

FAMILIA PROFESIONAL:

CICLOS FORMATIVOS:

MÓDULO:

Informática y Comunicaciones

Desarrollo de Aplicaciones Multiplataforma,

Desarrollo de Aplicaciones Web

Programación

UNIDAD 4: CADENAS DE CARACTERES Y EXPRESIONES REGULARES

ACTIVIDADES



AUTORES: **Fernando Rodríguez Alonso**
Sonia Pasamar Franco

Este documento está bajo licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional License.

Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

Usted es libre de:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato.

El licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia.

Bajo los siguientes términos:

- **Atribución** — Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.
- **NoComercial** — Usted no puede hacer uso del material con propósitos comerciales.
- **SinDerivadas** — Si remezcla, transforma o crea a partir del material, no podrá distribuir el material modificado.

No hay restricciones adicionales — No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

CARACTERES

ACTIVIDAD 1x01

Codifica una función **esNumeroArabigo** que tenga como parámetro un carácter y que devuelva verdadero si dicho carácter es un número arábigo (del 0 al 9) o falso en caso contrario.

Codifica un programa principal **main** que lea por teclado un carácter y que visualice en consola un mensaje informativo indicando si dicho carácter es un número arábigo o no.

Un ejemplo de ejecución del programa podría ser:

```
¿Carácter? 5
El carácter 5 es un número arábigo.
```

Otro ejemplo de ejecución del programa podría ser:

```
¿Carácter? A
El carácter A no es un número arábigo.
```

ACTIVIDAD 1x02

Codifica una función **esLetraMayuscula** que tenga como parámetro un carácter y que devuelva verdadero si dicho carácter es una letra mayúscula (según el alfabeto español, incluyendo la letra Ñ) o falso en caso contrario.

Codifica una función **esLetraMinuscula** que tenga como parámetro un carácter y que devuelva verdadero si dicho carácter es una letra mayúscula (según el alfabeto español, incluyendo la letra ñ) o falso en caso contrario.

Codifica un programa principal **main** que lea por teclado un carácter y que visualice en consola un mensaje informativo indicando si dicho carácter es una letra mayúscula, es una letra minúscula o no es una letra.

Un primer ejemplo de ejecución del programa podría ser:

```
¿Carácter? E
El carácter E es una letra mayúscula.
```

Un segundo ejemplo de ejecución del programa podría ser:

```
¿Carácter? n
El carácter n es una letra minúscula.
```

Un tercer ejemplo de ejecución del programa podría ser:

```
¿Carácter? 7
El carácter 7 no es una letra.
```

ACTIVIDAD 1x03 (repaso)

Codifica una función **esSignoPuntuacion** que tenga como parámetro un carácter y que devuelva verdadero si dicho carácter es un signo de puntuación o falso en caso contrario. Los signos de puntuación son el punto (.), la coma (,), el punto y coma (;), los dos puntos (:), el guión (-), las comillas (''), los paréntesis (() [] { }), los signos de interrogación (¿ ?) y los signos de exclamación (¡ !).

Codifica un programa principal **main** que lea por teclado un carácter y que visualice en consola un mensaje informativo indicando si dicho carácter es un signo de puntuación o no.

Un ejemplo de ejecución del programa podría ser:

```
¿Carácter? :  
El carácter : es un signo de puntuación.
```

Otro ejemplo de ejecución del programa podría ser:

```
¿Carácter? T  
El carácter T no es un signo de puntuación.
```

ACTIVIDAD 1x04 (ampliación)

Codifica una función **convertirALetraMayuscula** que tenga como parámetro un carácter. Si dicho carácter es una letra minúscula (según el alfabeto español, incluyendo la letra ñ), devolverá la letra mayúscula correspondiente a este carácter. En caso contrario, devolverá el propio carácter sin convertir.

