

TEST (1.5 puntos). Los aciertos cuentan como 0.3 puntos. 3 fallos descuentan 1 acierto.

1. Indicar cuál de las siguientes funciones es una tarea del sistema operativo
  - a. La reproducción de ficheros de audio
  - b. La gestión de memoria \*\*\*\*\*
  - c. El navegador de internet
  - d. La gestión del sistema de citas de la agenda electrónica
  
2. Indicar cuál de las siguientes afirmaciones es VERDADERA
  - a. El esquema general de funcionamiento de un programa consiste siempre en dos únicas partes: entrada de datos y proceso de los resultados
  - b. Los programadores escriben los programas en código binario
  - c. Los lenguajes de alto nivel son aquellos en los que las instrucciones se expresan según las particularidades de cada CPU
  - d. Un lenguaje de programación es un conjunto de reglas, símbolos y palabras especiales utilizados para construir un programa \*\*\*\*\*
  
3. Indicar la opción correcta:
  - a. Los ordenadores procesan la información en formato analógico
  - b. Con n bits se pueden representar n\*2 valores diferentes
  - c. 1 Kbyte = 8192 bits \*\*\*\*\*
  - d. El sistema más extendido para el almacenamiento de caracteres en un ordenador es el ASCII, donde se usan 6 bits para cada carácter
  
4. Dado el siguiente bucle, `for( t=0.0; t < 100.0; t+=0.5 ) { }`, indicar cual de los siguientes deja en la variable 't' el mismo resultado que el anterior tras finalizar su ejecución
  - a. `for( t=-0.0; t <= 100.0; t= t + 0.5 ) { }`
  - b. `t=-0.0; while(t <= 100.0) { t= t + 0.5 }`
  - c. `t=-11.5; do { t= t + 0.5 } while (t < 100.0);` \*\*\*\*\*
  - d. `for ( t=100.0; t < 100.5; t= t + 0.5 ) { }`
  
5. ¿Puede un programa escrito en lenguaje C ejecutarse sin contener la función main() ?
  - a. Sólo cuando se ejecuta dentro del entorno Linux. En Windows, sí.
  - b. No, nunca \*\*\*\*\*
  - c. Sí, pero sólo cuando contiene la cláusula #include
  - d. Sí, pero sólo cuando no contiene la cláusula #define

