

Aula		Fila		Col.	
Nombre		1 ^{er} Ap.		2 ^o Ap.	
					Firma:

NOTAS DE OBLIGATORIA LECTURA

1. Las soluciones de examen entregadas escritas a lápiz, pueden ser anuladas.
2. Para la realización de este examen es obligatoria la presentación de identificación oficial actualizada con fotografía: carné de estudiante, DNI, pasaporte o permiso de conducir formato tarjeta.
3. El examen de tipo test se entrega por separado. Todas las preguntas de desarrollo se contestan en hoja aparte y en cualquier orden. Pueden contestarse varias preguntas en la misma hoja.
4. Esta prueba durará 15 minutos como máximo.

TEST (1.5 puntos). Los aciertos cuentan como 0.25 puntos. 1 fallo descuenta 1/3 de acierto.

1. **¿Qué es un proceso para un sistema operativo?**
 - a) Un programa en ejecución.
 - b) Cada una de las tareas necesarias en la compilación de un código fuente.
 - c) El programa controlador de un dispositivo hardware.
 - d) Cada función que aparece en un programa en C.
2. **¿Cuál de las siguientes listas de dispositivos presenta una velocidad de transferencia de datos DECRECIENTE?**
 - a) Memoria Cache – memoria RAM – Disco duro
 - b) Disco duro – Disquete 1.44MB – memoria RAM
 - c) Disco duro – CD – Memoria Cache
 - d) Disquete 1.44MB – CD ROM – Memoria RAM
3. **¿Cuál es el rango de un "unsigned short" de C (16 bits)?**
 - a) Desde -32768 a +32767
 - b) Desde 1 a 65536
 - c) Desde 0 a 65535
 - d) Desde 0 a 255
4. **¿Cuál de los siguientes no es un tipo de memoria basada en microchips?**
 - a) DVD-ROM.
 - b) Pendisk, memoria USB, USB stick,...
 - c) RAM.
 - d) ROM.
5. **¿Cuántos bits hacen falta para una variable en la que se quiere poder almacenar 10 valores diferentes?**
 - a) 2
 - b) 3
 - c) 4
 - d) 5
6. **¿Cuál de las siguientes instrucciones de C es correcta?**
 - a) int edad; scanf("%d",edad);
 - b) float nota; scanf("%f",nota);
 - c) char nombre[20]; scanf("%s",nombre);
 - d) char sexo; scanf("%c",sexo);

Respuestas

1	A	2	A	3	C	4	A	5	C	6	C
---	---	---	---	---	---	---	---	---	---	---	---

ENUNCIADOS

1. (1.5 puntos) La letra del NIF se obtiene tomando el residuo o resto de la división entera del número del DNI entre 23 y accediendo a la posición resultante en la tabla siguiente:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

Así, si el DNI es el 25.879.963, al dividirlo por 23, el resto es 18. La posición 18 de la tabla de traducción anterior es la letra 'H', por lo tanto, el NIF asociado al DNI anterior es el 25.879.963H.

Hacer un programa que calcule la letra del NIF a partir del número del DNI. Se pide que el alumno realice un programa principal encargado de

- Pedir al usuario el número del DNI.
- Pasar el número obtenido a la función "Letra", la cual devolverá una única letra a partir del número entero largo sin signo que se le pase como parámetro. Implementar también la función. (0.75 ptos.)
- Imprimir por pantalla un mensaje al usuario en el que le indicará el NIF completo. Es decir, dicho DNI terminado en la letra asociada.

Queda terminantemente **PROHIBIDO** emplear la instrucción *switch()-case* o construcciones basadas en *if-else* anidados. Cualquier solución que los contenga será declarada nula.

