



## **Práctica Nº 1.**

### **Entorno de Programación Code::Blocks e Introducción a C++**

#### **Preliminares.**

1. Para familiarizarse con el Entorno de Programación (IDE) Code::Blocks, el alumno debe seguir las indicaciones que se proporcionan en el documento “Guía de uso de Code::Blocks”, que se encuentra en el Campus Virtual. En dicho documento se proporciona información sobre cómo crear, editar y compilar ficheros de código fuente en C++, así como realizar la ejecución del fichero ejecutable generado.

#### **Ejercicios de clase.**

2. El siguiente programa escrito en C++ calcula la cantidad bruta y neta a pagar por un trabajo realizado en función de las horas y días trabajados. Sin embargo, en el momento en que se intenta compilar se producen una serie de errores. El alumno debe localizar dichos errores y corregirlos. Para ello debe examinar los mensajes que proporciona el compilador e interpretarlos convenientemente.

```
#include <iostream>
using namespace std;

const tasa : 25.0;
const PRECIO_HORA = 60.0;

int main()
{
    double horas,dias,total,neto;

    cout << "Introduzca las horas trabajadas: ";
    cin << horas;
    cout << "Introduzca los días trabajados: ";
    cin >> dias;
    horas*dias*PRECIO_HORA = total;
    neto = total-TASA;
    cout << "El valor total a pagar es: ";
    cout << total << endl;
    cout << "El valor neto a pagar es: ";
    cout << NETO << endl;

    return 0;
}
```

Una vez compilado correctamente, la ejecución de este programa para 8 horas trabajadas durante 5 días produce el siguiente resultado:

```
Introduzca las horas trabajadas: 8
Introduzca los días trabajados: 5
El valor total a pagar es: 2400
El valor neto a pagar es: 2375
```

3. Escriba un programa que acepte un dato de tipo **int** de teclado y posteriormente lo escriba en pantalla. Ejecútelo introduciendo un número **int** válido, y posteriormente ejecútelo introduciendo por teclado un dato que no pertenezca al tipo **int**, por ejemplo una palabra cualquiera. Añada un comentario al principio del programa en el que explique cuáles son las diferencias que ha encontrado entre ambas ejecuciones del mismo programa.
4. Escriba un programa que visualice por pantalla el tamaño en bytes que ocupan todos y cada uno de los tipos básicos vistos en clase: **int**, **long**, **char**, etc. Para ello debe usarse el operador **sizeof**. Por ejemplo, para el tipo **int**:

```
cout << "int: " << sizeof(int) << " bytes" << endl;
```

5. Escriba un programa que lea cuatro letras por teclado, y posteriormente escriba dichas letras de manera que cada una de ellas se encuentre codificada sustituyéndola por aquel carácter que le sigue en la tabla de código ASCII. Por ejemplo, ante la entrada **SETO** debe producir la salida **TFUP**.
6. Escriba un programa que lea de teclado un número natural, que representa una cierta cantidad de Bytes, y muestre por pantalla los MBytes, KBytes y Bytes que podemos obtener (Considerando que 1 KByte es equivalente a 1024 Bytes, y 1 MByte es equivalente a 1024 KBytes). Por ejemplo, dado el número 26871979, el resultado sería 25 MBytes, 642 KBytes y 171 Bytes, ya que  $26871979 \text{ Bytes} = 25 \text{ MBytes} + 642 \text{ KBytes} + 171 \text{ Bytes}$ .

## Ejercicios de refuerzo.

7. El siguiente programa escrito en C++ calcula el área y la longitud de una circunferencia. Al compilarlo se producen una serie de errores que el alumno debe localizar y corregir.

```
#include <iostream>
using namespace std;

const int PI=3.1416

int main()
{
    double longitud ,area;
    int radio;

    out << "Hola" ; << endl;
    cout<< "Este programa calcula la longitud y el area de un circulo"

    cin << "Introduce el radio del circulo: " >> radio;

    long = 2*PI*radio;
    area = PI*(radio*radio)
    cout << 'Area = ' << area << endl;
    cout << 'Longitud = ' << area << endl;

    return 0;
}
```

Una vez compilado correctamente, la ejecución de este programa para una circunferencia de radio 30 produce el siguiente resultado:

```
Hola
Este programa calcula la longitud y el área de un círculo
Introduce el radio del círculo: 30
Area = 2827.44
Longitud = 188.496
```

8. Escriba un programa que lea de teclado cuatro letras minúsculas (supondremos que la entrada de datos es correcta), y posteriormente escriba las letras mayúsculas correspondientes a las letras minúsculas leídas previamente.

Tenga presente que no es necesario conocer ni consultar la tabla ASCII de codificación de los caracteres. Sólo es necesario conocer que las letras minúsculas (desde la 'a' hasta la 'z') están consecutivas, y que las letras mayúsculas (desde la 'A' hasta la 'Z') están consecutivas.

Ejecute el programa y compruebe si produce los resultados esperados. Por ejemplo, ante la entrada `hola` debe producir la salida `HOLA`.

9. Escriba un programa que sólo declare variables de tipo **int**. El programa deberá leer dos números desde el teclado; posteriormente los sumará y almacenará el resultado en una variable; finalmente escribirá por pantalla el resultado de la suma. Ejecute el programa con

datos cualesquiera y verifique que funciona. Después ejecute dicho programa tomando como datos de entrada 1 y 3000000000. ¿Por qué no funciona?

10. Escriba un programa que lea por teclado una cierta cantidad de segundos y muestre su equivalente en semanas, días, horas, minutos y segundos. Por ejemplo, si se lee la cantidad de 3672 segundos, la salida será: 0 semanas, 0 días, 1 hora, 1 minuto, 12 segundos.
11. Escriba un programa que calcule la nota final de una asignatura. Para ello deberá leer por teclado la nota de la parte de teoría y la nota de la parte de problemas, y habrá de calcular la nota final considerando que la parte de teoría vale un 70% de la nota final y la de práctica un 30%. Por ejemplo, si se introduce un 10 para teoría y otro 10 para problemas, la salida será un 10. Si se introduce un 8 para teoría y un 5 para problemas, la salida será un 7.1.
12. Escriba el siguiente código, ejecútelo y descubra qué hace este programa y cómo lo hace.

```
#include <iostream>
using namespace std;

int main()
{
    int a=6, b=14;
    int auxiliar;
    cout << "a vale " << a << " y b vale " << b << endl;

    // ¿Qué hacen estas tres sentencias?
    auxiliar = a;
    a = b;
    b = auxiliar;

    cout << "a vale " << a << " y b vale " << b << endl;

    return 0;
}
```

Sustituya las tres asignaciones que hay tras el comentario por estas otras tres:

```
a = a + b;
b = a - b;
a = a - b;
```

Puede comprobarse que el resultado es análogo al caso anterior: estudie cómo funciona este nuevo programa.