

Actividad resuelta 6.1

Escribir un programa que muestre todos los caracteres Unicode junto a su code point, cuyo valor esté comprendido entre `\u0000` y `\uFFFF`.

Actividad resuelta 6.2

Introducir por teclado dos frases e indicar cuál de ellas es la más corta, es decir, la que contiene menos caracteres.

Actividad resuelta 6.3

Diseñar el juego «Acierta la contraseña». La mecánica del juego es la siguiente: el primer jugador introduce la contraseña; a continuación, el segundo jugador debe teclear palabras hasta que la acierte. Realizar dos versiones; en la primera se facilita el juego indicando si la palabra introducida es mayor o menor alfabéticamente que la contraseña. En la segunda, el programa mostrará la longitud de la contraseña y una cadena con los caracteres acertados en sus lugares respectivos y asteriscos en los no acertados.

Actividad resuelta 6.4

Diseñar una aplicación que pida al usuario que introduzca una frase por teclado e indique cuántos espacios en blanco tiene.

Solución

Actividad resuelta 6.5

Diseñar una función a la que se le pase una cadena de caracteres y la devuelva invertida. Un ejemplo, la cadena «Hola mundo» quedaría «odnum aloH».

Solución

Actividad resuelta 6.6

Escribir un programa que pida el nombre completo al usuario y lo muestre sin vocales (mayúsculas, minúsculas y acentuadas). Por ejemplo, "Álvaro Pérez" se mostrará «lvr Prz».

Solución

Actividad resuelta 6.7

Diseñar un programa que solicite al usuario una frase y una palabra. A continuación buscará cuántas veces aparece la palabra en la frase.

Solución

Actividad resuelta 6.8

Los habitantes de Javalandia tienen un idioma algo extraño; cuando hablan siempre comienzan sus frases con «Javalín, javalón», para después hacer una pausa más o menos larga (la pausa se representa mediante espacios en blanco o tabuladores) y a continuación expresan el mensaje. Existe un dialecto que no comienza sus frases con la muletilla anterior, pero siempre las terminan con un silencio, más o menos prolongado y la coletilla «javalén, len, len». Se pide diseñar un traductor que, en primer lugar, nos diga si la frase introducida está escrita en el idioma de Javalandia (en cualquiera de sus dialectos), y en caso afirmativo, nos muestre solo el mensaje sin muletillas.

Actividad resuelta 6.9

Introducir por teclado una frase palabra a palabra, y mostrar la frase completa separando las palabras introducidas con espacios en blanco. Terminar de leer la frase cuando alguna de las palabras introducidas sea la cadena «fin» escrita con cualquier combinación de mayúsculas y minúsculas. La cadena «fin» no aparecerá en la frase final.

Solución

Actividad resuelta 6.10

Realizar un programa que lea una frase del teclado y nos indique si es palíndroma, es decir, que la frase sea igual leyendo de izquierda a derecha, que de derecha a izquierda, sin tener en cuenta los espacios. Un ejemplo de frase palíndroma es: «Dábale arroz a la zorra el abad».

Las vocales con tilde hacen que los algoritmos consideren una frase palíndroma como si no lo fuese. Por esto, supondremos que el usuario introduce la frase sin tildes.

Solución

Actividad resuelta 6.11

Se dispone de la siguiente relación de letras:

| | | | | | | | | | | | |
|-------------|---|---|---|---|---|---|---|---|---|---|---|
| conjunto 1: | e | i | k | m | p | q | r | s | t | u | v |
| conjunto 2: | p | v | i | u | m | t | e | r | k | q | s |

Con ella es posible codificar un texto, convirtiendo cada letra del conjunto 1 en su correspondiente del conjunto 2. El resto de las letras no se modifican. Los conjuntos se utilizan tanto para codificar mayúsculas como minúsculas, mostrando siempre la codificación en minúsculas.

Un ejemplo: la palabra «PaquíTo» se codifica como «matqvko».