Solución

```
/*Programa de cálculo de la letra del NIF*/
```

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

```
#define TOTALETTRAS 23
```

```
#define LONGMAXNIF 8
```

```
/*Función de calcula el número entero largo a partir de un vector de 8 enteros pequeños de 8 bits cada uno*/
```

```
unsigned long int Numero (unsigned char DNI[LONGMAXNIF])
```

```
{
```

```
    unsigned long int Resultado = 0;
```

```
    unsigned short int digito ;
```

```
    for (digito=0 ;digito<LONGMAXNIF ;digito++)
```

```
        Resultado = Resultado*10 + DNI[digito];
```

```
    return Resultado
```

```
}
```

```
/*Función que calcula la letra asociada al DNI correspondiente*/
```

```
char Letra (unsigned long int DNI, char Letras[TOTALETTRAS])
```

```
{return Letras[DNI%TOTALETTRAS];}
```

```
int main(void)
```

```
{
```

```
char
```

```
L,
```

```
Letras[TOTALETTRAS] = "TRWAGMYFPDXBNJZSQVHLCKE";
```

```
unsigned char
```

```
DNI[LONGMAXNIF];
```

```
unsigned short int
```

```
digito;
```

```
unsigned long int
```

```
DNI2;
```

```
/*Petición del NÚMERO DE dni*/
printf("Introduzca, por favor, los ocho dígitos del DNI. ");

for (digito=0;digito< LONGMAXNIF ;digito++)
do { printf("\nIntroduzca el dígito número %d: ", digito+1);
scanf("%c", &DNI[digito]);
}

/*Conversión del formato vector a número entero largo de 32 bits*/
DNI2 = Numero (DNI);

/*Obtención de la letra asociada al DNI*/
L = Letra(DNI2, Letras);

/*Presentación de los resultados por pantalla al usuario*/
printf ("El NIF asociado al DNI %d es %d%c", DNI2, DNI2, L);

/*Fin del programa*/
return 0;
}
```

2. (1.0 puntos) Realizar un programa que pida por teclado al usuario una matriz cuadrada de dimensión (NxN) y la escriba de nuevo en pantalla girada 90° en sentido antihorario. El programa primero tiene que pedir al usuario la dimensión de la matriz cuadrada (que no puede ser mayor que 10 ni menor que 2) y a continuación los elementos de la matriz original por filas. El programa debe imprimir a continuación por pantalla la matriz original, tal y como la ha introducido el usuario, y seguidamente la misma matriz girada 90° en sentido antihorario. A continuación se muestra un ejemplo de la salida por pantalla del programa:

Matriz original

1 2 3

4 5 6

7 8 9

Matriz girada

3 6 9

2 5 8

1 4 7

Aclaración

No es necesario girar previamente la matriz en memoria para luego imprimirla. Es suficiente con mostrarla directamente girada por pantalla. Se supone que el usuario siempre introducirá un n° comprendido entre 2 y 10 y nunca se equivocará o actuará de mala fe.

Solución

```
/* girar una matriz rectangular */
#include<stdio.h>
#define MAY 10

void main(void)
{
    int MaxFilCol;
    int i, j;
    double a[MAY][MAY];
```

```
/*Petición del tamaño de la matriz y comprobación de errores*/
do {
    printf("\nIntroduzca el número de filas: ");
    scanf("%d", &MaxFilCol);
    if(MaxFilCol>MAY || MaxFilCol<0)
    {
        printf("El número de filas debe ser menor que %d y mayor que 0 \n",
            MAY);
    }
}while (MaxFilCol>MAY || MaxFilCol<0);

/*Rellenado de la matriz por el usuario*/
for (i=0; i< MaxFilCol; i++)
    for(j=0; j< MaxFilCol; j++){
        printf("\nElemento (%d,%d): ", i+1, j+1);
        scanf("%lf", &a[i][j]);
    }

/*Escritura de la matriz original, tal y como la introdujo el usuario*/
printf("\n\nLa matriz original es: ");
for (i=0; i< MaxFilCol; i++) {
    printf("\n");
    for (j=0; j< MaxFilCol; j++)
        printf("%12.4lf ", a[i][j]);
}

/*Escritura de la matriz rotada una vez en sentido antihorario*/
printf("\n\nLa matriz girada es:\n");
for (j= MaxFilCol-1; j>=0; j--) {
    printf("\n");
    for(i=0; i< MaxFilCol; i++)
        printf("%12.4lf ", a[i][j]);
}
}
```

3. (2.0 puntos) Se dispone de un fichero denominado ALMACEN.TXT con la descripción de los productos guardados en un almacén. El fichero siempre contiene 100 líneas, correspondientes a los 100 diferentes productos con los que se trabaja. Para cada producto, y por lo tanto, en cada línea, aparecen siempre dos valores numéricos:
- El primero es el precio en euros de una unidad de ese producto.
 - El segundo número es la cantidad de unidades de ese producto que hay en el almacén.

