

Hoja de Ejercicios de programación: cadenas de caracteres

Bloque 1

Ejercicio 1

Escribir por pantalla cada carácter de una cadena introducida por teclado.

Ejercicio 2

Realizar un programa que comprueba si una cadena leída por teclado comienza por una subcadena introducida por teclado.

Ejercicio 3

Pide una cadena y un carácter por teclado (valida que sea un carácter) y muestra cuantas veces aparece el carácter en la cadena.

Ejercicio 4

Suponiendo que hemos introducido una cadena por teclado que representa una frase (palabras separadas por espacios), realiza un programa que cuente cuantas palabras tiene.

Ejercicio 5

Si tenemos una cadena con un nombre y apellidos, realizar un programa que muestre las iniciales en mayúsculas.

Ejercicio 6

Realizar un programa que dada una cadena de caracteres por caracteres, genere otra cadena resultado de invertir la primera.

Ejercicio 7

Pide una cadena y dos caracteres por teclado (valida que sea un carácter), sustituye la aparición del primer carácter en la cadena por el segundo carácter.

Ejercicio 8

Realizar un programa que lea una cadena por teclado y convierta las mayúsculas a minúsculas y viceversa.

Ejercicio 9

Realizar un programa que compruebe si una cadena contiene una subcadena. Las dos cadenas se introducen por teclado.

Ejercicio 10

Introducir una cadena de caracteres e indicar si es un palíndromo. Una palabra palíndroma es aquella que se lee igual adelante que atrás

Ejercicio 11. Realizar una función a la que se le pasa por parámetro una cadena con un email, y dicha función devuelve si el mail está bien formado.

Un email está bien formado cuando:

- 1.1.- Contiene una y solo una arroba.
- 1.2.- No puede contener dos puntos seguidos '.' después de la arroba.
- 1.3.- No puede contener ' '

Ejercicio 12. Realizar un programa en java que gestione los datos de stock de una tienda de comestibles, la información a recoger será: nombre del producto (no se puede repetir), precio, cantidad en stock.

La tienda dispone de 10 productos distintos. El programa debe ser capaz de:

- 1. Dar de alta un producto nuevo.
- 2. Buscar un producto por su nombre.
- 3. Modificar el stock y precio de un producto dado.

Bloque 2

1. Pedir al usuario su nombre y saludarlo.
2. Programa que diga si una cadena dada por el usuario está o no vacía.
3. Pedir al usuario una cadena y decir si empieza por el carácter 'H'
4. Como el anterior, pero sin importar si es mayúscula o minúscula.
5. Programa que pida un carácter c y un número n y que devuelva una cadena con c repetido n veces.
6. Elabora un programa que escriba todos los caracteres de una cadena cada uno en una línea.
7. Pedir al usuario una frase y un carácter, y decir en qué posiciones está ese carácter, o indicar que no está si procede. No se puede usar indexOf o similar.
8. Repite el ejercicio anterior pero usando indexOf.
9. Programa que reciba como entrada una cadena y un carácter y escriba cuántas veces aparece ese carácter en la cadena. No se puede usar indexOf.
10. Repite el ejercicio anterior pero esta vez usando indexOf.
11. Programa que lea una frase carácter a carácter hasta que reciba '#' y lo guarde todo en una cadena.

12. Contar el número de vocales de una frase.

13. Trata de averiguar el número de vocales de una frase pero sin usar una variable contador.

14. Sustituir todos los espacios de una frase por * (sin usar replace). Pedir una frase y un carácter, escribir en pantalla la frase inicial pero sólo hasta la primera aparición del carácter dado (sin incluirlo). No se puede utilizar indexOf.

15. Repite el ejercicio anterior usando indexOf.

16. Pedir una frase y eliminar las vocales. No se puede usar replace.

17. Repite el ejercicio anterior usando replace.

18. Sustituir mayúsculas por minúsculas y minúsculas por mayúsculas en una frase.

19. Dada una frase, comprobar si el balanceo de paréntesis es correcto (se abren y se cierran correctamente).

20. Programa que diga cuántas palabras de menos de 3 letras tiene una frase, considerando como separador de palabras únicamente el inicio de cadena, el espacio y el fin de cadena.

21. Programa que lea una cadena. Esa cadena contendrá una serie de palabras separadas por espacios. Escribir en líneas separadas cada una de las palabras.

22. Indicar el número de palabras de una cadena, teniendo en cuenta todos los posibles delimitadores de palabra: inicio y fin de frase, punto, coma, punto y coma, dos puntos, comillas,

espacios.

23. Calcula cuántas palabras de más de 3 vocales tiene una frase.

24. Pedir una frase y poner el primer carácter de cada palabra en mayúscula.

25. Pedir dos cadenas y juntarlas carácter a carácter. Es decir, si el usuario escribe «pata» y «coco» el resultado es «pcaotcao». OJO: las cadenas tienen que ser del mismo tamaño.

26. Mejorar el anterior para que no importe que las palabras sean de distinto tamaño. Es decir, si el usuario escribe «patata» y «coco» el resultado sería «pcaotcaota», añadiendo la cadena sobrante directamente al final.

