



## Enunciado LAB 9-B.

Vamos a desarrollar en este caso dos ejercicios por separado; el primero enfocado al uso de vectores y el segundo enfocado al tratamiento de cadenas de caracteres.

**Ejercicio LAB\_9B\_1.** Un banco diseña un fondo de inversión para particulares. Está pensado para una inversión a 3 años de un capital de entre 8.000 y 50.000 euros. Se requiere un programa en C para hacer simulaciones sobre el rendimiento que ofrece un fondo, pudiendo consultar el valor del mismo en cada uno de los 36 meses. El cliente invierte su capital  $C_0$  en el mes considerado cero, y el fondo ofrece un resultado de dicho fondo mes a mes. Cada mes, se obtendrá un % de interés mensual de rendimiento, que aplicado al valor del fondo a primeros de mes ofrecerá el valor del mismo. Por ejemplo, si en el mes cero el cliente invierte 8.000 eur y sabemos que al mes 1 el interés de la inversión es un 0,07%, el valor del fondo al comienzo del mes 1 es  $8.000 * (1+0,07)\% = 8.005,60$  eur. Este es el valor de partida para el mes 1. A principios del mes 2 conoceremos el tipo que se aplica a este valor, para obtener el valor del fondo a mes 2. Y así

### Cuadro de simulación.

Se debe poder obtener una presentación del cuadro de simulación del fondo, presentando para cada mes: el valor del fondo a mes anterior, el porcentaje aplicado y el valor del fondo al final del mes. El valor del fondo a final del mes 35 es el valor de la inversión para el cliente (lo que percibirá). Para esta simulación, se debe proporcionar el interés de cada mes. Generalmente, los analistas ofrecen un interés trimestral, que puede convertirse fácilmente a su equivalente mensual.

### Simulación probable

Los analistas estiman como muy probables los siguientes intereses trimestrales: +0,21%, +0,15%, +0,33%, -0,45%, -1,00%, +0,12%, +0,24%, -0,66%, +0,22%, +0,36%, -0,12%, +1,25%. Obtener el cuadro de simulación correspondiente a estas indicaciones.

### Función de firma de código.

Debe existir una función denominada **firma\_de\_codigo()** que ofrezca el valor de firma para cada alumno. Dada una dirección de correo electrónico de alumno del tipo [abc1234@lu.ubu.es](mailto:abc1234@lu.ubu.es), se define como **NIA** al conjunto de 7 caracteres antes del dominio @alu.ubu.es. En este caso, la función de firma debe tomar los caracteres que ocupan posición impar en el NIA, tomar su código ASCII y aplicar a cada uno una función XOR con la máscara 11. De igual manera, debe tomar los caracteres que ocupan posición par en el NIA, tomar su código ASCII y aplicar a cada uno una función XOR con la máscara 10. Debe obtenerse la división modular entre 65 de la suma de los anteriores resultados parciales, concatenando el resultado de ambas. Por ejemplo, para un NIA abc1234, los caracteres impares son b,1,3 y los pares son a,c,2,4. La suma módulo 65 de los impares es 24, de los pares es 6, y la firma sería 2406. **NOTA: todas las funciones que sean empleadas en las opciones de 1 a 4 recibirán de manera obligatoria como último argumento el valor de la firma de código.** Su código se ejecutará sólo si el valor de la firma aportado como argumento es idéntico al que aporta la función **firma\_de\_codigo** detallada en la opción 5; en caso contrario, deberá aparecer un mensaje de error al efecto.



### Programa a realizar.

Se solicita realizar un programa en lenguaje C como una herramienta que sea capaz de realizar las siguientes acciones ofrecidas por menú:

1. **Solicitar y validar** los datos que permiten calcular el cuadro de simulación del fondo ofreciendo capital de inversión e intereses trimestrales.
2. **Ofrecer el cuadro de simulación.** El cuadro debe ofrecerse con información mensual. y se debe poder visualizar dicho cuadro en la pantalla o bien obtener un fichero de texto con el contenido del mismo (simulación.txt).
3. Realizar el **cálculo del cuadro de simulación probable** y poder visualizar dicho cuadro en la pantalla o bien obtener un fichero de texto con el contenido del mismo (simulación\_probable.txt).
4. **Ofrecer por pantalla el valor del fondo en el mes deseado.** En particular, es interesante ese valor al final de cada año. Ofrecer el valor de la inversión al final del tercer año, y ofrecer el porcentaje global de rendimiento de la inversión.
5. **Firma de código:** ofrecer por pantalla el valor de la firma de código.

