—NO SE ADMITIRÁ NINGÚN EXAMEN ESCRITO A LÁPIZ—

—SE PUEDE ESCRIBIR POR AMBAS CARAS DE CADA HOJA DE EXAMEN—

TEST (1.5 puntos): (Se contesta en hojas aparte)

1. Indicad cuál de las siguientes afirmaciones es cierta:
   a) El lenguaje ensamblador es aquel que entiende directamente el ordenador, ya que sus instrucciones son secuencias binarias.
   b) El lenguaje máquina es el utilizado directamente por los programadores, siendo portables los programas desarrollados.
   c) Un compilador es un programa encargado de traducir un programa fuente, escrito en un lenguaje de alto nivel, a código objeto escrito en lenguaje máquina.
   d) A la hora de compilar un programa, la fase de compilación y la de ejecución se llevan a cabo al mismo tiempo.

2. Referente a las unidades de información, podemos afirmar que:
   a) El *byte* es la unidad mínima de información.
   b) 1 *bit* equivale a 8 *bytes*.
   c) La información en un ordenador está codificada en base 10.
   d) 1024 *Mbytes* = 1 *Giga-byte* = 2^{30} *bytes*.

3. Indicad cuál de las siguientes afirmaciones es cierta:
   a) La unidad aritmético-lógica opera con los datos siguiendo las indicaciones de la unidad de control.
   b) La unidad de control opera con los datos siguiendo las indicaciones de la unidad aritmético-lógica.
   c) La unidad aritmetico-lógica es el generador de las señales temporizadas que marcan las fases en la ejecución de una instrucción dentro del procesador.
   d) Los resultados de las operaciones aritméticas se almacenan en unos dispositivos de almacenamiento llamados registros ubicados en la memoria central.

4. Si hablamos de periféricos podemos decir que:
   a) El tiempo de acceso a un periférico de almacenamiento siempre es menor que el de acceso a la memoria central.
   b) Una de las razones por las que surgen los dispositivos de almacenamiento secundario es porque la memoria central es volátil y de tamaño reducido.
   c) La impresora es un ejemplo de dispositivo de almacenamiento secundario.
   d) Todo aquello que en un ordenador está dentro del sistema central se agrupa bajo el nombre de periféricos.

5. ¿Cuál de las siguientes afirmaciones es cierta ?
   a) El bus y la memoria central guardan la información que no cabe en la CPU.
   b) El disco duro guarda los datos de los programas que están en ejecución.
   c) La memoria principal guarda la información más utilizada recientemente, evitando así accesos innecesarios a la memoria caché.
   d) Los puertos de tipo paralelo del ordenador transmiten más de 1 bit en cada transacción.

6. ¿Cuál de las siguientes afirmaciones es falsa a la hora de hablar de un sistema operativo?
   a) Es un conjunto de programas que no resulta imprescindible para que funcione un ordenador.
   b) Asigna los recursos necesarios (CPU, memoria, etc.) a los programas en ejecución.
   c) Se encarga de comunicar al ordenador con los dispositivos conectados a él mediante los manejadores de dispositivos o *drivers*.
   d) Se encarga de gestionar el espacio libre en un disco duro, permitiendo que los procesos guarden información en archivos.

PROBLEMAS: (Se contestan en hojas aparte)

1. (1.5 puntos)

   Explica qué hace el siguiente programa en no más de 4 líneas.

```
#include <stdio.h>

int main()
{
    int     X, Y, Z;

    for( X=1; X <= 9; X++ )
        for( Y=0; Y <= 9; Y++ )
            for( Z=0; Z <= 9; Z++ )
                if ( (X != Y) && (X != Z) && (Y != Z) )
                    printf( "%d%d%d\n", X, Y, Z );

    return 0;
}
```

   NOTA: No se pide la traza ni que se expliquen los bucles, sino qué muestra por pantalla.

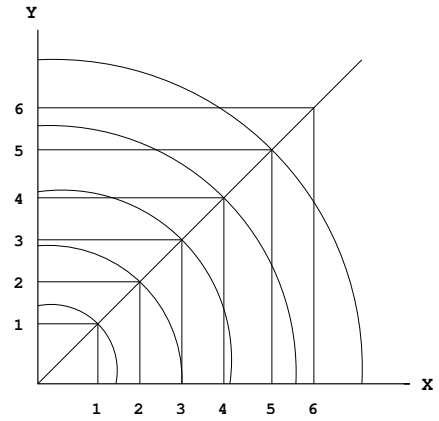
2. (1.0 puntos)