27. Eliminar la última palabra de una frase.

28. Programa que lea una cadena que contiene el nombre y apellidos de una persona, y lo devuelva en formato «Apellidos, Iniciales». Ejemplo:

Entrada -> Juan Francisco García Ruiz

Salida -> García Ruiz, J. F.

29. Contar el número de veces que aparece un carácter en un texto.

30. Realiza el mismo ejercicio anterior pero sin usar una variable contador.

31. Elimina los caracteres en blanco a la izquierda de una cadena. No puedes usar trim.

32. Programa que elimine los caracteres en blanco a la derecha de una cadena. No puedes usar

trim.

33. Quita los caracteres en blanco a la izquierda y a la derecha de una cadena. No puedes usar trim.

34. Programa que vaya escribiendo una cadena siguiendo la secuencia de posiciones 1er carácter, último, 2º carácter, último-1, ...

35. Programa que diga si una cadena dada por el usuario es o no un número, sin usar la función para convertir a número.

36. Leer una cadena de caracteres, donde en lugar de ñ se han utilizado los caracteres ny. 37. Crear una nueva cadena de caracteres sustituyendo ny por ñ. No puedes usar replace.

38. Programa que escriba una cadena dada por el usuario de forma invertida.

39. Invierte todas las palabras de una frase.

40. Calcula para una cadena dada, cuántas minúsculas tiene.

41. Programa que indique para una cadena dada, cuántas mayúsculas tiene, cuantos números, y cuántos signos de puntuación, indicando el porcentaje de cada uno con respecto a la cadena total.

42. Elabora un programa que codifique una cadena, de tal modo que en el resultado se inviertan cada 2 caracteres. Los caracteres intercambiados no pueden volver a intercambiarse. Ejemplo:

Entrada -> Hola mundo

Salida -> oHalm nuod

43. Realizar un programa que compruebe si una palabra es palíndromo, es decir, si se lee igual de izquierda a derecha que de derecha a izquierda. Por ejemplo: RADAR, ANILINA....

44. Realizar un programa que compruebe si una frase es palíndroma. Se dice que una frase es palíndroma si tras eliminarle todos los espacios la palabra resultante es palíndroma.

«anula las alas a la luna»

«amor a roma»

«dabale arroz a la zorra el abad»

45. Programa que reciba una cadena y dos caracteres y reemplace en la cadena todas las ocurrencias del primer carácter por el segundo.

46. Programa que desplace a la derecha un número de veces dado por el usuario los caracteres de una cadena también dada.

47. Escribir un programa que lea frase y la procese como si pasase a través de una ventana (de 20 posiciones), Debe aparecer dicha frase moviéndose de derecha a izquierda. Cuando por la ventana pase el último carácter de la frase, ésta volverá a desfilar de nuevo desde el principio. Así un número de veces indicado por el usuario.

48. Programa para codificar o decodificar un texto utilizando el método de cifrado de César. Supondremos que el texto solo contiene letras mayúsculas o minúsculas. Las letras serán las correspondientes al alfabeto inglés (26 caracteres, excluimos la ñ y Ñ).

49. En este método de cifrado cada letra del texto se sustituye por otra letra que se encuentra N posiciones adelante en el alfabeto. Se considera que el alfabeto es circular, es decir, la letra siguiente a la 'z' es la 'a'.

Por ejemplo, si N es 3, la 'a' se transformaría en 'd', la 'b' en 'e', la 'c' en 'f', etc.

Ejemplo de cifrado César: si el texto es "casa" y $N = 3$ el texto cifrado es "fdvd"

Para descifrar un texto se realiza la operación contraria. Se calcula la letra que está N posiciones por detrás en el alfabeto. Como el alfabeto es circular, la letra anterior a la 'a' es la 'z'.

50. El programa pedirá por teclado un texto, a continuación el valor de N y si queremos codificar o decodificar el texto. Finalmente se mostrará el texto resultante.

Implementar las operaciones de inserción, borrado, búsqueda y copia de caracteres dentro de una cadena. Sólo se permite utilizar la función Subcadena(cadena,pos,pos) para una posición, no para varios caracteres.

51. Dada una cadena, en la que cada vez que aparece la palabra número a continuación, y tras un espacio en blanco, se encuentra un dígito, construir un programa que calcule el doble de cada dígito que aparece en la cadena y los muestre en orden inverso al que se introdujeron.

Dada una cadena de entrada, comprobar si es una contraseña FUERTE o DÉBIL. Se considera que una contraseña es fuerte si contiene 8 o más caracteres, y entre ellos al menos hay una mayúscula, una minúscula, un signo de puntuación y un dígito.

52. Escribir un programa que codifique y decodifique una frase. El procedimiento se invierte para decodificar la línea de texto. Para codificar el texto se emplea el siguiente proceso:

Convertir cada carácter (incluido blancos) en su ASCII equivalente.

Generar un entero aleatorio y sumarlo a cada valor ASCII (el mismo valor se usa en todo el proceso).

Los valores N1 y N2 representan el valor menor y mayor permitido en el código ASCII. Si el número obtenido en el paso anterior es mayor que N2 se resta de ese número el mayor múltiplo posible de N2 y se suma la diferencia a N1.

Imprimir los caracteres ASCII.