

## Ejercicios propuestos de las clases String, Arrays y tipos de datos Básicos

1. Mostrar por pantalla el abecedario completo en Mayúsculas.
2. Introducir un texto, pasarlo a vector de caracteres e indicar si representa un número.
3. Introducir un texto, volvedlo a mostrar en mayúsculas.
4. Indica el número de aces, y de vocales que posee el String : "Mañana es sabado sabadete y voy a irme a tomar unas copillas por los barrios bajos de Logroño".
5. Introducir un texto, e indicar si está la 'W' (en mayúscula o minúscula), y la posición que ocupa la primera vez.
6. Dado el texto del ejercicio 4, indicar si está la 'T' (en mayúscula o minúscula), y las posiciones que ocupan (es necesario utilizar un array).
7. Introducir un texto, e indicar si está la 'p', comenzando por la última posición.
8. Introduce una cadena, devuelve la invertida.
9. Introducir un texto, comprueba si todo él está compuesto por letras –que sea una palabra- (mayúsculas o minúsculas)
10. Introducir un texto, muéstrala de nuevo sin espacios en blanco.
11. Comprobad si un número o palabra es palíndromo/a (Se lee de la misma manera de izquierda a derecha, que de derecha a izquierda). Por ejemplo, "A cavar a Caravaca" o "1221".
12. Introducir un texto, y mostrar la suma de los caracteres correspondientes a dígitos pertenecientes a la secuencia almacenada en un String. Por ejemplo, si la cadena es ABC12m4XYZ, entonces debe devolver el valor numérico entero 7.
13. Partiendo de la cadena String "En mi proxima vida, creere en la reencarnacion" declarada e inicializada como variable primitiva, mostrar por si se encuentra la subcadena "creere"

## Ejercicios Refuerzo Tema 4.

1. (NumMayorPrimo.java) Leer 10 enteros, almacenarlos en un vector y determinar en qué posición del vector está el mayor número primo leído.
2. (NumPrimos2.java) Almacenar en un vector de 10 posiciones los 10 números primos comprendidos entre 100 y 300. Luego mostrarlos en pantalla.
3. (NumTerminados4.java) Leer 10 números enteros, almacenarlos en un vector y determinar en qué posiciones se encuentran los números terminados en 4.
4. (SumaDigitos.java) Leer 10 números enteros, almacenarlos en un vector y determinar en qué posición está el número cuya suma de dígitos sea la mayor.
5. (NumConMasDigitos.java) Leer 10 números enteros, almacenarlos en un vector y determinar en qué posición está el número con mas dígitos.
6. (Factorial.java) Leer 10 números enteros, almacenarlos en un vector y calcularle el factorial a cada uno de los números leídos almacenándolos en otro vector.
7. (NumTerminados15.java) Leer 10 números enteros, almacenarlos en un vector y determinar cuántos números de los almacenados en dicho vector terminan en 15.
8. (Encuesta.java) Con base en una encuesta realizada a N estudiantes ( $N \leq 50$ ) – donde N se introduce por teclado- en una Universidad donde se solicitó la siguiente información: CÉDULA, SEXO, SUELDO, TRABAJA. Donde:
  1. CEDULA (Es un número entero)
  2. SEXO (1 - Masculino 2 – Femenino)
  3. TRABAJA (1 - Si trabaja 2 - No trabaja)
  4. SUELDO (Es un valor entero)

Escriba el algoritmo que permita leer y almacenar en una matriz los datos solicitados, luego, calcular e imprimir:

1. Porcentaje de hombres en la Universidad
  2. Porcentaje de mujeres en la Universidad
  3. Porcentaje de hombres que trabajan y sueldo promedio
  4. Porcentaje de mujeres que trabajan y sueldo promedio
9. (`Vehiculos.java`) Escriba un algoritmo que a partir de leer y almacenar en vectores el valor comercial de 20 vehículos y el tipo (familiar (1), público (2), de carga (3)), calcule y muestre su valor final con base en: (guardar una matriz de 20 filas, 3 columnas (1 para el tipo, otra para el valor comercial)):
1. Para vehículos de valor superior a 35.000€, cobrar IVA de 30%, para los demás el IVA es del 21%
  2. Para vehículos tipo 1, con valor hasta 15.000€, aplicar descuento equivalente al 50% del IVA cobrado
  3. Para vehículos tipo 2 y 3, con valor superior a 30.000€, aplicar sobrecosto por impuesto de rodamiento del 5%
  4. Para todos los vehículos, si el valor final es inferior a 20.000€, aplicar descuento adicional del 5% del valor comercial.
10. (`SalaSistemas.java`). El Departamento de Sistemas de la UCPR cuenta con cinco salas de sistemas, cada una de ellas con 20 ordenadores. Se requiere sistematizar el proceso de reserva o cancelación de turnos para dichas salas en el horario disponible (12 a 2 pm, únicamente), los demás horarios están reservados para clases.

Se debe crear una matriz bidimensional, con 5 filas correspondientes a las salas, 20 columnas correspondientes a cada uno de los ordenadores de cada sala (0- sin reservar, 1-reservado), y 2 turnos para cada ordenador de cada sala, el 0 representa fuera de rango 12-2pm, el 1 representa reservado por el cliente.

Construya un algoritmo que solicite al usuario la siguiente información (sala, equipo), tanto para asignar turno como para cancelarlo. Igualmente, el programa deberá permitir conocer el número de equipos disponibles (sin asignar) por sala, y el número total de turnos asignado (todas las salas). Tenga en cuenta que el algoritmo deberá funcionar permanentemente mostrando un menú de opciones así:

1. Asignar o cancelar turno
2. Cantidad de equipos disponibles por sala
3. Cantidad de turnos asignados en toda la UCPR
4. Salir