## PROBLEMAS



1.-

```
/*Programa de gestión de los juegos del Mediterráneo de Almería*/

#include <stdio.h>

#define VERDADERO -1
#define FALSO 0

#define C 100      /*Total de carreras existentes*/
#define D 1000     /*Total de deportistas inscritos*/

/*Opciones de menú*/
#define GANADOR_CARRERA 1
#define TIEMPO_MEDIO 2
#define GANADOR_COMPETICION 3
#define SALIR 4

#define OP_GANADOR "Ganador de una carrera"
#define OP_T_MEDIO "Tiempo medio empleado por un corredor"
#define OP_GANADOR_TOTAL "Ganador de toda la competición"
#define OP_SALIR "Salir"

void LeerTiempos (float Tiempos[C][D]);

/*Función que muestra por pantalla las opciones que presenta el programa y
devuelve la opción seleccionada por el usuario*/

int Menu ()
{
    int opcion;
    do {
        printf("\n\nPor favor, introduzca la opción de programa que desea
ejecutar:\n");
        printf("\t%d .- %s\n", GANADOR_CARRERA, OP_GANADOR);
        printf("\t%d .- %s\n", TIEMPO_MEDIO, OP_T_MEDIO);
        printf("\t%d .- %s\n", GANADOR_COMPETICION, OP_GANADOR_TOTAL);
        printf("\t%d .- %s\n", SALIR, OP_SALIR);
        scanf ("%d", &opcion);
    } while (opcion<GANADOR_CARRERA || opcion>SALIR);
    return opcion;
}

/*Función que averigua el corredor que ha ganado una carrera determinada.
Para ello determina cual es el que ha empleado menor tiempo en realizarla*/
int GanadorCarrera ( float Tiempos[C][D], int Carrera)
{
    int Ganador, Corredor;

    /*Se recorren todos los tiempos obtenidos por todos los corredores*/
    for (Corredor = 1, Ganador = 0;
        Corredor<D;
        Corredor++)
        if (Tiempos[Carrera][Corredor]<Tiempos[Carrera][Ganador]) Ganador =
Corredor;
```

```
        return Ganador;
    }

/*Calcula el tiempo medio que ha invertido un corredor en recorrer todas las
carreras de la competición*/
float TiempoMedio (float Tiempos[C][D], int Corredor)
{
    int Carrera;
    float Tiempo;

    /*Se recorren todos los tiempos obtenidos por dicho corredor*/
    for (Carrera = 0, Tiempo = 0.0;
        Carrera<C;
        Carrera++)
        Tiempo += Tiempos[Carrera][Corredor];
    return Tiempo/C;
}

/*Averigua cual es el corredor que menos tiempo ha invertido en recorrer todas
las pruebas*/

int GanadorAbsoluto (float Tiempos[C][D])
{
    int Corredor, Ganador;
    float MejorTiempo, Tiempo;

    for (Corredor=1, Ganador = 0, MejorTiempo = TiempoMedio(Tiempos, Ganador);
        Corredor<D;
        Corredor++)
    {
        Tiempo = TiempoMedio(Tiempos, Corredor);
        if (MejorTiempo>Tiempo) {MejorTiempo=Tiempo; Ganador = Corredor;}
    }
    return Ganador;
}

int main()
{
    float Tiempos[C][D];
    int Corredor, Opcion, Carrera, continuar = VERDADERO;
    char c;

    /* Inicialización de variables */

    /* Entrada de datos */
    do {
        switch (Menu())
        {
            case GANADOR_CARRERA:
                do {
                    printf("Indique por favor la carrera para la cual quiere mostrar el
ganador: ");
                    scanf ("%d", &Carrera);
                } while (Carrera<1 || Carrera>C);
                printf ("El deportista que ha ganado la carrera nº %d es el dorsal nº:
%d\n", Carrera, GanadorCarrera (Tiempos, Carrera-1)+1);
                printf ("Pulse una tecla para continuar");
                scanf("%c", &c);break;
            case TIEMPO_MEDIO:
                do {
```

```
        printf("Indique por favor el corredor para el cual quiere mostrar
su tiempo medio: ");
        scanf ("%d", &Corredor);
    } while (Corredor<1 || Corredor>D);
    printf ("El tiempo medio empleado por el deportista cuyo nº de dorsal
es %d es %f s.\n", Corredor, TiempoMedio (Tiempos, Corredor-1));
    printf ("Pulse una tecla para continuar");
    scanf("%c", &c);break;
    case GANADOR_COMPETICION:
        printf ("El ganador de todo los juegos es el deportista cuyo nº de
dorsal es %d\n.", GanadorAbsoluto(Tiempos)+1);
        printf ("Pulse una tecla para continuar");
        scanf("%c", &c);break;
    case SALIR: continuar = FALSO; break;
    default: printf ("Opción no válida. Por favor vuelva a pulsar de nuevo la
opción deseada entre %d y %d\n", GANADOR_CARRERA, SALIR);
    } /*Del switch*/
} while (continuar);
return 0; /*Final de programa*/
}
```

```
2.- #include <stdio.h>

int main()
{
    int edad, personas, hombres, mujeres, hombres_menores_18, mujeres_mayores_50,
    suma_hombres, suma_mujeres;
    char sexo;
    float media_hombres, media_mujeres;

    /* Inicialización de variables */
    hombres=0;
    mujeres=0;
    hombres_menores_18=0;
    mujeres_mayores_50=0;
    suma_hombres=0;
    suma_mujeres=0;

    /* Entrada de datos */
    do {
        printf("\nIntroduzca la edad: "); scanf("%d", &edad);
        if ( edad >= 0 ) {
            do {
                printf("Introduzca el sexo (H/M): "); scanf("\n%c", &sexo);
            } while ( sexo!='H' && sexo!='M' );

            if ( sexo == 'H' ) {
                hombres++;
                if ( edad < 18 ) hombres_menores_18++;
                suma_hombres += edad;
            } else {
                mujeres++;
                if ( edad >= 50 ) mujeres_mayores_50++;
                suma_mujeres += edad;
            }
        }
    } while ( edad >= 0 );

    personas = hombres + mujeres;
    /* Cálculo de las medias */
    if ( hombres > 0 )
        media_hombres = (float)suma_hombres / hombres;
    else
        media_hombres = 0.f;

    if ( mujeres > 0 )
        media_mujeres = (float)suma_mujeres / mujeres;
    else
        media_mujeres = 0.f;

    /* Salida por pantalla */
    printf("\nEn este grupo hay %d personas, de las cuales:\n", personas);

    printf(" Hombres: %d , menores de edad: %d , media: %.2f años\n",
        hombres, hombres_menores_18, media_hombres);

    printf(" Mujeres: %d , mayores de 50: %d , media: %.2f años\n\n",
        mujeres, mujeres_mayores_50, media_mujeres);

    return 0;
}
```

```
3.- #include <stdio.h>
#include <string.h>

#define TRUE 1
#define FALSE 0
#define N 31

void conversor (char unidad1[N], char unidad2[N])
{
    FILE *f;
    int unidades, i, encontrado=FALSE, encontrado_uno=FALSE;
    float factor;
    char u1[N], u2[N];

    if ((f=fopen("ConvUni.txt", "r"))==NULL)
        printf("Error apertura del fichero\n");
    else
    {
        fscanf(f, "%d", &unidades);
        i=0;
        while (i<unidades && !encontrado)
        {
            fscanf(f, "%s%f%s", u1, &factor, u2);
            if (strcmp(u1, unidad1)==0)
            {
                encontrado_uno=TRUE;
                if (strcmp(u2, unidad2)==0)
                    encontrado=TRUE;
            } /*Del if*/
            i++;
        } /*Del While*/
        if (encontrado)
            printf("Un %s equivale a %f %s.\n\n", unidad1, factor, unidad2);
        else
        {
            if (encontrado_uno)
                printf("No se dispone de la conversión de %s a %s.\n\n", unidad1,
unidad2);
            else
                printf("Unidad %s no disponible para convertir.\n\n", unidad1);
            fclose(f);
        }
    }
}

void main()
{
    int fin=FALSE;
    char unidad1[N], unidad2[N];

    do
    {
        printf("Primera unidad: ");
        scanf("%s", unidad1);
        printf("Segunda unidad: ");
        scanf("%s", unidad2);
        if (strcmp(unidad1, "fin")==0 || strcmp(unidad2, "fin")==0)
            fin=TRUE;
        else
            conversor (unidad1, unidad2);
    }
    while (!fin);
}
```