

## PREUNIVERSITARIO

### *Ejercicios con manejo de cadenas de texto*

#### *Ejercicios*

- 1) Pedir una frase y una letra al usuario. Indicar si en esa frase existe la letra que el usuario ingresó.

Algoritmo Ejercicio1

```
// Definimos las variables necesarias.

Definir frase, letra Como Texto

frase = ""

letra = ""

Definir cantidad, i Como Entero

cantidad = 0

i = 0

Definir encontrada Como Logico

encontrada = Falso // Inicialmente, asumimos que la letra no ha
sido encontrada.

// Solicitar al usuario que ingrese la frase y la letra a buscar.

Escribir "Dime la frase."

Leer frase

Escribir "Dime la letra."

Leer letra

cantidad = Longitud(frase) // Obtener la longitud de la frase
ingresada.

// Iterar a través de la frase para buscar la letra.

Para i = 0 Hasta cantidad-1 Con Paso 1 Hacer

    // Comparamos la letra actual en la posición i con la letra a
    buscar.

    Si Subcadena(frase, i, i) = letra Entonces

        encontrada = Verdadero // Cambiamos a Verdadero si
        encontramos la letra.
```

```
Finsi

FinPara

// Mostrar si la letra se encuentra en la frase.

Si encontrada = Verdadero Entonces

    Escribir "La letra " letra " se encuentra en la frase."

SiNo

    Escribir "La letra " letra " no se encuentra en la frase."

FinSi

FinAlgoritmo
```

**1) Cuenta las vocales y las consonantes que tiene una palabra dada por el usuario.**

Algoritmo Ejercicio2

```
// Definimos las variables necesarias.

Definir palabra, letra Como Cadena

Definir cantidad, i, vocales Como Entero

cantidad = 0 // Número de caracteres de la palabra.

i = 0 // Variable para el bucle.

vocales = 0// Para contar la cantidad de vocales en la palabra.


// Solicitar al usuario que ingrese una palabra.

Escribir "Escribe una palabra"

Leer palabra

palabra = Minusculas(palabra) // Convertir la palabra a minúsculas
para simplificar la comparación.

cantidad = Longitud(palabra) // Obtener la longitud de la palabra
ingresada.

// Iterar a través de los caracteres de la palabra.

Para i = 0 Hasta cantidad-1 Con Paso 1 Hacer

    letra = SubCadena(palabra, i, i) // Obtener el carácter en la
    posición i.

    // Verificar si el carácter es una vocal.

    Si letra = "a" O letra="e" O letra="i" O letra="o" O letra="u"
    Entonces
```

```
vocales = vocales + 1 // Incrementar el contador de
vocales.

Finsi

FinPara

// Mostrar la cantidad de vocales y consonantes en la palabra.

Escribir "La palabra tiene ", vocales, " vocales, y ", cantidad-
vocales, " consonantes."

FinAlgoritmo
```

- 2) Averigua si una palabra dada por el usuario es igual a otra palabra que haya ingresado al darle la vuelta.

Algoritmo Ejercicio3

```
// Definimos las variables necesarias.

Definir palabra, palabra2, inversa, letra Como Cadena;

inversa = ""

letra = ""

Definir cantidad, i Como Entero

cantidad = 0 // Número de caracteres de la palabra.

i = 0 // Variable para el bucle.

// Solicitar al usuario que ingrese dos palabras.

Escribir "Dime una palabra: "

Leer palabra

Escribir "Dime otra palabra: "

Leer palabra2

palabra = Minusculas(palabra) // Convertir la primera palabra a
minúsculas.

palabra2 = Minusculas(palabra2) // Convertir la segunda palabra a
minúsculas.

cantidad = Longitud(palabra) // Obtener la longitud de la primera
palabra.

// Invertir la primera palabra.

Para i = 0 Hasta cantidad-1 Con Paso 1 Hacer

    letra = Subcadena(palabra, i, i) // Obtener el carácter en la
    posición i.
```

```
    inversa = Concatenar(letra, inversa) // Agregar el carácter al
    inicio de la palabra inversa.
```

```
FinPara
```

```
// Comprobar si las palabras son iguales al darles la vuelta a
una.
```

```
Si inversa = palabra2 Entonces
```

```
    Escribir "Son iguales al darle la vuelta a una."
```

```
SiNo
```

```
    Escribir "No son iguales al darle la vuelta a una."
```

```
FinSi
```

```
FinAlgoritmo
```

### 3) Cuenta las palabras que tiene una frase dada por el usuario.

Algoritmo Ejercicio4

```
// Definimos las variables necesarias.
```

```
Definir frase Como Cadena
```

```
Definir espacios, i, cantidad Como Entero
```

```
espacios = 0 // Para contar los espacios en blanco.
```

```
i = 0 // Variable para el bucle.
```

```
cantidad = 0 // Para contar los caracteres de la frase.
```

```
// Solicitar al usuario que ingrese una frase.
```

```
Escribir "Dime una frase: "
```

```
Leer frase
```

```
cantidad = Longitud(frase); // Obtener la longitud de la frase
ingresada.
```

```
// Contar los espacios en blanco en la frase para determinar la
cantidad de palabras.
```

```
Para i = 0 Hasta cantidad-1 Con Paso 1 Hacer
```

```
    Si Subcadena(frase, i, i) = " " Entonces
```

```
        espacios = espacios + 1
```

```
    Finsi
```

```
FinPara
```

```
// Una frase tendrá siempre tantas palabras como espacios + 1.
```

```
Escribir "La frase tiene ", espacios + 1, " palabras."
```

```
FinAlgoritmo
```

- 4) ¡Encuentra la palabra oculta! El siguiente mensaje está encriptado de manera que para obtenerlo hay que escoger la primera letra de cada palabra: "Estoy seguro te urge devolver intereses ahora.". Crea un algoritmo para encontrar la solución al enigma.

```
Algoritmo Ejercicio5
```

```
// Definimos las variables necesarias.

Definir palabra Como Caracter

Definir frase Como Caracter

Definir cantidad Como Entero

Definir i Como Entero

cantidad = 0 // Variable para contar la cantidad de caracteres en
la frase.

// Solicitar al usuario que ingrese una frase.

Escribir "Dime la frase del enigma"

Leer frase

cantidad = Longitud(frase)// Obtener la longitud de la frase
ingresada.

palabra = Subcadena(frase, 0, 0) // Inicializar la variable
'palabra' con la primera letra de la frase.

// Iterar a través de la frase para extraer la primera letra de
cada palabra.

Para i = 0 Hasta cantidad-1 Con Paso 1 Hacer

    Si Subcadena(frase, i, i) == " " Entonces

        palabra = Concatenar(palabra, Subcadena(frase, i + 1, i +
1)) // Concatenar la siguiente letra de la palabra.

    FinSi

FinPara

// Mostrar la palabra encriptada formada por la primera letra de
cada palabra.

Escribir "La palabra encriptada es: ", palabra

FinAlgoritmo
```