



UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA INDUSTRIAL
Departamento E.I.A.I.

APELLIDOS

NOMBRE N° Mat.

ASIGNATURA **INFORMÁTICA**

CURSO 1º GRUPO: FECHA 11/07/2013

Calificación

Fecha de publicación de notas: jueves 18 de Julio.
Fecha de revisión: viernes 19 de Julio a las 10:00h. Despacho C-209.

Ejercicio 1 (2 puntos. Tiempo: 25 minutos)

Se desea desarrollar un programa en C que permita gestionar el saldo de una cuenta bancaria. En esta cuenta bancaria únicamente se pueden realizar dos movimientos: ingresos (I) y gastos (G). Para ello hay que programar una función que se encargue de anotar ese movimiento en la cuenta y actualizar el saldo. El prototipo de la función es el siguiente:

```
int apuntarMovimiento(float *, char, float);
```

La función devuelve 0 si el movimiento se ha anotado correctamente o -1 en caso de que la operación no se haya podido realizar (en el caso de que no haya saldo suficiente para anotar un gasto). Esta función se debe encargar de actualizar el saldo con la cantidad y el tipo de operación indicadas como parámetros.

El programa principal debe mostrar las diferentes opciones al usuario para que elija si quiere anotar un ingreso o un gasto o desea salir del programa. Hasta que el usuario no elige la opción de salir el programa se repite continuamente. El programa principal imprime el saldo resultante tras cada operación e indica si el gasto no ha podido ser realizado al no haber saldo suficiente.

Ejemplo de funcionamiento:

```
Indica la opcion: gasto G, ingreso I, salir S
G
Indica la cantidad
20
Saldo actual: 480.000000
Indica la opcion: gasto G, ingreso I, salir S
I
Indica la cantidad
30
Saldo actual: 510.000000
Indica la opcion: gasto G, ingreso I, salir S
G
Indica la cantidad
520
No hay saldo suficiente. Operacion no realizada
Saldo actual: 510.000000
Indica la opcion: gasto G, ingreso I, salir S
S
```



UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA INDUSTRIAL
Departamento E.I.A.I.

APELLIDOS	<input type="text"/>	
NOMBRE	<input type="text"/>	Nº Mat. <input type="text"/>
ASIGNATURA	INFORMÁTICA	Calificación
CURSO 1º	GRUPO:	

Ejercicio 2 (2 puntos. Tiempo: 30 minutos)

Se desea realizar un programa en C, para pasar de un código de letras y números, introducido por el usuario, a una cadena de caracteres siguiendo un determinado formato. Para facilitar la programación se programarán varias funciones. El código introducido es una secuencia de letras y dígitos (por ejemplo: M3T1K4), y el resultado final debe ser una secuencia de letras, cada letra se repite tantas veces como indique el dígito que le sigue (en ese ejemplo el resultado es: MMMTKKKK). El código siempre tiene una o varias parejas formadas de una sola letra seguida de un solo dígito, por tanto, la longitud de la cadena es siempre par. Se pide:

1. Programar una función llamada **calcular_int**, con un parámetro llamado **letra** de tipo char, y devuelve un valor int. Si el parámetro está entre los caracteres '0' y '9', devolverá su valor entero, si no, devolverá un -1.
2. Programar una función llamada **separador**, que acepta 3 parámetros: **codigo** es un vector (o cadena) de char, **letras** es un vector de char y **numeros** es un vector de int. Los 3 vectores tienen longitud desconocida. La función devolverá un valor int. Esta función va a examinar la cadena pasada en **codigo**, sabiendo que cumple con el formato de letra y dígito indicados (ej: M3T1K4), y debe ir colocando en orden las letras por un lado, en el vector **letras**, y los números (convertidos a su valor int) por otro lado en el vector **numeros**, (según el ejemplo, como resultado, **letras** tendría "MTK", y **numeros** tendría 3,1,4). La función devolverá el número de parejas letra-número que ha tratado (en el ejemplo, 3). Para convertir de un dígito de la cadena **codigo** a un valor int, se usará la función **calcular_int**. Si esa función devuelve menos de 1, esa pareja letra-número no hay que usarla y por tanto no se incluirá en los vectores **letras** y **numeros**, y no se cuenta.
3. Programar una función llamada **rellenar_con**, con 3 parámetros: **destino** es un vector de chars, de tamaño desconocido, **letra** es de tipo char, **repeticiones** es de tipo int y **comienzo** es de tipo int. Esta función debe colocar el valor de **letra** en el vector **destino**, comenzando por la posición indicada en **comienzo** y sucesivas, tantas veces como indique el parámetro **repeticiones**.
4. Programar una función llamada **generador**, con 4 parámetros: **destino** es un vector de chars (o cadena de caracteres), **letras** es un vector de chars, **numeros** es un vector de int y **longitud** es un valor int. Los vectores tienen una longitud desconocida. El parámetro **longitud** indica la longitud de datos útiles de los vectores **letras** y **numeros**. La función generador debe examinar por orden las letras y números de los vectores, e ir colocando caracteres en la cadena destino para generar el resultado completo. Para ello, irá llamando las veces necesarias a la función **rellenar_con**, para colocar las letras correspondientes a cada pareja letra-número. Tenga en cuenta donde comienza cada secuencia de letras para generar el resultado correcto.



UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA INDUSTRIAL
Departamento E.I.A.I.

APELLIDOS

NOMBRE N° Mat.

ASIGNATURA **INFORMÁTICA**

CURSO 1º GRUPO: FECHA 11/07/2013

Calificación

5. Completar la función main, para que haga las siguientes cosas:

- Pedir al usuario una cadena de caracteres (**codigo**).
- Usar la función **rellenar_con** para asignar el carácter '\0' a todas las posiciones de la cadena **resultado**
- Usar la función **separador** para obtener la secuencia de **letras** y **números**, y su longitud
- Usar la función **generador** para calcular la cadena **resultado**.