(0.5 pts) Se pide realizar un programa en C que, a partir de la información contenida en el fichero, indique:

- (0.5 pts) El precio del producto más caro.
- (0.5 pts) La cantidad de productos sin existencias (de los que hay 0 unidades).
- (0.5 pts) El valor total en euros de todo el contenido del almacén.

Solución

```
#include <stdio.h>

#define ALUMNOS 60
#define ASIGNATURAS 10
```

```
#define MAXLEN 31

int menu()
{
    int opcion;

    printf(" 1.Introducir datos.\n");
    printf(" 2.Ver notas y media de un alumno.\n");
    printf(" 3.Ver notas de una asignatura.\n");
    printf(" 4.Ver alumnos con todo aprobado.\n");
    printf(" 5.Salir.\n");

    scanf("%d",&opcion);

    return opcion;
}

void introducir_datos(char nombres[ALUMNOS][MAXLEN],float
notas[ALUMNOS][ASIGNATURAS])
{
    int alumno,asignatura;

    printf("\nEntrada de datos\n\n");

    for( alumno = 0 ; alumno < ALUMNOS ; alumno++ ) {
        printf("\nNombre del alumno %d: ",alumno+1);
        scanf("\n"); gets(nombres[alumno]);

        printf("\nNotas: ");
        for( asignatura = 0 ; asignatura < ASIGNATURAS ; asignatura++ ) {
            scanf("%f",&notas[alumno][asignatura]);
        }
    }
}

void ver_alumno(float notas[ALUMNOS][ASIGNATURAS])
{
    int alumno,asignatura;
    float media;

    scanf("%d",&alumno);
    alumno--;

    printf("\nNotas:");
    media = 0;
    for( asignatura = 0 ; asignatura < ASIGNATURAS ; asignatura++ ) {
        printf(" %5.2f",notas[alumno][asignatura]);
        media += notas[alumno][asignatura];
    }
    media /= ASIGNATURAS;
    printf("\nNota media: %.2f\n",media);
}

void ver_asignatura(char nombres[ALUMNOS][MAXLEN],float
notas[ALUMNOS][ASIGNATURAS])
{
    int alumno,asignatura;

    scanf("%d",&asignatura);
    asignatura--;
```

```
for( alumno = 0 ; alumno < ALUMNOS ; alumno++ ) {
    printf(" %-30s %5.2f\n",nombres[alumno],notas[alumno][asignatura]);
}

void ver_aprobados(char nombres[ALUMNOS][MAXLEN],float
notas[ALUMNOS][ASIGNATURAS])
{
    int alumno,asignatura,aprobadas;

    printf("\nAlumnos con todas las asignaturas aprobadas\n");
    for( alumno = 0 ; alumno < ALUMNOS ; alumno++ ) {
        aprobadas = 0;
        for( asignatura = 0 ; asignatura < ASIGNATURAS ; asignatura++ ) {
            if ( notas[alumno][asignatura] >= 5 )
                aprobadas++;
        }
        if ( aprobadas == ASIGNATURAS )
            printf(" %s\n",nombres[alumno]);
    }
}

int main()
{
    char nombres[ALUMNOS][MAXLEN];
    float notas[ALUMNOS][ASIGNATURAS];
    int opcion;

    do {
        opcion = menu();
        switch ( opcion ) {
            case 1:
                introducir_datos(nombres,notas);
                break;
            case 2:
                ver_alumno(notas);
                break;
            case 3:
                ver_asignatura(nombres,notas);
                break;
            case 4:
                ver_aprobados(nombres,notas);
                break;
        }
    } while ( opcion != 5 );

    return 0;
}
```

4. (4.0 puntos) Desarrollar un programa completo para realizar algunas operaciones con las notas en 10 asignaturas de un grupo de 60 alumnos. El programa dispondrá del nombre completo de cada uno de los 60 alumnos y su nota para cada una de las distintas asignaturas.

