

LABORATORIO 1

OPERACIONES DE ENTRADA Y SALIDA

1 OBJETIVOS

Al finalizar esta actividad, serás capaz de:

- Entender el funcionamiento del compilador.
- Definir variables y efectuar asignaciones de valores con operaciones aritméticas básicas.
- Escribir un programa sencillo con entrada y salida de datos.
- Crear un sistema de almacenamiento de datos para tus trabajos.

2 MOTIVACIÓN

Para escribir cualquier programa vamos a tener que utilizar el compilador, conviene, por tanto, conocer adecuadamente su funcionamiento.

En la resolución de la mayoría de los problemas tendremos que pedir datos al usuario para realizar cálculos u operaciones con ellos y mostrar resultados. La entrada y salida de datos será, por tanto, una parte básica de todos los programas que es *imprescindible* dominar.

Conviene conocer adecuadamente los sistemas de almacenamiento del ordenador si queremos poder recuperar nuestros programas o llevarlos a otro ordenador.

3 LAS VARIABLES, ENTRADAS Y SALIDAS EN C

Todos los datos de entrada, salida e intermedios de un programa tendrán que estar almacenados en variables o constantes.

Recordamos que una variable no es más que una parte reservada de la memoria a la cual ponemos un nombre para poder utilizarla y que cambia su valor a lo largo del programa. La manera de definirla es indicando el tipo y nombre seguido de punto y coma. Ejemplo:

```
int numero;  
float altura;
```

Las constantes son objetos cuyo valor no cambia a lo largo del programa y se indican en la parte de las directivas del preprocesador. Ejemplo:

```
#define PI 3,1416
```

Para la salida de datos utilizaremos la sentencia printf con todos los argumentos necesarios. Ejemplo:

```
printf("\nel valor de num es: %d y el de cant es: %d", num, cant);
```

Para la entrada de datos utilizaremos la sentencia scanf con sus argumentos necesarios. Ejemplo:

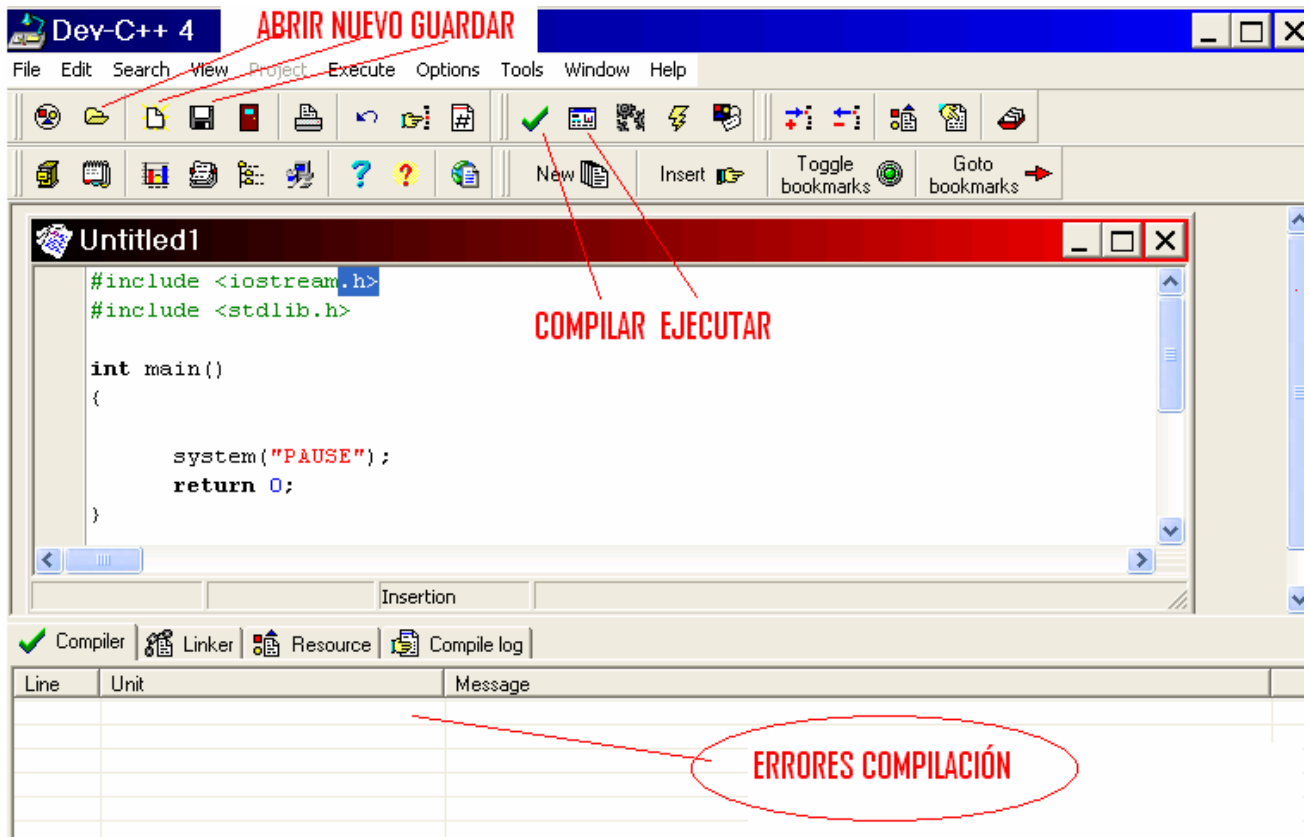
```
int a, b = 18, c;  
scanf("%d", &a);
```