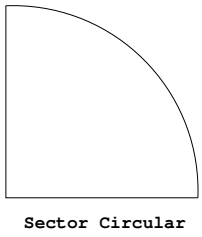
   Completad el siguiente programa para que se evalúe la función $f(x) = x^5 - 4x^4 + 3x^3 - 2x^2 + x - 10$

```
#include <stdio.h>

float f( float x )
{

    /* Aquí tienen que ir las instrucciones necesarias que se os piden. */

}

int main()
{
    float   x;

    for( x = -10.0 ; x <= 10.0; x += 0.5 ) {
        printf( " f( %.2f ) = %.2f \n", x, f(x) );
    }
    return 0;
}
```

3. (2.0 puntos)

Escribid un programa en **C** que calcule el área de los **sectores circulares** limitados por los ejes X e Y y las sucesivas circunferencias que tienen el centro en el punto (0,0) y pasan por la diagonal en los puntos (1,1), (2,2), (3,3), etc. hasta la que pase por el punto (`n`,`n`). El valor de `n` será solicitado al usuario.





Información adicional: el área de un círculo es πr^2.

4. (4.0 puntos)

Realizad un programa en lenguaje C que se encargue de gestionar la liga de fútbol de 1ª división. El programa cargará nada más comenzar su ejecución todos los resultados de un fichero denominado `Liga.txt`, y los guardará sobre una matriz interna de partidos en la que las filas hacen referencia a los equipos locales y las columnas a los visitantes. En cada posición de la matriz se almacenará un número (1, 2 ó 3), cuya codificación se muestra en el programa incluido más adelante, indicando el resultado del partido. Inicialmente todas las posiciones de la matriz deberán contener un 0, indicando que el partido todavía no se ha jugado.

El fichero de texto `Liga.txt` contiene en cada línea la siguiente información: nº de equipo local, nº de equipo visitante y un número (1, 2 ó 3) indicando el resultado. Por ejemplo:

```
0 1 1
0 2 1
. . .
. . .
```

El programa presentará las siguientes opciones:

1. Introducir/modificar un resultado.
2. Puntuación de todos los equipos.
3. Campeón de la liga hasta el momento.
4. Salir.

En caso de que el usuario elija la opción 1 (Introducir/modificar un resultado), el computador deberá solicitar el número del equipo local, el número del visitante y un número con el resultado (0=NO JUGADO, 1=GANÓ EL LOCAL, 2=GANÓ EL VISITANTE y 3=EMPATE). En caso de que ya exista un resultado para esa combinación de equipos (resultado distinto de NO JUGADO), se deberá preguntar al usuario si desea actualizar el resultado. Si la respuesta es afirmativa el nuevo valor sustituirá al anterior. Debemos tener en cuenta que un equipo no puede enfrentarse a sí mismo.

Si el usuario opta por la opción 2 (puntuación de todos los equipos) se deberá mostrar por pantalla, para cada equipo, su nombre, su código y la puntuación conseguida.

Para calcular la puntuación de un equipo se debe de tener en cuenta que una victoria cuenta como 3 puntos, un empate como 1 y que la derrota no puntúa: puntos = 3*Ganados+1*Empatados.

En caso de elegir la opción 3 (campeón de la liga hasta el momento), se debe mostrar el nombre del equipo que hasta ahora ha conseguido más puntos y su código. Si son varios los equipos que consiguen la misma puntuación máxima se elegirá cualquiera de ellos como ganador de la liga.

Al finalizar el programa, tras pulsar la opción de salir, el programa deberá guardar los datos actualizados en el mismo fichero, sustituyendo a todos los que hubieran en dicho fichero.

Aclaraciones:

El programa dispondrá de un vector de cadenas de caracteres llamado `Equipos` donde estará almacenado el nombre de cada equipo. El código de un equipo se corresponde con la posición que ocupa en dicho vector.

Cada una de las opciones del programa y el menú se implementarán en funciones independientes.

No se permiten otras variables globales que no sean las que se indican en el comienzo del programa que se presenta a continuación.

Programa:

```
#include <stdio.h>

/* Resultados posibles */
#define NO_JUGADO     0 /* El encuentro todavía no se ha jugado */
#define GANADO_LOCAL  1 /* El partido se ha jugado y ha ganado el equipo local */
#define GANADO_VSTE   2 /* El partido se ha jugado y ha ganado el equipo visitante */
#define EMPATE        3 /* El partido se ha jugado y han empatado */

#define TOTAL_EQUIP 20
char Equipos[TOTAL_EQUIP][] = { "Valencia", "F.C. Barcelona", "Sevilla", ...};

int main()
{


}
```