Las operaciones que debe realizar el programa, mediante su elección a través de un menú, son:

1. Introducir los datos de los 60 alumnos. Se irá preguntando por el nombre completo y cada una de sus 10 notas. (0.75 puntos)

2. Dado un alumno (un número del 1 al 60), mostrar por pantalla todas sus notas y su nota media, que habrá que calcular. (0.75 puntos)
3. Dada una asignatura (un número del 1 al 10), sacar un listado con las notas de todos los alumnos en esa asignatura (para cada alumno saldrá su nombre completo y su nota). (0.75 puntos)
4. Sacar un listado con el nombre (no hay que sacar las notas) de todos los alumnos que hayan aprobado todas las asignaturas. (1 punto)
5. Salir del programa.

La impresión del menú y lectura de la opción elegida por el usuario debe hacerse en una función independiente. (Esto junto al programa principal son 0.75 puntos)

Cada opción, excepto la de "salir", debe realizarse en una función que será invocada desde el programa principal. Se recuerda que no se permite el uso de variables globales. **Se debe utilizar** el comienzo del programa principal que se detalla seguidamente y rellenar lo que haga falta.

```
#include <stdio.h>

/* Aquí faltan la definición de constantes y las funciones */

int main()
{
    char nombres[ALUMNOS][100]; /* para los nombres de cada alumno. Máximo,
100 caracteres por nombre */
    float notas[ALUMNOS][ASIGNATURAS]; /* para las notas de cada alumno */
    /* Declarar el resto de variables que hagan falta */

    /* Aquí falta llamar a la función que lee la opción del usuario
    y llamar a la función que toque en cada caso,
    repitiendo todo hasta que el usuario pulse la opción de salir */

    return 0;
}
```

Solución

```
#include <stdio.h>

#define ALUMNOS 60
#define ASIGNATURAS 10
#define MAXLEN 31

int menu()
{
    int opcion;

    printf("\n\nMenú:\n");
    printf(" 1.Introducir datos.\n");
    printf(" 2.Ver notas y media de un alumno.\n");
    printf(" 3.Ver notas de una asignatura.\n");
    printf(" 4.Ver alumnos con todo aprobado.\n");
    printf(" 5.Salir.\n");

    printf("\nElija opción: ");
    scanf("%d",&opcion);
}
```

```
    return opcion;
}

void introducir_datos(char nombres[ALUMNOS][MAXLEN],float
notas[ALUMNOS][ASIGNATURAS])
{
    int alumno,asignatura;

    printf("\nEntrada de datos\n\n");

    for( alumno = 0 ; alumno < ALUMNOS ; alumno++ ) {
        printf("\nNombre del alumno %d: ",alumno+1);
        scanf("\n"); gets(nombres[alumno]);

        printf("\nNotas: ");
        for( asignatura = 0 ; asignatura < ASIGNATURAS ; asignatura++ ) {
            scanf("%f",&notas[alumno][asignatura]);
        }
    }
}

void ver_alumno(float notas[ALUMNOS][ASIGNATURAS])
{
    int alumno,asignatura;
    float media;

    printf("\nNúmero de alumno a mostrar: ");
    scanf("%d",&alumno);
    alumno--;

    printf("\nNotas:");
    media = 0;
    for( asignatura = 0 ; asignatura < ASIGNATURAS ; asignatura++ ) {
        printf(" %5.2f",notas[alumno][asignatura]);
        media += notas[alumno][asignatura];
    }
    media /= ASIGNATURAS;
    printf("\nNota media: %.2f\n",media);
}

void ver_asignatura(char nombres[ALUMNOS][MAXLEN],float
notas[ALUMNOS][ASIGNATURAS])
{
    int alumno,asignatura;

    printf("\nNúmero de asignatura a mostrar: ");
    scanf("%d",&asignatura);
    asignatura--;

    for( alumno = 0 ; alumno < ALUMNOS ; alumno++ ) {
        printf(" %-30s %5.2f\n",nombres[alumno],notas[alumno][asignatura]);
    }
}

void ver_aprobados(char nombres[ALUMNOS][MAXLEN],float
notas[ALUMNOS][ASIGNATURAS])
{
    int alumno,asignatura,aprobadas;
```

```
printf("\nAlumnos con todas las asignaturas aprobadas\n");
for( alumno = 0 ; alumno < ALUMNOS ; alumno++ ) {
    aprobadas = 0;
    for( asignatura = 0 ; asignatura < ASIGNATURAS ; asignatura++ ) {
        if ( notas[alumno][asignatura] >= 5 )
            aprobadas++;
    }
    if ( aprobadas == ASIGNATURAS )
        printf(" %s\n",nombres[alumno]);
}

int main()
{
    char nombres[ALUMNOS][MAXLEN];
    float notas[ALUMNOS][ASIGNATURAS];
    int opcion;

    do {
        opcion = menu();
        switch ( opcion ) {
            case 1:
                introducir_datos(nombres,notas);
                break;
            case 2:
                ver_alumno(notas);
                break;
            case 3:
                ver_asignatura(nombres,notas);
                break;
            case 4:
                ver_aprobados(nombres,notas);
                break;
        }
    } while ( opcion != 5 );

    return 0;
}
```