Realizar un programa que codifique un texto. Para ello implementar la siguiente función:

```
char codifica(char conjunto1[],char conjunto2[], char c)
```

que devuelve el carácter `c` codificado según los conjuntos 1 y 2 que se le pasan.

Actividad resuelta 6.12

Un anagrama es una palabra que resulta del cambio del orden de los caracteres de otra. Ejemplos de anagramas para la palabra *roma* son: *amor*, *ramo* o *mora*. Construir un programa que solicite al usuario dos palabras e indique si son anagramas una de otra.

Solución

```
import sys
```

Actividad resuelta 6.13

Diseñar un algoritmo que lea del teclado una frase e indique, para cada letra que aparece en la frase, cuántas veces se repite. Se consideran iguales las letras mayúsculas y las minúsculas para realizar la cuenta. Un ejemplo sería:

Frase: En un lugar de La Mancha.

Resultado:

a: 4 veces

c: 1 vez

d: 1 vez

e: 2 veces

...

Actividad resuelta 6.14

Implementar el juego del anagrama, que consiste en que un jugador escribe una palabra y la aplicación muestra un anagrama (cambio del orden de los caracteres) generado al azar. A continuación, otro jugador tiene que acertar cuál es el texto original. La aplicación no

debe permitir que el texto introducido por el jugador 1 sea la cadena vacía. Por ejemplo, si el jugador 1 escribe «teclado», la aplicación muestra como pista un anagrama al azar, como por ejemplo: «etcloda».