Codifica una función **convertirALetraMinuscula** que tenga como parámetro un carácter. Si dicho carácter es una letra mayúscula (según el alfabeto español, incluyendo la letra Ñ), devolverá la letra minúscula correspondiente a este carácter. En caso contrario, devolverá el propio carácter sin convertir.

Codifica un programa principal **main** que lea por teclado un carácter y que:

- Visualice en consola un mensaje informativo indicando dicho carácter convertido a letra mayúscula.
- Visualice en consola un mensaje informativo indicando dicho carácter convertido a letra minúscula.

Un primer ejemplo de ejecución del programa podría ser:

```
¿Carácter? e  
El caracter e convertido a letra mayúscula es E.  
El caracter e convertido a letra minúscula es e.
```

Un segundo ejemplo de ejecución del programa podría ser:

```
¿Carácter? R  
El caracter R convertido a letra mayúscula es R.  
El caracter R convertido a letra minúscula es r.
```

Un tercer ejemplo de ejecución del programa podría ser:

```
¿Carácter? 2  
El caracter 2 convertido a letra mayúscula es 2.  
El caracter 2 convertido a letra minúscula es 2.
```

CADENAS DE CARACTERES 1

ACTIVIDAD 2x01

Codifica un programa principal **main** que:

- Lea por teclado una frase que tenga 60 caracteres como máximo. Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.
- Visualice en consola el número de letras mayúsculas de la frase y visualice en consola las letras mayúsculas contenidas en la frase. Las letras mayúsculas permitidas son las letras del alfabeto español escritas en mayúscula (incluida la letra Ñ).
- Visualice en consola el número de letras minúsculas de la frase y visualice en consola las letras minúsculas contenidas en la frase. Las letras minúsculas permitidas son las letras del alfabeto español escritas en minúscula (incluida la letra ñ).

Un ejemplo de ejecución del programa podría ser:

```
¿Frase? Cataluña tiene 4 provincias: Barcelona, Gerona, Lérida y Tarragona.
La frase debe tener 60 caracteres como máximo.
¿Frase? Aragón tiene 3 provincias: Zaragoza, Huesca y Teruel.
La frase tiene 4 letras mayúsculas.
Letras Mayúsculas: AZHT
La frase tiene 37 letras minúsculas.
Letras Minúsculas: ragntieneprovinciasaragozauescayeruel
```

ACTIVIDAD 2x02 (repaso)

Codifica un programa principal **main** que:

- Lea por teclado una frase que tenga 60 caracteres como máximo. Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.
- Visualice en consola el número de signos de puntuación de la frase y visualice en consola los signos de puntuación contenidos en la frase. Los signos de puntuación permitidos son el punto (.), la coma (,), el punto y coma (;), los dos puntos (:), el guión (-), las comillas (''), los paréntesis ([] { }), los signos de interrogación (¿ ?) y los signos de exclamación (¡ !).

Un ejemplo de ejecución del programa podría ser:

```
¿Frase? Cataluña tiene 4 provincias: Barcelona, Gerona, Lérida y Tarragona.
La frase debe tener 60 caracteres como máximo.
¿Frase? Aragón tiene 3 provincias: Zaragoza, Huesca y Teruel.
La frase tiene 3 signos de puntuación.
Signos de Puntuación: :.,.
```

ACTIVIDAD 2x03

Codifica un programa principal **main** que:

- Lea por teclado una frase que tenga 60 caracteres como máximo. Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.
- Visualice en consola el número de letras vocales de la frase y visualice en consola las letras vocales contenidas en la frase. Las letras vocales permitidas son las letras vocales del alfabeto español escritas en mayúscula y en minúscula.
- Visualice en consola el número de letras consonantes de la frase y visualice en consola las letras consonantes contenidas en la frase. Las letras consonantes permitidas son las letras consonantes del alfabeto español escritas en mayúscula y en minúscula (incluidas las letras Ñ y ñ).