4 EJERCICIOS

1.1 Ejercicio 1

1.1.1 Objetivo

Familiarizarse con el manejo básico del compilador.



1.1.2 Ayuda

Ver imagen del apartado anterior (apartado 1.1.1).

1.1.3 Enunciado

En este ejercicio el alumno debe abrir el compilador Dev C++ y familiarizarse con los botones y áreas de mayor uso de la pantalla principal. La imagen del apartado 1.1.1 muestra la pantalla principal del compilador y resalta los objetos de mayor interés.

1.2 Ejercicio 2

1.2.1 Objetivo

El objetivo de este ejercicio es poner en práctica lo visto acerca del compilador así como la utilización de las instrucciones de entrada/salida de datos utilizando una única variable en cada operación.

1.2.2 Ayuda

En este caso y por ser el primer ejercicio se ofrece la solución completa del enunciado. Se tratará por tanto de copiarla utilizando el editor del compilador, compilar, ver que no da errores, guardar y ejecutar el código.

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

#define PI 3.1415

void main()
{
    float radio,area,longitud;
    printf("Introduce el radio: ");
    scanf("%f",&radio);
    longitud=2*PI*radio;
    printf("Longitud: %.2f",longitud);
    area=PI*radio*radio;
    printf("\nArea: %.2f\n",area);
    system("PAUSE");
}
```

1.2.3 Enunciado

Leer el radio de un círculo y calcular y mostrar su longitud y su área.

Ejemplo:

```
Introduce el radio: 3.1
Longitud: 19.48
Area: 30.19
Presione una tecla para continuar . . .
```

1.3 Ejercicio 3

1.3.1 Objetivo

El objetivo de este ejercicio es realizar nuestro primer programa utilizando definiciones de variables, la entrada y salida de datos y asignaciones de variables de modo correcto para realizar una conversión.

1.3.2 Ayuda

En el siguiente ejemplo se realiza la conversión de grados Centígrados a Fahrenheit utilizando la fórmula $Fahrenheit = (9/5 * Celsius) + 32$.

```
#include<stdio.h>
#include<stdlib.h>
/*Conversion de grados Centigrados a Farenheit*/
```

```
void main()
{
    float gradosc;
    float gradosf;
    printf("Conversion de grados centigrados a grados fahrenheit\n");
    printf("Dame los grados centigrados: ");
    scanf("%f",&gradosc);
    gradosf=(9.0/5.0*gradosc)+32.0;
    printf("\nLos grados fahrenheit son:%f",gradosf);
    system("pause");
}
```

1.3.3 Enunciado

Escribir un programa que implemente una euro-calculadora. El programa pide un número en euros y lo convierte a pesetas.

Utiliza el modelo anterior para solicitar euros en lugar de grados centígrados y asignar a una variable llamada pesetas el valor correspondiente a la conversión utilizando la fórmula $\text{pesetas} = \text{euros} * 166.386$

Ejemplo:

```
Conversión de euros a pesetas
Introduce la cantidad en euros: 6
La cantidad en pesetas equivalente es: 998
```

1.4 Ejercicio 4

1.4.1 Objetivo

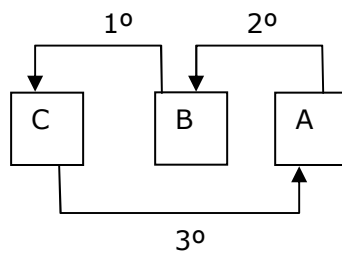
El objetivo de este ejercicio es realizar un programa con definiciones de variables y la entrada y salida de datos y utilizar las asignaciones de variables para resolver un problema concreto que implica la comprensión del funcionamiento de las variables e introduce el concepto de variable auxiliar.

1.4.2 Ayuda

Cuando cambiamos de valor una variable el valor anterior se pierde, es decir, que si yo hago la asignación $a=b$; la variable a pasa a valer lo que valía b pero si no he guardado el valor de a antes sobre otra tercera variable auxiliar, perderé su valor inicial.

Si quiero intercambiar los valores de dos variables a y b los pasos serán por tanto:

1. Pasar de a a una variable auxiliar c para guardar su contenido.
2. Machacar el contenido de a con lo que tiene b .
3. Modificar b con lo que tiene c (que será lo que tenía a antes de cambiar su valor).



1.4.3 Enunciado

Leer dos números por teclado e intercambiar sus valores. NOTA: Es importante entender que cuando se dice intercambiar hay que MODIFICAR su valor, y no solamente mostrar el mensaje indicando una variable y visualizando la otra.

Ejemplo:

Intercambio del contenido de 2 variables.

Introduce el valor de A: 7

Introduce el valor de B: 3

Tras intercambiar los valores, A vale 3 y B vale 7.

1.5 Ejercicio 5

1.5.1 Objetivo

El objetivo de este ejercicio es utilizar varios datos de entrada así como las constantes simbólicas para efectuar un cálculo.

1.5.2 Ayuda

Una constante simbólica se define como un nombre que cuando es utilizado en el programa tiene un valor fijo. Podemos utilizarlo por ejemplo, para especificar lo que se pagará por hora extra o por hora extraordinaria y luego en nuestro programa utilizamos el nombre para calcular sueldos en base a las horas trabajadas y a estos precios por hora constantes para todos los trabajadores:

```
# define EXTRA 40
# define ORDINARIA 20
```

1.5.3 Enunciado

Realizar un programa que solicite dos datos: el número de horas extras y ordinarias trabajadas por un empleado (serán enteros) y calcule cuál es su sueldo total utilizando el precio por hora de 40 y 20 euros la hora respectivamente.

NOTA: Utiliza los defines del ejemplo.

Ejemplo:

Calculo del sueldo en función de horas ordinarias y extras.

Introduce número de horas ordinarias: 10

Introduce número de horas extra: 20

De acuerdo con los datos introducidos el sueldo es: 1000

1.6 Ejercicio 6

1.6.1 Objetivo

El objetivo de este ejercicio es realizar unos cálculos más complejos que en los casos anteriores que utilicen varias variables e impliquen varias operaciones.

1.6.2 Ayuda

El siguiente ejemplo solicita un número de días e indica cuántas semanas completas hay por ejemplo en 15 días indicando que son dos semanas y un día.

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

void main()
{
    int semanas,dias,rdias;
    printf("Numero de dias: ");
    scanf("%d",&dias);
    semanas = dias/7;
    rdias = dias%7;
    printf("\n %d dias son:\n %d semanas y %d dias \n",dias, semanas, rdias);
    system("PAUSE");
}
```

1.6.3 Enunciado

Una empresa se encarga de la producción de tres tipos de piezas P1,P2 y P3 y los tiempos de producción por pieza son de 3 horas para P1, 4 horas para P2 y 5 horas para P3.

Pedir al usuario un número x de piezas P1, un número y de P2, z de P3 calcular cuánto se tardará en la producción, es decir cuántas semanas, días y para que hora de la jornada laboral estará el pedido sabiendo que el horario de producción de la empresa es continuo de 8:00 a 16:00.

Nota: Utiliza la idea en el ejemplo para obtener las semanas, días, y horas restantes en el total de horas de producción implicadas.

Ejemplo:

Introduce el numero de piezas P1: 10
Introduce el numero de piezas P2: 11
Introduce el numero de piezas P3: 12

El total de horas necesarias es: 134

Teniendo en cuenta el horario de la empresa, si empezaran mañana a fabricar las piezas, acabarían después de 3 semanas y 1 día completos de trabajo, a las 14 horas del día siguiente.