Duración: 3.5 horas

Resultados de aprendizaje que se evalúan en este examen:

R1: Utilizar herramientas de programación como compiladores, depuradores y entornos de desarrollo
R2: Saber emplear las estructuras de control de flujo de programación para implementar algoritmos sencillos.
R3: Utilizar memoria RAM estática y dinámica para el almacenamiento de datos a través de estructuras de datos sencillas como arrays y estructuras.
R4: Utilizar un sistema de archivos para el almacenamiento persistente de información.
R5: Comprender el paradigma de la programación estructurada y diseñar programas que siguen dicho paradigma.
R6: Saber implementar una solución a un problema sencillo de ingeniería empleando un lenguaje de programación estructurado.

Ejercicio 1 (1.0 puntos; R1; R2, R3, R4, R5, R6) Tiempo estimado: 15 minutos. Las respuestas acertadas suman 1 punto y las falladas restan 0.5 puntos. Las que no se respondan no restarán nada.

1. El bus del sistema
   a) proporciona persistencia para los datos
   b) es un elemento opcional en una arquitectura computacional
   c) se comunica con los dispositivos de entrada y salida a través de los controladores
   d) ninguna de las anteriores
2. 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?
   a) 4 bytes
   b) 6 bytes
   c) 7 bytes
   d) ninguna de las anteriores
3. ¿Quién realiza el enlace de los distintos módulos que componen un programa?
   a) la CPU
   b) el sistema operativo
   c) el compilador
   d) ninguna de las anteriores
4. La memoria principal
   a) es más lenta que el disco duro
   b) es más lenta que los registros
   c) es más lenta que los registros
   d) ninguna de las anteriores
5. La memoria ROM
   a) es lo mismo que la memoria RAM
   b) proporciona acceso aleatorio
   c) su contenido no se pierde cuando se apaga el ordenador
   d) ninguna de las anteriores
6. La función fputs permite escribir
   a) caracteres en un archivo binario
   b) cadenas de caracteres en un archivo de texto
   c) caracteres en un archivo de texto
   d) cadenas de caracteres en un archivo binario
7. En C ¿se puede mezclar dentro de una función instrucciones y declaraciones de variables?
   a) no
   b) depende del tipo de función
   c) sólo en C99
   a) ninguna de las anteriores
8. Escribe la salida que efectúa el siguiente trozo de código.

```
char cadena[]="Pedro\n";

printf("%d",strlen(cadena));
```

9. La instrucción fseek (f, 200, SEEK_CUR); desplazará el apuntador del manejador de fichero representado por f:

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

a) 200 bytes desde la posición actual
b) 200 multiplicado por el tamaño del dato almacenado en el fichero desde la posición actual
c) 200 bytes desde el principio del archivo
d) 200 bytes desde el final del archivo

10. El comando "gcc –c"
    a) compila el código, pero no lo enlaza
    b) enlaza el código, pero no lo compila
    c) enlaza el código y lo compila
    d) preprocesa el código

Ejercicio 2 (2.0 puntos; R1; **R2, R3, R4, R5, R6) Tiempo estimado: 25 minutos**. Todos los ejercicios tienen la misma puntuación.

1. Enumera las diferencias entre la primera y la segunda generación de ordenadores.

2. Define brevemente qué es la programación estructurada.

3. Explica el proceso de traducción a código máquina del código fuente de un lenguaje de programación compilado. Idealmente apóyate en diagramas. Indica que herramientas se usa en cada paso.

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

4. 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++;