Un ejemplo de ejecución del programa podría ser:

```
¿Frase? Cataluña tiene 4 provincias: Barcelona, Gerona, Lérida y Tarragona.
La frase debe tener 60 caracteres como máximo.
¿Frase? Aragón tiene 3 provincias: Zaragoza, Huesca y Teruel.
La frase tiene 19 letras vocales.
Letras Vocales: Aaieeoiaaaaoaueaeue
La frase tiene 22 letras consonantes.
Letras Consonantes: rgntnprvncsZrgzHscyTrl
```

ACTIVIDAD 2x04 (repasso)

Codifica un programa principal **main** que:

- Lea por teclado una frase que tenga 60 caracteres como máximo. Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.
- Visualice en consola una cadena con los caracteres de las posiciones pares de la frase.
- Visualice en consola otra cadena con los caracteres de las posiciones impares de la frase.

Un ejemplo de ejecución del programa podría ser:

```
¿Frase? Hace dos días cayó una tromba de agua que inundó toda la calle.
La frase debe tener 60 caracteres como máximo.
¿Frase? Hoy no hay nubes en el cielo y hace calor.
Los caracteres de posiciones pares de la frase son:
Hyn a ue ne il aeclr
Los caracteres de posiciones impares de la frase son:
o ohynbse lceoyhc ao.
```

ACTIVIDAD 2x05

Codifica un programa principal `main` que gestione un menú con dos opciones:

0) **Salir del programa.**

1) **Convertir todos los caracteres de una frase a letras mayúsculas.**

Leerá por teclado una frase que tenga 60 caracteres como máximo. Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.

Convertirá todos los caracteres de la frase a letras mayúsculas y visualizará en consola esta frase modificada.

2) **Convertir todos los caracteres de una frase a letras minúsculas.**

Leerá por teclado una frase que tenga 60 caracteres como máximo. Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.

Convertirá todos los caracteres de la frase a letras minúsculas y visualizará en consola esta frase modificada.

Este programa principal deberá validar que la opción de menú elegida sea válida (comprendida entre 0 y 2). Si no lo es, visualizará en consola el mensaje:

La opción de menú debe estar comprendida entre 0 y 2.

Un ejemplo de ejecución del programa podría ser:

```
(0) Salir del programa.
(1) Convertir todos los caracteres de la frase a letras mayúsculas.
(2) Convertir todos los caracteres de la frase a letras minúsculas.
¿Opción? 1
¿Frase? Voy a viajar a Sevilla, pasando por Madrid.
VOY A VIAJAR A SEVILLA, PASANDO POR MADRID.
```

```
(0) Salir del programa.
(1) Convertir todos los caracteres de la frase a letras mayúsculas.
(2) Convertir todos los caracteres de la frase a letras minúsculas.
¿Opción? 2
¿Frase? Voy a viajar a Sevilla, pasando por Madrid.
voy a viajar a sevilla, pasando por madrid.
```

```
(0) Salir del programa.
(1) Convertir todos los caracteres de la frase a letras mayúsculas.
(2) Convertir todos los caracteres de la frase a letras minúsculas.
¿Opción? 3
La opción de menú debe estar comprendida entre 0 y 2.
```

```
(0) Salir del programa.
(1) Convertir todos los caracteres de la frase a letras mayúsculas.
(2) Convertir todos los caracteres de la frase a letras minúsculas.
¿Opción? 0
```

ACTIVIDAD 2x06

Codifica un programa principal **main** que:

- Lea por teclado una frase que tenga 5 caracteres como mínimo y 40 caracteres como máximo. Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.
- Visualice en consola la frase progresivamente, carácter a carácter. Es decir, visualizará en la primera línea el primer carácter de la frase, y en cada línea siguiente, visualizará la frase parcial que se ha mostrado en la línea anterior y el siguiente carácter de la frase.