Guarda el código en el fichero **lab\_9b\_1.c**. El programa debe funcionar al menos con el ejemplo descrito, y debería funcionar en general para cualquier ejemplo.



**Ejercicio LAB\_9B\_2.** Un alumno de programación, harto de los comentarios de su profesor, se plantea desarrollar una herramienta para **auto documentar** el código que genera en lenguaje C, particularmente para las variables y para los comentarios de cabecera de funciones. Es decir, se trata de generar otro fichero de código en C que contenga el mismo código fuente pero comentado convenientemente, conforme al ejemplo que se adjunta. Se muestra en primer lugar el código en bruto, y después el código que debe obtenerse gracias a este programa.

### Código "en bruto"

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

int sumar (int sumnado1, int sumando2);
unsigned char esImPar (int valor);
void escribeHola();

int main() {
    int valor1;
    int valor2;
    int suma;
    int resul;
    unsigned char paridad;

    printf("Teclea los sumandos: ");

    resul=scanf("%d %d", &valor1, &valor2);
    if (resul == 2 )
        suma = sumar (valor1, valor2);
    else
        suma=0;

    if (resul >= 1 )
        paridad = esPar(valor1);
    else
        paridad=-1;

    printf("Suma: %d\n", suma);
    printf("Paridad: %u\n", paridad);
    escribeHola();

    return 0;
}

int sumar (int sumando1, int sumando2) {
    int suma;

    suma = sumando1 + sumando2;
    return suma;
}

unsigned char esImPar (int valor) {
    unsigned char paridad;

    paridad = valor % 2 !=0;
    return paridad;
}

void escribeHola () {
    printf("Hola\n");
    return;
}
```



**Código resultante:**

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

/*
-----
Programa   : ejemplo_ok.c
Autor      : Paco Gonzalez Moya
Versión    : 1.0
Fecha      : 10-05-2020
Nº líneas  : 64
Objetivo   :
-----
*/

int sumar (int sumandol, int sumando2); //Prototipo de sumar()
unsigned char esImPar (int valor);      //Prototipo de esImPar()
void escribeHola();                    //Prototipo de escribeHola();

int main() {
    int valor1;           //Variable valor1:
    int valor2;           //Variable valor2:
    int suma;             //Variable suma:
    int resul;            //Variable resul:
    unsigned char paridad; //Variable paridad:

    printf("Teclea los sumandos: ");

    resul=scanf("%d %d", &valor1, &valor2);
    if (resul == 2 )
        suma = sumar (valor1, valor2);
    else
        suma=0;

    if (resul >= 1 )
        paridad = esPar(valor1);
    else
        paridad=-1;

    printf("Suma: %d\n", suma);
    printf("Paridad: %u\n", paridad);
    escribeHola();

    return 0;
}

/*
-----
Función   :   sumar()
Objetivo  :
Versión   : 1.0

@param int sumandol E/ :
@param int sumando2 E/ :
@return   :
-----
*/

int sumar (int sumandol, int sumando2) {
    int suma; // Variable suma:

    suma = sumandol + sumando2;
    return suma;
}
```



```
/*
-----
Función :   esImPar()
Objetivo :
Versión  : 1.0

@param int valor E/ :
@return :
-----
*/
unsigned char esImPar (int valor) {
    unsigned char paridad; //Variable paridad:

    paridad = valor % 2 !=0;
    return paridad;

}

/*
-----
Función :   escribeHola()
Objetivo :
Versión  : 1.0

@return :
-----
*/
void escribeHola () {
    printf("Hola\n");
    return;
}
```

Por desgracia, este alumno no sabe cómo hacerlo, así que te ha pedido ayuda. Debes realizar el código en lenguaje C que permita **conseguir el código resultante, que evidentemente debe compilar sin errores como lo hace el código en bruto**. Puedes hacer algunas suposiciones razonables sobre el tamaño de la línea del código fuente (no suele pasar de 80 caracteres).

Hay aspectos del documentador que deberán estar parametrizados y debe ser posible cambiarlo vía el programa (mínimo nombre del autor, fecha y versión, y aquellos que entiendas que se puede proporcionar).

Guarda el código en el fichero **lab\_9b\_2.c**. El programa debe funcionar al menos con el ejemplo descrito, y debería funcionar en general para cualquier ejemplo. Puedes emplear las funciones de la librería string.h