


Ejercicio 1 (1 puntos; R3, R4, R5, R6) Tiempo estimado: 10 minutos. Las respuestas acertadas suman 1 punto y las falladas restan 0.5 puntos. Las que no se respondan no restarán nada.

- La función `rewind`:
 - permite volver al principio de una cadena de caracteres
 - permite volver un puntero a su posición original
 - permite volver al principio de una estructura
 - d) permite volver al principio de un fichero**
- La función `ftell` devuelve
 - el tamaño del archivo en bytes
 - el número de bytes que quedan por leer del archivo
 - c) el número de bytes que hay desde el principio del archivo hasta el apuntador**
 - ninguna de las anteriores
- Escribe la salida que efectúa el siguiente trozo de código.

```
char cadena[]="Antonio\n";  
  
printf("%d",strlen(cadena));
```


- Dada la definición `const int *px`; ¿qué es lo que no podemos modificar?
 - `px`
 - b) *px**
 - `px*`
 - podemos modificar todo
- ¿Tras la ejecución de cuáles de las siguientes instrucciones se puede modificar o se modifica el valor de la variable `a`?

```
f(a,10)  
f(&a)  
f(a)  
a=f(a)
```

 - todas pueden modificar/modifican el valor de `a`
 - ninguna modifica el valor de `a`
 - las tres últimas pueden modificar/modifican el valor de `a`
 - d) la segunda y la cuarta pueden modificar/modifican el valor de `a`**
- La instrucción `fseek(f, 100, SEEK_CUR)`; desplazará el apuntador del manejador de fichero representado por `f`:
 - a) 100 bytes desde la posición actual**
 - 100 bits desde la posición actual
 - 100 multiplicado por el tamaño del dato almacenado en el fichero desde la posición actual
 - ninguna de las anteriores
- Dadas dos cadenas de caracteres "a" y "b", si queremos hacer que la cadena "a" pase a tener el contenido de "a" y de "b" unidas (concatenar) tenemos que hacer:
 - `a=a+b`
 - `a= strcat(a,b)`
 - `strcpy(a,b)`
 - d) `strcat(a,b)`**
- Si dentro de una función definimos una variable como `"static int a;"`
 - la variable será accesible a otras funciones
 - la variable conservará su valor de una ejecución a otra de la función
 - la variable se almacena en un registro de la CPU
 - la variable se almacena en el disco duro
- Si en un ordenador un `char` ocupa 1 byte, un `int` 4 bytes y un `float` 4 bytes ¿cuánto ocupará el número real 132.321 al escribirlo como texto a un archivo?
 - 4 bytes
 - 6 bytes
 - 7 bytes
 - 8 bytes
- La función `fputs` permite escribir
 - caracteres en un archivo binario
 - cadena de caracteres en un archivo de texto
 - c) caracteres en un archivo de texto**
 - cadena de caracteres en un archivo binario

Entregar esta hoja con el nombre y apellidos del alumno, DNI y grupo.

Ejercicio 2 (3 puntos; R2, R3, R4, R5, R6) Tiempo estimado: 20 minutos. Todos los ejercicios tienen la misma puntuación.

11. Dado un array bidimensional char A[20][10]; escribe el código necesario para inicializar todos los valores del array con la letra j. Define todas las variables que necesites.




12. Cuando se ejecuta el siguiente código, el usuario cuando el programa le pide entrada escribe en la consola "6 8 4 5 6" y pulsa enter. ¿Qué imprimirá el printf?

```
int i,j,k;
int Primero[21];
for (i=0;i<10;i++)
    Primero[i]=i-2;
scanf("%d %d",&j,&k);
for (i=j;i<=k;i++)
    printf("%d ",Primero[i]);
```

13. Escribe el valor de x en los lugares marcados por la flecha:




```
a)
#include <stdio.h>
void modifica(int *x)
{
    x++;
}

main()
{
    int x=0;
```

```
    modifica(&x);
    -----> 
    modifica(&x);
    -----> 
    x++;
    -----> 
}
```

```
b)
#include <stdio.h>
void modifica(int *a)
{
    *a++;
}

main()
{
    int x=0;
```

```
    modifica(&x);
    -----> 
    modifica(&x);
    -----> 
    x++;
    -----> 
}
```

14. Escribe el valor de x en los lugares marcados por la flecha:

```
a)
#include <stdio.h>
void modifica(int x)
{
    x++;
```

```
b)
#include <stdio.h>
void modifica(int *a)
{
    a++;
```

<pre> } main() { int x=0; modifica(x); -----> modifica(x); -----> x++; -----> } </pre>	<pre> } main() { int x=0; modifica(&x); -----> modifica(&x); -----> x++; -----> } </pre>
--	--

15. Escribe el código necesario para abrir un fichero llamado "entrada.txt" para sólo lectura. Define todas las variables que sean necesarias. Muestra un error por pantalla en caso de que el fichero que se pretende abrir no exista.

