

Práctica 1:

Primeros Pasos en Programación. Entrada/Salida y operaciones simples.

Objetivos:

- Familiarizarse con el entorno de programación.*
- Detectar y corregir errores de compilación y de diseño y ejecución.*
- Usar y entender sentencias de Entrada/Salida.*
- Manejo básico de variables.*
- Uso correcto de los operadores aritméticos básicos en C.*

Desarrollo:

Comenzaremos con una serie de ejercicios en los que trabajaremos con variables, asignaciones y expresiones. Estos ejercicios también nos servirán para familiarizarnos con el entorno de programación que utilizaremos durante el curso, en este caso C-Lion.

Con cada uno de estos ejercicios que se planteen, se creará un proyecto en el entorno. Una vez escrito el código, ejecútalo para ver el resultado. Prueba a ejecutarlo paso a paso mediante el depurador (debugger) para familiarizarte con este mecanismo de ejecución que resultará de gran utilidad para detectar errores de ejecución.

Los problemas más sencillos de reflejar mediante un algoritmo son siempre problemas de cálculo, (problemas de física, economía, estadística...). En general, los problemas matemáticos son fácilmente trasladables a algoritmos, por el hecho de que en matemáticas realmente, tanto el proceso que utilizamos, como el lenguaje ya es de por sí algorítmico. Es por eso que lo más fácil para iniciarse suelen ser problemas de este tipo.

Para ello haremos los siguientes programas, los cuales deben tener la siguiente estructura:

```
#include <stdio.h>
void main(void) {
    /* Declaración de variables */
    ...
    /* Lectura de datos */
    ...
    /* Cálculos intermedios */
    ...
    /* Salida de datos por pantalla */
    ...
}
```

Ejercicios propuestos:

1 Resultado de la ecuación $a=b^2+1$

Este programa muestra el resultado de la ecuación $a=b^2+1$ para distintos valores de b (0, 2 y -2). ¿Es correcto? Si no lo es, escribir el programa correctamente.

```
#include <stdio.h>
main() {
    int a, b=0;
    a=b*b+1;
    printf ("Si b=%d, entonces a=%d\n", b, a);
    b=2;
    printf ("Si b=%d, entonces a=%d\n", b, a);
    b=-2;
    printf ("Si b=%d, entonces a=%d\n", b, a);
}
```

2 Calculadora:

Queremos implementar un programa en C que dados dos números enteros (que leeremos del teclado) calcule: suma, resta, producto, división entera.

- Los dos números deberán almacenarse en variables.
- Los resultados de las operaciones deberán guardarse en variables.
- Tras la ejecución del programa se mostrará para cada operación su resultado.

Para ayudarte en este primer programa aquí tienes la estructura:

```
#include <stdio.h>
int main(void){
    /* Declaración de variables. */
    int num1,num2;

    /* Elimina los comentarios y sustituye los
    interrogantes por el tipo de datos correspondiente. */
    int sum;
    ??? res;
    ??? prod;
    ??? div_e;

    /* Cuerpo de programa */
    printf("Bienvenido/a a \"Calculadora v1.0\".\n");
    printf("\n");
    printf("Introduce Primer operando:\t");
    scanf("%d", &num1);
    printf("Introduce Segundo operando:\t");
    scanf("%d", &num2);

    // Añade un salto de línea más.
    printf("...calculando resultados...\n");

    // Calcula suma (éste te lo damos hecho)
    sum = num1 + num2;

    // Calcula resta (no te olvides de sustituir los interrogantes...)
    res = ???;
    // Calcula producto (completa el código)
    ???
    // Calcula división entera
    ???

    //Muestra los resultados
    printf("Suma:\t\t%d\n", sum);
    printf("Resta:\t\t%d\n", ???);
    ???
    ???

    printf("\nBye\n");
    return 0;
}
```

3 Modifica el ejercicio anterior para:

- Que el resultado de la división sea un número con dos decimales.(OPCIONAL)
- Añadir las operaciones de: división real, resto de la división (módulo), el tanto por ciento de uno de los números, media de los dos números, truncar un número real a entero, obtener la mitad de la suma del primer número más el doble del segundo.

4 Cambio a mayúsculas

Escribe un programa que lea un carácter en minúsculas e imprima en pantalla el mismo carácter en mayúsculas.

El programa deberá:

- a) Leer un carácter en minúscula que se almacenará en la variable *minusculta*.
- b) Transformar este carácter en minúscula al correspondiente carácter en mayúscula, que se almacenará en la variable *mayuscula*.

Proceso que permitirá transformación: Para ello, se utiliza el código ASCII del carácter en minúscula. Cada uno de los caracteres que es posible representar en el ordenador ('1', '2', 'a', 'b', 'A', 'B', '?', etc.) tiene asociado un código que lo identifica, llamado código ASCII. Así,

- los caracteres que representan dígitos (0, 1, 2, ..., 9) tienen códigos consecutivos a partir del 48 (para el carácter '0')
 - los caracteres que representan letras del abecedario en mayúscula (A, B, ..., Z) tienen asignados códigos consecutivos entre el 65 (para el carácter 'A') y el 90 (para el carácter 'Z')
 - los caracteres que representan letras del abecedario en minúscula (a, b, ..., z) tienen asignados códigos consecutivos entre el 97 (para el carácter 'a') y el 122 (para el carácter 'z').
- c) Escribir en pantalla el carácter en mayúscula obtenido, es decir, el contenido de la variable *mayuscula*.

Implementación

```
#include <stdio.h>
/* Programa que lee un caracter en minusculas y
lo escribe en mayusculas */

main() {
char minusculta, mayuscula;

    /* Lectura de datos: minusculta. */
    /* Proceso: transformacion a mayuscula */
    /* Salida */
}
```

5 Terminación de número premiado en el sorteo de la ONCE:

Escribe un programa que lea el número que ha salido en el sorteo de la ONCE e imprima en pantalla la última cifra de este número.

6 Informe de ventas:

Una compañía de refrescos comercializa tres productos: de cola, de naranja y de limón. Se desea realizar un programa que calcule las ventas realizadas de cada producto. Para ello, se leerá la cantidad vendida (máximo 5000000) y el precio en euros de cada producto y se mostrará un informe de ventas como el que sigue:

Producto	Ventas	Precio	Total
Cola	1000000	0.17	170000.00
Naranja	350000	0.20	70000.00
Limon	530000	0.19	100700.00
		TOTAL	340700.00