

Hoja 6

Importante: Los ficheros deben entregarse a través de web (por **Domjudge** y **Blackboard**). Para cada ejercicio se entregan **tres ficheros**:

Las funciones deben ir en un fichero de nombre:

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

El programa principal (la interfaz) debe ir en un fichero de nombre:

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

El fichero de cabecera (header) será de nombre:

`<<nombre>><<apellidos>><<numE>>header.h`

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

Nota 1: En los casos que el programa deba dar un error con ciertos datos de entrada, se deberá mostrar únicamente la palabra **ERROR** en el output.

Nota 2: Cuando se deban imprimir números reales (float) se utilizará el formato `"%.2f"`.

Nota 3: 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.

Ejercicio 29 (5 puntos): El objetivo de este ejercicio es juntar los programas realizados en ejercicios anteriores en un solo programa. Para ello se definen funciones que implementen los ejercicios anteriores y luego se escribe un programa principal que recibe los datos y la operación deseada y muestra los resultados.

Ojo: Los datos para hacer los ejercicios se piden en el programa principal. Las funciones recibirán los datos como argumentos y **no deben pedir ningún dato** al usuario. Está **prohibido** usar variables externas en este ejercicio.

Escribir una función llamada **“calcular”** que reciba tres números **reales** y según lo que se ha explicado en el ejercicio 7 de la hoja 2 calcule el resultado y lo devuelva (como un float) al programa. Utilizar el código de dicho ejercicio (habrá que modificar el tipo de los datos del ejercicio 7) para implementar la función.

Utilizando el código del ejercicio 12 de la hoja 3, escribir una función llamada “**ordenar**” que reciba tres números reales y los imprima ordenados de mayor a menor. Esta función no debe devolver nada al programa principal.

Utilizando el código del ejercicio 17 de la hoja 4, escribir una función llamada “**numerar**” que reciba un número real (habrá que modificar el tipo del número del ejercicio 17) e imprima todos los números **enteros** de 0 hasta el número introducido. La función debe devolver false (0) si el numero introducido es negativo y true (1) en caso contrario. Esta función no debe escribir el mensaje de error.

Utilizando el código del ejercicio 20 de la hoja 4, escribir una función llamada “**serie**” que reciba los tres números reales (habrá que modificar el tipo de los datos del ejercicio 20) e imprima la serie según el ejercicio. Esta función no debe devolver nada al programa.

Escribir un **programa principal** que reciba tres números reales y un carácter que corresponde a una de las siguientes opciones:

‘C’ o ‘c’: calcular
‘O’ o ‘o’: ordenar
‘N’ o ‘n’: numerar
‘S’ o ‘s’: serie

El programa funciona como se indica abajo según la opción elegida:

- Si la opción es ‘C’ o ‘c’, el programa llamará a la función “calcular” e imprimirá el resultado.
- Si la opción es ‘O’ o ‘o’, el programa llamará a la función “ordenar.
- Si la opción es ‘N’ o ‘n’, el programa llamará a la función “numerar” para cada número. Después de cada llamada, si la función devuelve false el programa imprimirá un mensaje de ERROR.
- Si la opción es ‘S’ o ‘s’, el programa llamará a la función “serie”.
- Si la opción elegida no es ninguno de los casos anteriores el programa imprimirá un mensaje de ERROR.

Ejemplo:

2.5 3.76 1.45 C
2.57

Ejemplo:

2.534 5 3.233 o
5.00 3.23 2.53

Ejemplo:

5.9567 -3.633 4.43 N

0 1 2 3 4 5

ERROR

0 1 2 3 4

Ejemplo:

5.2 -4 6.23 s

5.20 -4.00 6.23 -5.03 7.26 -6.06 8.29 -7.09 9.32 -8.12

Ejemplo:

2.5 2.4 5 w

ERROR

Ejercicio 30 (5 puntos): En este ejercicio se reutilizará los códigos de los ejercicios 25, 26, y 27 de la hoja 5 para implementar tres funciones como se explica a continuación.

La función **vocal** recibe una cadena y devuelve el número de vocales en la cadena.

La función **palabras** recibe una cadena y escribe cada palabra (ver la definición de palabra del ejercicio 26) de la frase en una línea indicando el número de caracteres de la palabra. Esta función no devuelva nada.

La función **contar** recibe una cadena e imprime el número de palabras que hay en ella y el de números (ver la definición de palabra y número del ejercicio 27). Esta función no devuelva nada.