16. Dada la siguiente estructura, y la siguiente variable:

```

struct fecha{
int mes;
int día;
int año;
};
struct cuenta{
int no_cuenta;
char tipo_cuenta;
char nombre[80];
float saldo;
struct fecha ultimopago;
};

struct cuenta x1;

```

Escribe el código necesario para escribir la variable x1 en binario en el archivo apuntado por el manejador de fichero "FILE* fichero;"; el manejador ya se encuentra inicializado.

17. Escribir código necesario para guardar en el mismo fichero que en el caso anterior, también en binario, el siguiente array: "struct cuenta clientes [100]";

18. Escribe el código necesario para leer de un fichero los datos que has almacenado en el ejercicio anterior. Puedes suponer que el apuntador del fichero (FILE* fichero;) se encuentra justo antes de los datos que tienes que leer; sólo tienes que leerlos. Carga los datos en el siguiente array: "struct cuenta clientes [100]";
19. ¿Cómo leerías una cadena de caracteres que puede contener espacios en blanco en la variable "char nombre [100]" asegurándote de que en ninguna circunstancia se van a leer más datos de los que cogen en dicha variable? (Poner el código fuente).
20. Dada la siguiente estructura y la variable "a" ¿cuál es la dirección de memoria del campo "y" de dicha variable?.

```
struct dato
{
int x;
float y;
};
struct dato a;
```



Ejercicio 3 (1.5 Puntos; R2, R3, R5, R6) Tiempo estimado: 15 minutos.

Escribe un módulo que permita realizar operaciones con vectores en tres dimensiones. El módulo deberá permitir sumar, restar, y realizar el producto escalar y el producto vectorial de dos vectores. Para cada una de estas operaciones se creará una función diferente en el módulo; dicha función deberá tomar siempre como argumentos los dos vectores sobre los que va a operar, y deberá devolver el resultado de la operación (otro vector o, cuando corresponda, un escalar) sin mostrar nada en la consola. Exporta la interfaz del módulo mediante un archivo de cabecera. Todas las funciones del módulo deberán ser llamadas desde otro módulo principal, que podrá invocar dichas funciones con cualquier valor para los parámetros. Los vectores deberán representarse mediante la siguiente estructura:

```
struct complejo{
float x;
float y;
float z;
}
```


Ejercicio 4 (2.0 puntos; R2, R3, R5, R6) Tiempo estimado: 15 minutos.

Escribir un programa que multiplique dos matrices. Sus dimensiones deben solicitarse al usuario por teclado y deberá reservarse espacio dinámicamente para ellas. Las matrices no tendrán que ser cuadradas. Deberá verificarse que las dimensiones de las matrices permiten su multiplicación, y en caso contrario se mostrará un mensaje de error al usuario. A continuación deberá leerse el contenido de las matrices. Después deberá realizarse la multiplicación y tras realizar la multiplicación debe visualizarse en pantalla ambas matrices y el resultado de la multiplicación. Deberá liberarse la memoria dinámica empleada. Úsense funciones cuando sea oportuno.

Ejercicio 5 (1.5 puntos; R2, R3, R4, R5, R6) Tiempo estimado: 15 minutos.

Escribe un programa que lea una cadena de caracteres (la primera línea) del archivo "datos.txt" e indique si es o no palíndroma (se lee igual de izquierda a derecha que de derecha a izquierda, sin tener en cuenta los espacios en blanco y las mayúsculas, por ejemplo "dábale arroz a la zorra el abad"). Sólo deberá hacerse esto con la primera línea del archivo. Deberá asegurarse que realmente se puede leer el dato del archivo y no se ha producido ningún error en la apertura de éste.

Ejercicio 6 (1 puntos; R2, R3, R4, R5, R6) Tiempo estimado: 15 minutos.

Escrebe un programa que pida el nombre de dos archivos al usuario y copie el contenido del primero en el segundo. Si se desea, puede suponer que el archivo original contiene solo texto. Debe gestionarse la situación en la cual no se pueda abrir el archivo original porque, por ejemplo, no existe, o la situación en la cual no sea posible crear el archivo copia.