Un ejemplo de ejecución del programa podría ser:

```
¿Frase? hola
La frase debe tener 5 caracteres como mínimo.
La frase debe tener 40 caracteres como máximo.
¿Frase? buenas tardes
b
bu
bue
buen
buena
buenas
buenas
buenas t
buenas ta
buenas tar
buenas tard
buenas tarde
buenas tardes
```

ACTIVIDAD 2x07

Codifica un programa principal **main** que:

- Lea por teclado una frase que tenga 2 caracteres como mínimo y 40 caracteres como máximo. Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.
- Visualice en consola todas las rotaciones hacia la derecha posibles de la frase, de forma que, en la siguiente rotación de caracteres, el primer carácter de la frase sea el último y los demás caracteres se trasladen una posición a la derecha.
- Visualice en consola todas las rotaciones hacia la izquierda posibles de la frase, de forma que, en la siguiente rotación de caracteres, el último carácter de la frase sea el primero y los demás caracteres se trasladen una posición a la izquierda.

Un ejemplo de ejecución del programa podría ser:

```
¿Frase? Buenas días, buenas tardes y buenas noches.
La frase debe tener 2 caracteres como mínimo.
La frase debe tener 40 caracteres como máximo.
¿Frase? hola
```


Rotaciones hacia la derecha:

hola
ahol
laho
olah

Rotaciones hacia la izquierda:

hola
olah
laho
ahol

ACTIVIDAD 2x08

Codifica un programa principal **main** que:

- Lea por teclado dos nombres completos que tengan dos palabras. Cada nombre completo deberá tener un único espacio en blanco, no deberá comenzar con un espacio en blanco y no deberá terminar con un espacio en blanco. Si un nombre completo no cumple las condiciones requeridas, se visualizará en consola un mensaje de error y se leerá por teclado otro nombre completo. Este proceso se repetirá hasta que se introduzca un nombre completo válido.
- Visualice en consola los dos nombres completos con los apellidos intercambiados entre sí.

Un ejemplo de ejecución del programa podría ser:

```
¿Nombre Completo 1? Jorge Pérez
¿Nombre Completo 2? María García
Jorge García
María Pérez
```

Otro ejemplo de ejecución del programa podría ser:

```
¿Nombre Completo 1? Isabel Serrano Martínez
El nombre completo debe tener 1 espacio en blanco.
El nombre completo no debe comenzar con un espacio en blanco.
El nombre completo no debe terminar con un espacio en blanco.
¿Nombre Completo 1? Isabel Serrano
¿Nombre Completo 2? Pedro Fernández Gil
El nombre completo debe tener 1 espacio en blanco.
El nombre completo no debe comenzar con un espacio en blanco.
El nombre completo no debe terminar con un espacio en blanco.
¿Nombre Completo 2? Pedro Fernández
Isabel Fernández
Pedro Serrano
```

ACTIVIDAD 2x09 (ampliación)

Codifica un programa principal **main** que:

- Lea por teclado un nombre completo que tenga tres palabras. El nombre completo deberá contener un nombre de pila, un primer apellido y un segundo apellido, separados por dos espacios en blanco. Si un nombre completo no cumple las condiciones requeridas, se visualizará en consola un mensaje de error y se leerá por teclado otro nombre completo. Este proceso se repetirá hasta que se introduzca un nombre completo válido.

- Visualice en consola el nombre completo con el formato Apellido1 Apellido2, Nombre.
- Visualice en consola las iniciales del nombre completo escritas en letras mayúsculas.

Un ejemplo de ejecución del programa podría ser:

```
¿Nombre Completo? Javier Navarro Hernández
Navarro Hernández, Javier
NHJ
```

Otro ejemplo de ejecución del programa podría ser:

```
¿Nombre Completo? Marta Romero
El nombre completo debe tener 2 espacios en blanco.
El nombre completo no debe comenzar con un espacio en blanco.
El nombre completo no debe terminar con un espacio en blanco.
¿Nombre Completo? Marta Romero Echegaray
Romero Echegaray, Marta
REM
```

ACTIVIDAD 2x10

Codifica una función **contarEspacios** que tenga como parámetro una frase y que devuelva el número de espacios en blanco que hay contenidos en dicha frase.