SOLUCIÓN ALTERNATIVA

```
#include <stdio.h>

/* Aquí están la definición de constantes y las funciones */
#define ALUMNOS 60
#define ASIGNATURAS 10

#define INTRODUCIR_ALUMNOS 0 /*Introducir los datos de los alumnos*/
#define MOSTRAR_NOTAS 1 /*Mostrar las notas del alumno seleccionado*/
#define MOSTRAR_ASIGNATURA 2 /*Mostrar las notas de la asignatura
seleccionada*/
#define MOSTRAR_APROBADOS 3 /*Mostrar el listado de todos los alumnos
completamente aprobados*/
#define SALIR 4
#define ULTIMA_OPCION 5 /*La misma que SALIR*/

char TextoMenuPpal[ULTIMA_OPCION][75] = {"Introducir los datos de los
alumnos.",
```

```

notas del alumno seleccionado",
notas de la asignatura seleccionada",
listado de todos los alumnos completamente aprobados",
programa"};

unsigned char LeeOpcionUsuario ()
{
    unsigned int opcion;

    do {
        printf("Bienvenido al programa de gestión de notas.\n");
        printf("Por favor, introduzca cual de las siguientes opciones
desea realizar:\n");

        for (opcion = 0;opcion<ULTIMA OPCION;opcion++) printf ("%d.-
%s\n", opcion, TextoMenuPpal[opcion]);
        printf("\nPor favor, pulsa la opción deseada: ");
        scanf("%u", &opcion);
    } while (opcion >= ULTIMA OPCION);

    return opcion;
}

void IntroducirAlumnos (char nombres[ALUMNOS][100], float
notas[ALUMNOS][ASIGNATURAS]) /*1. Introducir los datos de los 60 alumnos.
Se irá preguntando por el nombre completo y cada una de sus 10 notas.
(0.75 puntos) */
{
    unsigned char alumno, nota;

    for (alumno = 0; alumno < ALUMNOS; alumno++)
    {
        printf("Por favor, introduzca el nombre del alumno nº %d: ",
alumno+1);
        scanf("%s", nombres[alumno]);

        printf("\n"); /*Intervalo de separación de entre la
introducción de los nombres de los alumnos y sus correspondientes notas*/
        for(nota=0;nota<ASIGNATURAS;nota++)
        {
            printf ("\nIntroduzca, por favor, la nota correspondiente a la
asignatura nº%d del alumno %s: ", nota+1, nombres[alumno]);
            scanf("%f", &notas[alumno][nota]);
        }
    }
}

MostrarNotas(float notas[ALUMNOS][ASIGNATURAS]) /*2. Dado un alumno (un
número del 1 al 60), mostrar por pantalla todas sus notas y su nota media,
que habrá que calcular. (0.75 puntos)*/
{
    unsigned char alumno, nota;
    float media = 0.0;

    printf("Por favor, indique el nº del alumno: ");
    scanf("%u", &alumno);