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#define MAX_CAD_CODIGO 30
#define MAX_CAD_DESTINO 500

void main()
{
    char codigo[MAX_CAD_CODIGO];
    char letras[MAX_CAD_CODIGO];
    int numeros[MAX_CAD_CODIGO];
    char resultado[MAX_CAD_DESTINO];
    int longitud;

    printf("Resultado: <%s>\n", resultado);
    getch();
}
```



UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA INDUSTRIAL
Departamento E.I.A.I.

APELLIDOS

NOMBRE

Nº Mat.

ASIGNATURA INFORMÁTICA

CURSO 1º

GRUPO:

FECHA 11/07/2013

Calificación

Ejercicio 3 (2 puntos. Tiempo: 25 minutos)

Una compañía de servicios energéticos ha implementado un nuevo sistema software para registrar de datos de sus clientes. Dicho software almacena en una estructura de datos los siguientes datos del cliente.

```
struct Cliente {
    char nomyapel[20];
    char direccion[20];
    float historico_luz[12];
    float historico_gas[12];
};
```

Se desea añadir al actual código dos funciones que serán llamadas desde el programa principal. En una de las funciones se debe permitir la inicialización de los datos desde el teclado mientras que la otra función debe generar una copia de seguridad de dichos datos. Si la copia se ha realizado correctamente, la función devolverá un valor entero diferente de CERO. Para verificar que los datos se han copiado correctamente, imprima en el programa principal todos los campos de la estructura copiada.

Complete el programa principal y realice las dos funciones.
Suponga que sólo se trabaja con UN único cliente.

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

typedef struct {
    char nomyapel[20];
    char direccion[20];
    float historico_luz[12];
    float historico_gas[12];
} Cliente;
void IniCliente(Cliente * );
int BckCliente (Cliente, Cliente *);
```

```
int main()
{
    Cliente c1, backup_c1;
```

```
}
```



UNIVERSIDAD POLITÉCNICA DE MADRID
 ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA INDUSTRIAL
 Departamento E.I.A.I.

APELLIDOS

NOMBRE N° Mat.

ASIGNATURA **INFORMÁTICA**

CURSO 1º GRUPO: FECHA 11/07/2013

Calificación

Ejercicio 4 (2 puntos. Tiempo: 35 minutos)

La secretaría de alumnos de la universidad necesita un programa para calcular la nota media de los alumnos que han terminado la carrera. Dicha nota se obtiene, en una escala de 1 a 4, al multiplicar los créditos aprobados por 1, los notables por 2, los sobresalientes por 3 y las matrículas de honor por 4, y dividiendo la suma de los productos obtenidos por el número total de créditos cursados.

Nota	Calificación	Puntuación
5,0 a 6,9	Aprobado	1
7,0 a 8,9	Notable	2
9,0 a 9,9	Sobresaliente	3
10	Matrícula de honor	4

Para calcular la nota media, el programa leerá un fichero de texto cuyo nombre será introducido por el usuario desde el teclado. Dicho fichero contiene, en su primera línea, tres cadenas de caracteres sin espacios en blanco, con el nombre, apellidos y código de expediente del alumno y, a continuación, un número indeterminado de líneas, una por asignatura, con el nombre de la asignatura, su número de créditos y la nota obtenida por el alumno. Por ejemplo:

Ejemplo de fichero de entrada:

Bartolomé	GómezGómez	EXP00769
Física	4.5	7.3
Algebra	6.0	6.8
Cálculo	3.0	9.1
....		

En este ejemplo, la nota media se calcularía como:

$$Media = \frac{4.5 \times 2 + 6.0 \times 1 + 3.0 \times 3}{4.5 + 6.0 + 3.0} = 1.777778$$

El programa pedirá indefinidamente nombres de ficheros de entrada al usuario e irá guardando en un fichero de salida denominado "NotasMedias.txt" las notas medias calculadas para cada uno de ellos. Este fichero de salida tendrá dos columnas, una con el código del expediente y otra con la nota media calculada previamente. Por ejemplo:

Ejemplo del archivo de salida "NotasMedias.txt":

EXP00769	1.777778
EXP00912	2.500000
.....	



UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA INDUSTRIAL
Departamento E.I.A.I.

APELLIDOS

NOMBRE N° Mat.

ASIGNATURA **INFORMÁTICA**

CURSO 1º GRUPO: FECHA 11/07/2013

Calificación

Ejercicio 5 (2 puntos. Tiempo: 25 minutos)

Un Organismo público almacena en una estructura de datos la información correspondiente a las dietas de viaje de sus empleados (máximo 100 empleados). De cada empleado guarda el **número de identificación** que es un entero diferente de cero, el **nombre y los apellidos**, el **grupo** (una letra del conjunto A, B, C ó D) y los **días** de viaje. Realizar un programa que solicite estos datos a través de una estructura de control repetitiva que se ejecutará hasta que el usuario introduzca como identificación de empleado el número cero. A continuación y en función de la tabla siguiente, el programa calculará el importe de la dieta multiplicando el número de días de viaje por la dieta diaria en base al grupo y almacenará el resultado en el campo de la estructura llamado **total**. Por último, el programa calculará la suma de los totales de los empleados del grupo A y lo mostrará por pantalla.

```
struct empleados_dietas {  
    int identificacion;  
    char nomyapel[20];  
    char grupo;  
    int dias;  
    float total;  
};
```

Grupo	Dieta diaria
A	75 €
B	70 €
C	60 €
D	50 €