Codifica un programa principal **main** que:

- Lea por teclado una frase que tenga 10 caracteres como mínimo y 60 caracteres como máximo. Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida:
 - La frase puede tener espacios en blanco al principio y/o al final.
 - Entre dos palabras cualesquiera de la frase hay al menos un espacio en blanco.
- Visualice en consola el número de palabras que tiene la frase.

Un ejemplo de ejecución del programa podría ser:

```
¿Frase? adiós
La frase debe tener 10 caracteres como mínimo.
La frase debe tener 60 caracteres como máximo.
¿Frase? hola, ¿qué tal estás?
La frase tiene 4 palabras.
```

ACTIVIDAD 2x11

Codifica una función **invertirPalabra** que tenga como parámetro una palabra y que devuelva la palabra invertida, construida con los caracteres de dicha palabra colocados en orden inverso.

Codifica un programa principal **main** que:

- Lea por teclado una palabra (sin ningún espacio en blanco). Si la palabra no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra palabra. Este proceso se repetirá hasta que se introduzca una palabra válida.
- Visualice en consola la palabra invertida.

- Visualice en consola un mensaje informativo indicando si la palabra es palíndroma o no es palíndroma. Una palabra es palíndroma cuando es igual a su correspondiente palabra invertida.

Un ejemplo de ejecución del programa podría ser:

```
¿Palabra? saber hacer
La palabra no debe contener ningún espacio en blanco.
¿Palabra? reconocer
Palabra Invertida: reconocer
La palabra reconocer es palíndroma.
```

Otro ejemplo de ejecución del programa podría ser:

```
¿Palabra? saber ser
La palabra no debe contener ningún espacio en blanco.
¿Palabra? interpretar
Palabra Invertida: raterpretni
La palabra interpretar no es palíndroma.
```

ACTIVIDAD 2x12 (ampliación)

Codifica una función **invertirFrase** que tenga como parámetro una frase y que devuelva la frase invertida, construida con las palabras de dicha frase colocadas en orden inverso.

Codifica un programa principal **main** que:

- Lea por teclado una frase que tenga 10 caracteres como mínimo y 60 caracteres como máximo. Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.
- Visualice en consola la frase invertida.

Un ejemplo de ejecución del programa podría ser:

```
¿Frase? La ventana superior izquierda de esa casa tiene un cristal roto.
La frase debe tener 10 caracteres como mínimo.
La frase debe tener 60 caracteres como máximo.
¿Frase? La puerta del garaje está pintada de blanco, azul y rojo.
Frase Invertida: rojo. y azul blanco, de pintada está garaje del puerta La
```

ACTIVIDAD 2x13

Codifica una función **contarApariciones** que tenga como parámetros una letra y una frase y que devuelva el número de veces que aparece dicha letra en dicha frase.

Codifica un programa principal **main** que:

- Lea por teclado una frase en letras mayúsculas (que no contenga ninguna letra minúscula). Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.
- Para cada letra mayúscula que aparezca en la frase, visualice en consola el número de apariciones de dicha letra mayúscula en la frase. Las letras mayúsculas permitidas son las letras del alfabeto español escritas en mayúscula (incluida la letra Ñ).
- Visualice en consola el número de letras mayúsculas distintas que tiene la frase.