Solución

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        String original; //texto original que introduce el jugador 1
```


Actividad resuelta 6.15

Modificar la Actividad resuelta 6.14 para que el programa indique al jugador 2 cuántas letras coinciden (son iguales y están en la misma posición) entre el texto introducido por él y el original.

- 6.1. En Java, al igual que en otros muchos lenguajes de programación, las secuencias de escape se escriben mediante:
- a) Dos puntos (:).
 - b) El carácter *u* mayúscula (U).
 - c) El carácter *u* minúscula (u).
 - d) Un barra invertida (\).
- 6.2. La clase `Character` se encuentra ubicada en el paquete:
- a) `java.util.`
 - b) `java.character.`
 - c) `java.lang.`
 - d) `java.unicode.`
- 6.3. La aritmética de caracteres permite que exista una fuerte relación entre el tipo `char` y el tipo `int`. Sabiendo que el carácter 'a' tiene una representación numérica de 61_{hex}, ¿cómo conseguiremos mostrar el carácter 'i' por consola, a partir de la siguiente variable?
- ```
int codepoint = 0x61;
```
- a) `System.out.println((char) (codepoint + '8'));`
  - b) `System.out.println((char) (codepoint - '8'));`
  - c) `System.out.println((char) (codepoint - 8));`
  - d) `System.out.println((char) (codepoint + 8));`
- 6.4. Marca la opción que devuelve true:
- a) `Character.isLetter('2');`
  - b) `Character.isUpperCase('2');`
  - c) `Character.isLowerCase('2');`
  - d) `Character.isLetterOrDigit('2');`
- 6.5. Señala qué opción es cierta:
- a) 'a' es un carácter.
  - b) 'a' es una cadena de caracteres.
  - c) "a" es un carácter.
  - d) Todas las opciones anteriores son ciertas.

- 6.6. La forma correcta de comparar alfabéticamente dos cadenas es mediante:**
- a) El operador `==`.
  - b) El método `equal()` de `String`.
  - c) El método `equal()` de `Character`.
  - d) Todas permiten comparar dos cadenas.
- 6.7. La forma de extraer el cuarto carácter de la cadena contenida en la variable `cad` es mediante:**
- a) `cad.indexOf(3)`.
  - b) `cad.charAt(3)`.
  - c) `cad.position(3)`.
  - d) `cad.extract(3)`.

- 6.8.** La forma de concatenar dos cadenas es mediante:
- a) El operador `+`.
  - b) El operador `++`.
  - c) El método `concat()`.
  - d) Todas permiten concatenar cadenas.
- 6.9.** El método que permite eliminar los caracteres blancos del principio y el final de una cadena es:
- a) `isWhiteSpace()`.
  - b) `deleteWhiteSpace()`.
  - c) `strip()`.
  - d) `stripLeading()`.
- 6.10.** Existe una relación entre las cadenas (clase `String`) y las tablas de caracteres (`char[]`). ¿Qué métodos permiten convertir un `String` en un `char[]`?
- a) `toCharArray()`.
  - b) `valueOf()`.
  - c) `convertString()`.
  - d) `empty()`.

## ■ Actividades de aplicación

- 6.11.** Escribe un programa descodificador que muestre un texto codificado con el programa realizado en la Actividad resuelta 6.11.
- 6.12.** Realiza el juego del ahorcado. Las reglas del juego son:
- a)** El jugador A teclea una palabra, sin que el jugador B la vea.
  - b)** Ahora se le muestra tantos guiones como letras tenga la palabra secreta. Por ejemplo, para «hola» será «\_ \_ \_ \_».
  - c)** El jugador B intentará acertar, letra a letra, la palabra secreta.
  - d)** Cada acierto muestra la letra en su lugar y las letras no acertadas seguirán ocultas como guiones. Siguiendo con el ejemplo anterior, y suponiendo que se ha introducido: la 'o', la 'j' y la 'a', se mostrará: «\_ o \_ a».
  - e)** El jugador B solo tiene 7 intentos.
  - f)** La partida terminará al acertar todas las letras que forman la palabra secreta (gana el jugador B) o cuando se agoten todos los intentos (gana el jugador A).
- 6.13.** El preprocesador del lenguaje C elimina los comentarios (`/* ... */`) del código fuente antes de compilar. Diseña un programa que lea por teclado una sentencia en C, y elimine los comentarios.

Sentencia: `if (a==3) /* igual a tres */ a++; /* incrementamos a */`

Salida: `if (a==3) a++;`

**6.14.** Diseña una aplicación que se comporte como una pequeña agenda. Mediante un menú el usuario podrá elegir entre:

- a) Añadir un nuevo contacto (nombre y teléfono).
- b) Buscar el teléfono de un contacto a partir de su nombre.
- c) Mostrar la información de todos los contactos ordenados alfabéticamente mediante el nombre.

*Pista:* El nombre y el teléfono se pueden codificar como una única cadena con la forma «nombre:teléfono».

**6.15.** Escribe un programa que lea el título y el contenido de una página web. La aplicación generará por consola un documento HTML donde el título será un encabezado de primer nivel (<h1>) y el resto del contenido será un párrafo (<p>).

**6.16.** Lee una palabra o frase y muestra el proceso en el que cada letra se sustituye por otro símbolo no alfabético. Por ejemplo el carácter 'a' se podría sustituir por el carácter '@', la 'e' por '€', la 'i' por '¡', etcétera.

**6.17.** Construir un programa que convierta una palabra en secuencias de  $n$  letras. Por ejemplo, la palabra «destornillador», dividida en secuencias de 4 letras, se mostrará de la siguiente forma:

```
dest
orni
llad
or
```

**6.18.** Escribe una aplicación que convierta una frase (que puede estar escrita con cualquier combinación de mayúsculas y minúsculas) en el nombre de una variable que utilice la nomenclatura Camel. Por ejemplo, la frase «Me GUsta merenDAR gaLLEtas», se convertirá en «meGustaMerendarGalletas».

Supondremos que cada palabra que compone la frase está separada por un único espacio en blanco.

- 6.19.** Implementa un sencillo editor de texto que, una vez que se ha introducido el texto, permita reemplazar todas las ocurrencias de una palabra por otra.
- 6.20.** Implementa un programa que lea una frase y muestre todas sus palabras ordenadas de forma alfabética. Suponemos que cada palabra de la frase se separa de otra por un único espacio.