Utilizar el siguiente programa main para probar las funciones:

```
#include <stdio.h>
#define MAX 100 // maximo tamaño de la cadena
int main(){
    char frase1[MAX], frase2[MAX];
    int vocales1=0,vocales2=0;

    printf("Frase 1:\n");
    gets(frase1);

    printf("Frase 2:\n");
    gets(frase2);

    // calcular el numero de vocales
    vocales1 = vocal(frase1);
    vocales2 = vocal(frase2);

    //imprimir el resultado
    if (vocales1>vocales2)
        printf("Frase con mas vocales: %s\n",frase1);
    else if (vocales2>vocales1)
        printf("Frase con mas vocales: %s\n",frase2);
```

```
else
    printf("TIENEN LAS MISMAS VOCALES\n");

// contar el numero de palabras y numeros
printf("El numero de palabras y la cantidad de numeros de la frase 1:\n");
contar(frase1);
printf("El numero de palabras y la cantidad de numeros de la frase 2:\n");
contar(frase2);

// contar el numero de caracteres
printf("Cada palabra de la frase 1 y su numero de caracteres:\n");
palabras(frase1);
printf("Cada palabra de la frase 2 y su numero de caracteres:\n");
palabras(frase2);

return 0;
}
```

Ejemplo:

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

El numero de palabras y la cantidad de numeros de la frase 1:

6 0

El numero de palabras y la cantidad de numeros de la frase 2:

6 0

Cada palabra de la frase 1 y su numero de caracteres:

En 2

un 2

lugar 5

de 2

la 2

Mancha 6

Cada palabra de la frase 2 y su numero de caracteres:

de 2

cuyo 4

nombre 6

no 2

quiero 6

acordarme 9

Ejemplo:

Frase 1:

Frase 2:

Jorobate Flanders

Introducción a la Programación

Frase con mas vocales: Jorobate Flanders

El numero de palabras y la cantidad de numeros de la frase 1:

0 0

El numero de palabras y la cantidad de numeros de la frase 2:

2 0

Cada palabra de la frase 1 y su numero de caracteres:

Cada palabra de la frase 1 y su numero de caracteres:

Jorobate 8

Flanders 8

Ejemplo:

Frase 1:

aaa 33aa aa3aa3a44aaa5 333 33

Frase 2:

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

Frase con mas vocales: Solo hay 10 tipos de personas

las que saben binario y las que no

El numero de palabras y la cantidad de numeros de la frase 1:

3 2

El numero de palabras y la cantidad de numeros de la frase 2:

13 1

Cada palabra de la frase 1 y su numero de caracteres:

aaa 3

33aa 4

aa3aa3a44aaa5 13

333 3

33 2

Cada palabra de la frase 2 y su numero de caracteres:

Solo 4

hay 3

10 2

tipos 5

de 2

personas 8

las 3

que 3

saben 5

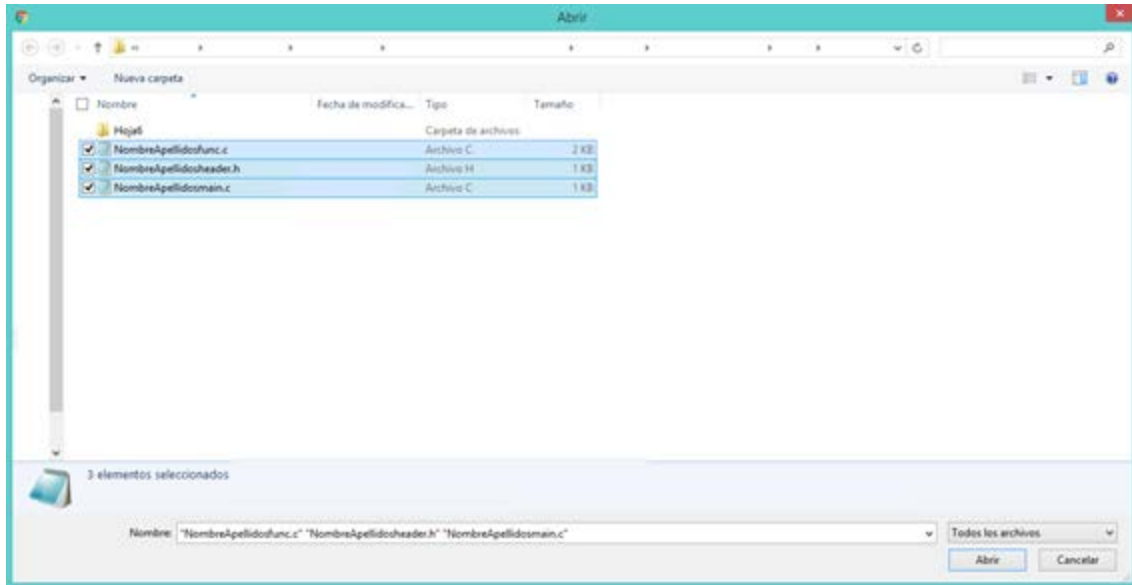
binario 7

y 1

las 3
que 3
no 2

Instrucciones para subir varios ficheros a DomJudge

- 1.- Hacemos click en Elegir archivos como cuando queremos subir un único fichero.
- 2.- Seleccionamos los 3 ficheros (ctrl + click izquierdo en cada uno de ellos) y a continuación hacemos click en abrir.



- 3.-Nos aparecerá que hemos seleccionado 3 archivos seleccionamos el ejercicio y el lenguaje y damos a submit.

Submissions

Elegir archivos 3 archivos Ej23 C submit cancel