Un ejemplo de ejecución del programa podría ser:

```
¿Frase? Los colores primarios de la luz son ROJO, VERDE y AZUL.
La frase no debe contener ninguna letra minúscula.
¿Frase? LOS COLORES PRIMARIOS DE LA LUZ SON ROJO, VERDE Y AZUL.
La frase tiene 17 letras distintas:
La letra A aparece 3 veces.
La letra C aparece 1 veces.
La letra D aparece 2 veces.
La letra E aparece 4 veces.
La letra I aparece 2 veces.
La letra J aparece 1 veces.
La letra L aparece 5 veces.
La letra M aparece 1 veces.
La letra N aparece 1 veces.
La letra O aparece 7 veces.
La letra P aparece 1 veces.
La letra R aparece 5 veces.
La letra S aparece 4 veces.
La letra U aparece 2 veces.
La letra V aparece 1 veces.
La letra Y aparece 1 veces.
La letra Z aparece 2 veces.
```

ACTIVIDAD 2x14 (ampliación)

Codifica una función **contarApariciones** que tenga como parámetros una letra doble y una frase y que devuelva el número de veces que aparece dicha letra doble en dicha frase.

Codifica un programa principal **main** que:

- Lea por teclado una frase en letras mayúsculas (que no contenga ninguna letra minúscula). Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.
- Para cada letra mayúscula doble que aparezca en la frase, visualice en consola el número de apariciones de dicha letra mayúscula doble en la frase. Las letras mayúsculas permitidas son las letras del alfabeto español escritas en mayúscula (incluida la letra Ñ).
- Visualice en consola el número de letras mayúsculas dobles distintas que tiene la frase.

Un ejemplo de ejecución del programa podría ser:

```
¿Frase? Tuvo un accidente con la silla EN LA CARRERA DE CAMELLOS.
La frase no debe contener ninguna letra minúscula.
¿Frase? TUVO UN ACCIDENTE CON LA SILLA EN LA CARRERA DE CAMELLOS.
La frase tiene 3 letras dobles distintas:
La letra doble CC aparece 1 veces.
La letra doble LL aparece 2 veces.
La letra doble RR aparece 1 veces.
```

ACTIVIDAD 2x15 (ampliación)

Codifica un programa principal **main** que:

- Lea por teclado una frase en letras mayúsculas (que no contenga ninguna letra minúscula). Si la frase no cumple la condición requerida, se visualizará en consola un mensaje de error y se leerá por teclado otra frase. Este proceso se repetirá hasta que se introduzca una frase válida.
- Visualice en consola la letra mayúscula vocal más frecuente (o las letras mayúsculas vocales más frecuentes, si hay varias) de la frase.
- Visualice en consola el número de apariciones de la letra mayúscula vocal más frecuente (o de las letras mayúsculas vocales más frecuentes, si hay varias) en la frase.

Un ejemplo de ejecución del programa podría ser:

```
¿Frase? He aprobado PROGRAMACION con un siete.
La frase no debe contener ninguna letra minúscula.
¿Frase? HE APROBADO PROGRAMACION CON UN SIETE.
Letras Vocales con más Apariciones: 0
Número de Apariciones: 5
```

Otro ejemplo de ejecución del programa podría ser:

```
¿Frase? Mañana tengo un examen de BASES DE DATOS.
La frase no debe contener ninguna letra minúscula.
¿Frase? MAÑANA TENGO UN EXAMEN DE BASES DE DATOS.
Letras Vocales con más Apariciones: AE
Número de Apariciones: 6
```

ACTIVIDAD 2x16 (ampliación)

Codifica una función **contarApariciones** que tenga como parámetros una letra y una palabra y que devuelva el número de veces que aparece dicha letra en dicha palabra.

Codifica una función **sonAnagramas** que tenga como parámetros dos palabras y que devuelva verdadero si las dos palabras son anagramas o falso en caso contrario. Dos palabras son anagramas cuando una palabra es una permutación de los caracteres de la otra palabra y viceversa.

Codifica un programa principal **main** que:

- Lea por teclado dos palabras en letras mayúsculas (que no contengan ninguna letra minúscula y ningún espacio en blanco). Si una palabra no cumple las condiciones requeridas, se visualizará en consola un mensaje de error y se leerá por teclado otra palabra. Este proceso se repetirá hasta que se introduzca una palabra válida.
- Visualice en consola un mensaje informativo indicando si las dos palabras son anagramas o no son anagramas.

