

Hoja 5

Importante: Los ejercicios deben entregarse a través de web. Cada ejercicio debe ir en un fichero de nombre:

`<<nombre>><<apellidos>><<numE>>.c`

donde `<<nombre>>` es el nombre del alumno, `<<apellidos>>` los apellidos, y `<<numE>>` es el número del ejercicio.

La fecha de entrega: consultar la página de la actividad en blackboard

Advertencia: No está permitido el uso de la sentencia **while** ni de librerías externas (salvo `stdio.h`) en ninguno de los ejercicios de esta hoja.

Nota: En los ejemplos de salida de cada ejercicio el texto subrayado se ha introducido por el usuario, el resto se ha escrito por el programa.

Cosas importantes para la claridad y la legibilidad del código:

- El programa debe estar bien comentado.
- Elegir un buen nombre para las variables.
- Sangrar adecuadamente las sentencias de control de flujo

Ejercicio 24 (1.5 punto). Escribir un programa que lea 10 números enteros por pantalla y los inserte en un array. Si el último número es positivo, deberá calcular la suma de los números contenidos en las posiciones pares del array. Si el último número es negativo, deberá calcular la suma de los números contenidos en las posiciones impares del array. Si el último número es 0, deberá calcular la suma de todos los números del array. En el caso de que alguno de los números introducidos sea 999, el programa se verá interrumpido y devolverá SALIDA INTERRUMPIDA.

Ejemplo 1:

```
10 numeros:  
5 2 -2 1 6 -3 4 4 12 2  
Resultado=25
```

Ejemplo 2:

```
10 numeros:  
5 12 -2 1 -16 -3 4 4 2 -4  
Resultado=10
```

Ejemplo 3:

```
10 numeros:  
5 -2 1 999  
SALIDA INTERRUMPIDA
```

Introducción a la Programación

Ejercicio 25 (1.5 punto). Escribir un programa que lea dos frases de hasta 100 caracteres cada una y calcule e imprima la frase que tenga mayor número de vocales. En el caso de que ambas frases tengan el mismo número de vocales, el programa escribirá “TIENEN LAS MISMAS VOCALES”.

Ejemplo 1:

Frase 1:

En un lugar de la Mancha

Frase 2:

de cuyo nombre no quiero acordarme

Frase con mas vocales: de cuyo nombre no quiero acordarme

Ejemplo 2:

Frase 1:

Cojo un muelle, y lo tiro por el retrete

Frase 2:

y ya son 2495 muelles los que el retrete se ha tragado

Frase con más vocales: y ya son 2495 muelles los que el retrete se ha tragado

Ejemplo 3:

Frase 1:

Hola, holita

Frase 2:

Jorobate Ned

TIENEN LAS MISMAS VOCALES

Ejercicio 26 (2 puntos): Escribir un programa que reciba una frase (una cadena de palabras separadas por uno o más espacio en blanco) de hasta 100 caracteres y escriba una palabra por línea indicando el número de caracteres de cada palabra.

Ejemplo:

Input:

Solo hay 10 tipos de
personas las que saben binario
y las que no

Output:

Solo 4
hay 3
10 2
tipos 5
de 2
personas 8
las 3
que 3
saben 5
binario 7
y 1
las 3
que 3
no 2

Introducción a la Programación

Ejercicio 27 (2 puntos): Escribe un programa que reciba una frase (una cadena de palabras o números separadas por uno o más espacio en blanco) de hasta 100 caracteres y muestre el número de palabras que hay en ella y el de números. Hay que tener en cuenta que entre dos palabras puede haber más de 1 espacio en blanco e incluso al principio y al final de la frase. Se considera:

- Una **palabra** es un conjunto de caracteres con al menos un carácter no numérico separado por espacio (i.e. asdf, as78, 78d68, 345y).
- Un **número** es un conjunto de dígitos separados por espacio en blanco (i.e. 12, 3).

De esta manera, la frase “ aaa 33aa aa3aa3a44aaa5 333 33 ” tiene 3 palabras y 2 números.

Ejemplo:

Input:	Output:
aaa 33aa aa3aa3a44aaa5 333 33	3 2

Ejercicio 28 (3 puntos). Escribir un programa que sea capaz de cifrar/descifrar frases de hasta 100 caracteres. Para ello ten en cuenta lo siguiente:

- Las frases pueden estar compuestas por los signos básicos de puntuación (‘ ’, ‘ ’, y ‘ . ’) y de palabras. Las palabras deben estar compuestas por letras cuyo código ASCII va de 97 a 122 (es decir, de la ‘a’ a la ‘z’ sin pasar por la ‘ñ’).
- El método de cifrado consiste en desplazar las letras N posiciones hacia la derecha (si N es positivo) o hacia la izquierda (si N es negativo), los signos básicos de puntuación quedan igual. El método de descifrado consiste en desplazar los caracteres N posiciones hacia la izquierda (si N es positivo) o hacia la derecha (si N es negativo). Al realizar el desplazamiento, considera que el alfabeto es circular (es decir, la letra siguiente a la ‘z’ es la ‘a’, y la letra anterior a la ‘a’ es la ‘z’). Por ejemplo, con un N=2 la codificación sería la siguiente:

Letra	a	b	c	d	e	...	v	w	x	y	z
Letra codificada	c	d	e	f	g	...	x	y	z	a	b

El programa deberá solicitar por pantalla el número N que establece las posiciones a desplazar, la opción O que determina si la operación es de cifrado o descifrado, así como la frase a cifrar/descifrar. Tras operar sobre la frase, el programa escribirá la frase ya cifrada/descifrada.

Los casos de error a contemplar son los siguientes:

- En el caso de que la frase introducida no cumpla las condiciones mencionadas en el apartado a, el programa devolverá un mensaje de error en el que se indique el primer carácter que ha generado el error (ver ejemplo 3).
- En el caso de que el valor N introducido sea 0, el programa devolverá error y finalizará la ejecución (ver ejemplo 4).
- En el caso de que el valor O introducido sea distinto de ‘C’ o de la ‘D’, el programa devolverá error y finalizará la ejecución (ver ejemplo 5).

Ejemplo 1:

Posiciones a desplazar:

3

Operacion (C=cifrar D=descifrar):

C

Frase:

cojo un muelle, y lo tiro por el retrete

Frase modificada: frmr xq pxhooh, b or wlur sru ho uhwhwh

Ejemplo 2:

Posiciones a desplazar:

3

Operacion (C=cifrar D=descifrar):

D

Frase:

frmr xq pxhooh, b or wlur sru ho uhwhwh

Frase modificada: cojo un muelle, y lo tiro por el retrete

Ejemplo 3:

Posiciones a desplazar:

3

Operacion (C=cifrar D=descifrar):

C

Frase:

Joróbate Flanders

ERROR, caracter J no permitido

Ejemplo 4:

Posiciones a desplazar:

0

ERROR

Ejemplo 5:

Posiciones a desplazar:

3

Operacion (C=cifrar D=descifrar):

F

ERROR