```

```
    printf("\nSecuencia de notas obtenidas por el alumno %u:\n", alumno);
/*Intervalo de separación de entre la introducción del número del alumno y
sus correspondientes notas*/
    for(nota=0;nota<ASIGNATURAS;nota++)
    {
        printf ("%f ", notas[alumno][nota]);
        media+=notas[alumno][nota];
    }
    printf("La nota media del alumno nº%u es %f\n", alumno,
media/ASIGNATURAS);
    printf ("Pulse una tecla para continuar");
    alumno = getchar();
}
```

```
MostrarAsignatura(char nombres[ALUMNOS][100], float
notas[ALUMNOS][ASIGNATURAS]) /*3. Dada una asignatura (un número del 1 al
10), sacar un listado con las notas de todos los alumnos en esa asignatura
(para cada alumno saldrá su nombre completo y su nota). (0.75 puntos)*/
{
    unsigned int asignatura, alumno;

    printf("Por favor, indica el nº de la asignatura a listar: ");
    scanf("%u",&asignatura);

    printf("\n\nListado de alumnos y notas de la asignatura nº %u.\n",
asignatura);
    printf("Alumno
Nota\n_____ \n\n");

    for (alumno=0;alumno<ALUMNOS;alumno++) printf("%s\t%f\n", nombres[alumno],
notas[alumno][asignatura]);

    printf("\n\nFin de listado.");
}
```

```
MostrarAprobados(char nombres[ALUMNOS][100], float
notas[ALUMNOS][ASIGNATURAS]) /*4. Sacar un listado con el nombre (no hay
que sacar las notas) de todos los alumnos que hayan aprobado todas las
asignaturas. (1 punto)*/
{
    unsigned int asignatura, alumno;

    printf("\nListado de todos aquellos alumnos que han aprobado todas las
asignaturas.\n");

    printf("_____
_____ \n");

    for (alumno=0;alumno<ALUMNOS;alumno++)
    {
        asignatura=0;
        while (asignatura<ASIGNATURAS && notas[alumno][asignatura]>=5.0)
asignatura++;
        if (asignatura==ASIGNATURAS) printf("%s\n", nombres[alumno]);
    }

    printf("\n\nFin del listado\n");
}
```

```
int main()
{
    char nombres[ALUMNOS][100]; /* para los nombres de cada alumno. Máximo,
100 caracteres por nombre */
    float notas[ALUMNOS][ASIGNATURAS]; /* para las notas de cada alumno */

    /* Declarar el resto de variables que hagan falta */
    unsigned char opcion;

    do {
        /* Aquí falta llamar a la función que lee la opción del usuario */
        opcion = LeeOpcionUsuario ();

        /*y llamar a la función que toque en cada caso...*/
        switch(opcion)
        {
            case INTRODUCIR_ALUMNOS: IntroducirAlumnos (nombres, notas);
/*Introducir los datos de los alumnos*/
                break;
            case MOSTRAR_NOTAS:      MostrarNotas(notas); /*Mostrar las notas del
alumno seleccionado*/
                break;
            case MOSTRAR_ASIGNATURA: MostrarAsignatura(nombres, notas);
/*Mostrar las notas de la asignatura seleccionada*/
                break;
            case MOSTRAR_APROBADOS:  MostrarAprobados(nombres, notas); /*Mostrar
el listado de todos los alumnos completamente aprobados*/
                break;
            case SALIR:
                break;
            default: printf ("ERROR: opción no definida. Por favor, vuelva a
intentarlo de nuevo. Gracias.\n");
        }
    } while (opcion != SALIR); /*...repetiendo todo hasta que el usuario
pulse la opción de salir.*/

    printf ("Fin de programa. Gracias por haber utilizado nuestros
servicios.\n");

    return 0;
}
```