Un ejemplo de ejecución del programa podría ser:

```
¿Palabra 1? electro magnetismo
La palabra no debe contener ningún espacio en blanco.
La palabra no debe contener ninguna letra minúscula.
¿Palabra 1? IRONICAMENTE
¿Palabra 2? RENACIMIENTO
Las palabras IRONICAMENTE y RENACIMIENTO son anagramas.
```

Otro ejemplo de ejecución del programa podría ser:

```
¿Palabra 1? super posicion
La palabra no debe contener ningún espacio en blanco.
La palabra no debe contener ninguna letra minúscula.
¿Palabra 1? ESCENARIO
¿Palabra 2? NECESIDAD
Las palabras ESCENARIO y NECESIDAD no son anagramas.
```

CADENAS DE CARACTERES 2

ACTIVIDAD 3x01

Codifica una función **esValido** que compruebe la validez de un código de caracteres correspondiente a un alumno. Esta función tendrá como parámetro un código de alumno y devolverá verdadero si dicho código de alumno cumple las siguientes condiciones o falso en caso contrario:

- El código deberá tener 6 caracteres.
- Los cuatro primeros caracteres serán un año comprendido entre 1990 y 1995.
- El quinto carácter será el curso, que solo podrá ser 1 o 2.
- El sexto carácter será el sexo, que solo podrá ser H o M.

Codifica un programa principal **main** que:

- Lea por teclado códigos de alumno hasta que se introduzca 000000. Para cada código de alumno, verificará la validez del código y visualizará en consola un mensaje informativo indicando si el código de alumno es válido o no es válido.
- Visualice en consola el número de códigos de alumno válidos que se han leído.
- Visualice en consola el número de alumnos hombres y el número de alumnas mujeres que hay en primer curso y en segundo curso.

Un ejemplo de ejecución del programa podría ser:

```
¿Código de Alumno ('000000' para salir)? 19901H
El Código de Alumno es válido.
¿Código de Alumno ('000000' para salir)? 19921H
El Código de Alumno es válido.
¿Código de Alumno ('000000' para salir)? 19941M
El Código de Alumno es válido.
¿Código de Alumno ('000000' para salir)? 1999XX
El Código de Alumno no es válido.
¿Código de Alumno ('000000' para salir)? 2002YY
El Código de Alumno no es válido.
¿Código de Alumno ('000000' para salir)? 19932M
El Código de Alumno es válido.
¿Código de Alumno ('000000' para salir)? 19952M
El Código de Alumno es válido.
¿Código de Alumno ('000000' para salir)? 19942H
El Código de Alumno es válido.
¿Código de Alumno ('000000' para salir)? 19911H
El Código de Alumno es válido.
¿Código de Alumno ('000000' para salir)? 000000
```

Códigos Válidos: 7

Alumnos Hombres en Primero: 3

Alumnas Mujeres en Primero: 1

Alumnos Hombres en Segundo: 1

Alumnas Mujeres en Segundo: 2

ACTIVIDAD 3x02

Codifica una función **esValido** que compruebe la validez de un código de caracteres correspondiente a un producto. Esta función tendrá como parámetro un código de producto y devolverá verdadero si dicho código de producto cumple las siguientes condiciones o falso en caso contrario:

- El código deberá tener 8 caracteres.
- El primer carácter será la categoría, que solo podrá ser A, B o C.
- Los cuatro siguientes caracteres serán un año comprendido entre 1980 y 2020.
- El sexto carácter será el estado, que solo podrá ser 1 o 0. Este valor indicará si está activo o inactivo.
- Los últimos dos caracteres podrán tener cualquier valor.

Codifica un programa principal **main** que:

- Lea por teclado 10 códigos de producto. Para cada código de producto, verificará la validez del código y visualizará en consola un mensaje informativo indicando si el código de producto es válido o no es válido.
- Visualice en consola el número de códigos de producto válidos que se han leído.
- Visualice en consola el número de productos activos e inactivos que hay en cada categoría.