------------------------------------------->

 }
```

5. Escribe el código necesario para abrir un fichero llamado entrada.txt para añadir datos al final. Define todas las variables que sean necesarias. Muestra un error por pantalla en caso de que el fichero que se pretende abrir no exista.

6. Escribir el código necesario para leer 90 datos tipo float de un archivo binario cuyo manejador de fichero es "f". El manejador de fichero ya está inicializado. Define las variables que necesites.

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

7. Escribir el código necesario para leer 20 datos tipo int, separados por espacios en blanco, de un archivo de texto cuyo manejador de fichero es f. Define las variables que necesites.

8. ¿Cuánto vale el número hexadecimal F37 en binario?:

9. ¿Cuánto vale el número octal 137 en decimal?:

10. ¿Si ejecutamos la sentencia

    int i = ‘a’+’A’+’9’+’:’ ;

    ¿Cuál será el valor de i?:

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

Tabla ASCII
128 caracteres

D / I	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	0	1	2	3	4	5	6	7	8	9	***LF*** 10	11	12	***CR*** 13	14	15
0001	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0010	***SP*** 32	! 33	“ 34	# 35	$ 36	% 37	& 38	‘ 39	(40	) 41	* 42	+ 43	´ 44	- 45	. 46	/ 47
0011	**0** 48	**1** 49	**2** 50	**3** 51	**4** 52	**5** 53	**6** 54	**7** 55	**8** 56	**9** 57	: 58	; 59	< 60	= 61	> 62	? 63
0100	@ 64	**A** 65	**B** 66	**C** 67	**D** 68	**E** 69	**F** 70	**G** 71	**H** 72	**I** 73	**J** 74	**K** 75	**L** 76	**M** 77	**N** 78	**O** 79
0101	**P** 80	**Q** 81	**R** 82	**S** 83	**T** 84	**U** 85	**V** 86	**W** 87	**X** 88	**Y** 89	**Z** 90	[91	\ 92	] 93	^ 94	_ 95
0110	` 96	**a** 97	**b** 98	**c** 99	**d** 100	**e** 101	**f** 102	**g** 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111
0111	**p** 112	**q** 113	**r** 114	**s** 115	**t** 116	**u** 117	**v** 118	**w** 11	**x** 120	**y** 121	**z** 122	{ 123	| 124	} 125	~ 126	127

Caracteres 0-31: no imprimibles (p.e. 10: LF, fin de línea; 13:CR, “retorno de carro”)
Caracter 32: “Espacio en blanco” (SP)
Caracter 127: no imprimible (DEL)

Ejercicio 3 (1.5 puntos; R2, R3, R5, R6, R7) Tiempo estimado: 20 minutos.

Escribe un módulo que permita realizar operaciones con números racionales, esto es, números de la forma a/b donde tanto a como b son números enteros. El módulo deberá permitir sumar, restar, multiplica, dividir y simplificar números racionales. Para cada una de estas operaciones se creará una función diferente en el módulo; dicha función deberá tomar siempre como argumentos los dos números racionales sobre los que va a operar, y deberá devolver el resultado de la operación (siempre otro número racional) 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 de parámetros. Deja claro el nombre del archivo en el cual va cada parte del código que escribas. Los números racionales deberán representarse mediante la siguiente estructura:

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

```
struct racional{
        int a;
        int b;
}
```

Ejercicio 4 (1.25 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. Úsese funciones cuando sea oportuno.

Ejercicio 4 (0.75 puntos; R2, R3, R5, R6) Tiempo estimado: 10 minutos.

Escribe una función en C con nombre cadenasinblancos (siguiendo el prototipo que se detalla abajo) que, dada una cadena de caracteres, genere una nueva cadena igual a la original pero sin espacios en blanco. La nueva cadena debe ser generada dinámicamente pidiendo el mínimo número de caracteres posible.

```
char *cadenasinblancos(char *cadena)
{
.......
}
```

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

Ejercicio 5 (3.5 puntos; R2, R3, R5, R6) Tiempo estimado: 60 minutos.

Una empresa que se dedica a la cría de ratones transgénicos para su posterior venta a laboratorios necesita una aplicación que les ayude a recoger información relativa a distintas poblaciones de ratones que cría en sus instalaciones. La aplicación debe permitir llevar un registro de los ratones y las características de los ratones que forman parte de una población, población que finalmente será vendida como un todo a un laboratorio.

Para cada población de ratones, inicialmente el científico indicará: un nombre para la población y el nombre de la persona de la compañía responsable de dicha población. Para cada ratón que forma parte de la población deberá poder indicarse un código de referencia único para cada ratón (un campo de texto), su peso (en gramos, un número entero), su sexo (sólo podrá tomar valores "Macho" y "Hembra"), y su temperatura corporal en grados centígrados (un número real). Cada población de ratones se almacenará en un archivo independiente en el disco duro. La aplicación sólo tiene que permitir gestionar una única población de ratones de modo simultáneo; para trabajar con diferentes poblaciones de ratones es necesario cargar un archivo diferente (o crear una nueva población), perdiéndose los datos del anterior.

La aplicación deberá tener un menú principal con las siguientes opciones:

1. Abrir un archivo que contenga una población de ratones
2. Crear una nueva población de ratones
3. Añadir un nuevo ratón a una población ya existente
4. Listar los códigos de referencia de todos los ratones de una población
5. Eliminar un ratón de una población indicando su código de referencia
6. Guardar la población de ratones a un archivo; se supone que previamente se ha abierto el archivo o se ha usado la opción 7
7. Guardar como: se solicitará al usuario el nombre del archivo donde desea guardar la población y la guardará.

Nota: la práctica deberá organizarse **al menos en dos módulos**. Uno se encargará de realizar las operaciones con las poblaciones de ratones y tendrá una interfaz que exportará a través de un **archivo de cabecera**. El segundo módulo se encargará de la interfaz del usuario. Se valorará el respetar adecuadamente las responsabilidades de cada módulo.

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