 4 15 6 88 3
```

NOTA: Sólo debe usarse un vector, no crear vectores auxiliares.

3. Realizar un programa que solicite por teclado números entre 1 y 1000 hasta que el usuario informe de que ya no desea introducir más números. Después debe mostrar:
  - La media de los números leídos.
  - Cuántos números terminan en cada uno de los dígitos (0 .. 9).
  - El dígito en el que más números terminan.
  - En qué dígitos no ha terminado ningún número.

Pista: crear un vector de contadores de 10 elementos.

4. Realizar un programa que solicite la fecha como tres datos numéricos (día, mes y año). Deberá mostrar la fecha en formato largo.

```
Introduce el día de la fecha: 15
Introduce el mes de la fecha: 3
Introduce el año de la fecha: 2009
La fecha en formato largo es 15 de Marzo de 2009
```

Si la fecha introducida es incorrecta se mostrará un mensaje de error: "Fecha incorrecta" y el programa terminará.

Crear la clase **Fecha** e incluir el método **pasarAformatoLargo** que devuelva una cadena con formato largo.

NOTAS:

- Para guardar los meses del año, utilizar un array de String.
- Utilizar el método **diasDeUnMes** que devuelve el número de días de un mes (Relación 4, ejercicio 4).
- Se deben tratar los errores con Excepciones.

5. Se desea realizar un programa que simule el reparto de cartas en un juego. El juego tiene 48 cartas, separadas en 4 palos (oros, copas, espadas y bastos) y numeradas del 1 al 12. Se desea realizar un programa que solicite cuantas cartas se van a repartir y cree un vector de cartas para el jugador 1 y otro para el jugador 2. Se deben cargar aleatoriamente las cartas de cada jugador. Las cartas no se repiten.

¿Cuántas cartas reparto?: 5

Cartas del jugador 1:

7 de oros, 8 de espadas, 5 de copas, 4 de copas, 3 de bastos

Cartas del jugador 2:

5 de espadas, 7 de copas, 9 de copas, 10 de bastos, 3 de oros

6. Realizar un programa que solicite una frase e informe de cuantas veces aparecen cada una de las letras del abecedario en la frase (sólo se mostrarán las letras que aparezcan al menos una vez).

Pista: Crear un array de contadores (entero) con tantos elementos como letras del abecedario y un String con las letras del abecedario.

7. Realiza un programa que solicite el número de alumnos de una clase y pida por pantalla para cada uno de los alumnos su nombre y su nota. Después se presentará un menú que permitirá realizar las siguientes operaciones:

1. Mostrar el nombre y la nota de los alumnos aprobados.
2. Mostrar el nombre y la nota los alumnos suspensos.
3. Mostrar la nota media de la clase.
4. Mostrar los alumnos que tienen más de una nota X (se solicitará X por teclado).
5. Mostrar el alumno con la nota máxima y mínima.
6. Mostrar un listado ordenado de forma creciente por la nota.
7. Salir.

8. Realizar un programa que presente el siguiente menú:

1. **Guardar un refrán:** Se solicitará la frase con el refrán y se guardará. Si el refrán ya estaba, se mostrará un mensaje de error y no se guardará.
2. **Buscar un refrán que contenga una palabra:** Se solicitará una palabra y se buscará el primer refrán que contenga esa palabra. Si no se encuentra se mostrará el mensaje de error
3. **Listar refranes:** Se mostrará por pantalla un listado de todos los refranes guardados ordenados alfabéticamente.
4. **Salir.**

**NOTAS:**

- Se guardarán un máximo de 50 refranes, si se intenta superar este máximo se producirá un error.
- Debe crearse una clase **Refrán**, otra clase **Refranes** que contenga el vector de Refrán y una clase **Principal**.