Un ejemplo de ejecución del programa podría ser:

```
¿Código de Producto 1? A19821XX
El Código de Producto es válido.
¿Código de Producto 2? A19831YY
El Código de Producto es válido.
¿Código de Producto 3? A19841ZZ
El Código de Producto es válido.
¿Código de Producto 4? A1985000
El Código de Producto es válido.
¿Código de Producto 5? B1990311
El Código de Producto no es válido.
¿Código de Producto 6? B1991122
El Código de Producto es válido.
¿Código de Producto 7? C2010133
El Código de Producto es válido.
¿Código de Producto 8? C2011144
El Código de Producto es válido.
¿Código de Producto 9? C2012055
El Código de Producto es válido.
¿Código de Producto 10? D2025266
El Código de Producto no es válido.
```

```
Códigos Válidos: 8
Productos Activos en Categoría A: 3
Productos Inactivos en Categoría A: 1
Productos Activos en Categoría B: 1
Productos Inactivos en Categoría B: 0
Productos Activos en Categoría C: 2
Productos Inactivos en Categoría C: 1
```


ACTIVIDAD 3x03 (ampliación)

Codifica una función **tieneFormatoValido** que compruebe la validez del formato de un código de caracteres. Esta función tendrá como parámetro un código de caracteres y devolverá verdadero si dicho código de caracteres cumple las siguientes condiciones o falso en caso contrario:

- El código deberá tener 11 caracteres.
- Los seis primeros caracteres serán letras consonantes mayúsculas (sin la letra Ñ).
- El séptimo carácter será el separador guión (-).
- Los cuatro últimos caracteres serán números arábigos (del 0 al 9).

Codifica una función **tieneContenidoValido** que compruebe la validez del contenido de un código de caracteres. Esta función tendrá como parámetro un código de caracteres y devolverá verdadero si dicho código de caracteres cumple las siguientes condiciones o falso en caso contrario:

- Las letras consonantes mayúsculas deberán comenzar por ZRG, HSC o TRL.
- Las letras consonantes mayúsculas deberán terminar en PQ, MD o GR.
- Los números arábigos deberán sumar 20 al menos.

Codifica un programa principal **main** que:

- Lea por teclado 10 códigos de caracteres. Para cada código de caracteres:
 - Verificará la validez del formato del código y la validez del contenido del código.
 - Visualizará en consola un mensaje informativo indicando si el código de caracteres es válido, el contenido del código no es válido o el formato del código no es válido. Un código de caracteres es válido cuando tiene un formato válido y tiene un contenido válido.
- Visualice en consola el número de códigos de caracteres válidos e inválidos que se han leído.

Un ejemplo de ejecución del programa podría ser:

```
¿Código de Caracteres 1? 01234567890123456789
El formato del Código de Caracteres no es válido.
¿Código de Caracteres 2? bcd fgh-0123
El formato del Código de Caracteres no es válido.
¿Código de Caracteres 3? BCDFGH:0123
El formato del Código de Caracteres no es válido.
¿Código de Caracteres 4? BCDFGH-aAeE
El formato del Código de Caracteres no es válido.
¿Código de Caracteres 5? XXXXGR-5555
El contenido del Código de Caracteres no es válido.
¿Código de Caracteres 6? ZRGXXX-5555
El contenido del Código de Caracteres no es válido.
¿Código de Caracteres 7? ZRGXGR-0000
El contenido del Código de Caracteres no es válido.
¿Código de Caracteres 8? ZRGXGR-9999
El Código de Caracteres es válido.
¿Código de Caracteres 9? HSCXMD-8888
El Código de Caracteres es válido.
¿Código de Caracteres 10? TRLXPQ-7777
El Código de Caracteres es válido.
```

Códigos Válidos: 3

Códigos Inválidos: 7

