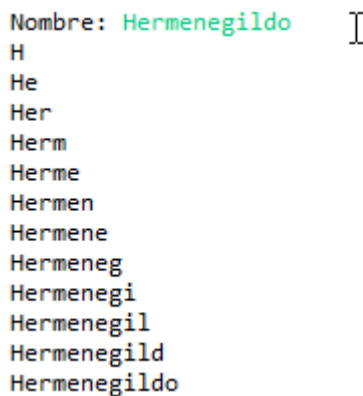


Ejercicios Java

Unidad 2: cadenas de caracteres

1. Programa que lea una cadena de teclado y muestre por pantalla el número de veces que aparece cada vocal en ella.
2. Programa que lea de teclado una cadena y la muestre invertida.
3. Escribe un programa que lea dos cadenas y muestre el número de veces que la segunda está contenida en la primera.
4. Escribe un programa que lea dos cadenas y muestre el número de veces que encuentra cada palabra de la primera dentro de la segunda.
5. Escribe un programa que lea líneas hasta que se introduzca en una de ellas la cadena "fin" sin importar si se hace en mayúsculas o en minúsculas o en cualquier combinación de ambas. Por cada línea leída, excepto la última, mostrará cuál es el primer carácter que no se repite.
6. Escribe un programa que lea nombres y construya un triángulo con sus caracteres tal y como se muestra en el ejemplo siguiente:



```
Nombre: Hermenegildo
H
He
Her
Herm
Herme
Hermen
Hermene
Hermeneg
Hermenegi
Hermenegil
Hermenegild
Hermenegildo
```

El programa finalizará cuando lea la palabra fin.

7. Escribe un programa que lea nombres y muestre el resultado de sustituir los caracteres centrales (todos menos el primero y el último) por asteriscos. Se ha de tener en cuenta que cada nombre puede estar precedido y/o seguido de un número arbitrario de espacios en blanco que se deben ignorar. Resolver cada sustitución sin usar sentencias repetitivas.
8. Escribe un método que acepte dos cadenas y un índice y retorne el resultado de insertar la segunda en la primera en la posición indicada por el índice.
9. Escribe un programa que lea secuencias de caracteres con el formato siguiente: *LetraNúmeroCarácter*. Por ejemplo, las siguientes son cadenas de este tipo: `M135t`, `b57X`, `n1335$` o `L91`. La entrada puede

constar de varias líneas y cada una puede contener cero o más secuencias separadas por espacios en blanco.

Cada secuencia se procesará para generar un valor numérico de la forma siguiente:

- Si la primera letra es mayúscula, el número que hay entre ella y el último carácter se multiplicará por el valor numérico de dicho carácter. En caso contrario, en lugar de la multiplicación se realizará una división.
- Si el último carácter es alfabético, se calculará la raíz cuadrada del resultado anterior. Si es un dígito, se calculará su logaritmo neperiano. En caso contrario se calculará el resultado de elevar el número de Euler a dicho resultado.

El programa finalizará, mostrando la suma de todos los valores numéricos obtenidos de cada secuencia, cuando se detecte en la entrada un EOF.

Se asume que no se cometen errores en la entrada de datos y todas las secuencias son válidas.

Ejemplo:

```
M135t
b57X n1335$
      L91
Resultado: 186642.28196718823
```

10. Escribe un método que acepte una cadena y retorne la suma de todos los dígitos que contenga.

Ejemplo: "la suma de 123 y 321 es igual a 444" \Rightarrow 24 (1+2+3+3+2+1+4+4+4)

Escribe un programa que lo ponga a prueba.

11. Escribe un programa que lea cadenas de caracteres por teclado y muestre para cada uno de sus caracteres el *code point* que le asigna Unicode. Los caracteres repetidos sólo se mostrarán una vez.

12. Escribe un método que acepte dos cadenas y retorne un valor que indique si la segunda es un anagrama de la primera.

Escribe un programa que lo ponga a prueba.

13. Escribe un método que acepte una cadena y retorne otra obtenida como resultado de intercambiar cada par de caracteres. Si el número de caracteres es impar, el último se queda como está.

Ejemplo: "Hola Mundo!" \Rightarrow "oHalM nuod!"

14. Escribe un programa que lea líneas de texto por teclado y muestre el resultado de eliminar todos los caracteres duplicados en cada una de ellas.

Ejemplo: "teorema del vago: lo intento, pero no lo hago" \Rightarrow "teorma dlvg:in,ph"

15. Escribe un programa que lea líneas de texto por teclado y por cada línea leída muestre la subcadena más larga sin caracteres duplicados.