Escuela Politécnica Superior de Elche

FUNDAMENTOS DE INFORMÁTICA
1º Ingeniería Industrial

PRÁCTICA 2:
Lenguaje C: Conceptos básicos y programación elemental

CURSO 2007-2008

División de Ingeniería de Sistemas y Automática

Práctica 2. Sesión 1.

EJERCICIO 1: ¿Qué intervalo de números enteros se puede representar con una variable tipo *int* en los ordenadores del aula de informática? ¿Y con una tipo *short*? ¿Y con una tipo *long*? ¿Cuántos bits utilizan dichos ordenadores para una variable tipo *float*? ¿Y para un *char*? Razona las respuestas.

Solución:

EJERCICIO 2: Calcular el valor de la variable de la izquierda en las siguientes expresiones, teniendo en cuenta que las variables se han definido del siguiente modo:

```
int x = 5, y = 5, z=2;
float a=2.9, b;
char c;
```

Suponer que en cada paso, el valor previo de las variables es el que tomaban tras evaluar la expresión de la fila anterior de la tabla.

	Expresión	Valor
1.	z *= (x+1);	z =
2.	z /= (5+x);	z =
3.	z = ++x - y;	z =
4.	z = x++ - 2;	z =
5.	z = y % (x+2) ;	z =
6.	b = x/2;	b =
7.	b = x/2.;	b =
7.	b = (float)x/2;	b =
8.	z = (x+a)/3;	z =

9.	b = (x+a)/3;	b =
10.	z = a+'5'-5;	z =
11.	c = x+y+'a';	c =
12.	z = ((x != y) || (a>4)) ;	z =
13.	z = ((x != y) && (a>4)) ;	z =
14.	x+y = z;	z =
15.	z = (x==y);	z =
16.	z = 2-x/3*y	z =

EJERCICIO 3: Escribir el siguiente programa en Dev-C++:

```
void main(void)
{
      int seg, min, resto;
      int segMin = 60;
      printf("Introduzca número de segundos:\n");
      scanf("%d", &seg);
      min = seg/segMin;
      resto = seg%segMin;
      printf("%d seg. Son %d min. y %d seg.\n", seg,min,resto);
}
```

Este programa lee un número expresado en segundos y lo transforma en un número expresado en minutos y segundos. La sentencia *scanf("%d", &seg);* sirve para leer por teclado un número entero (*int*) y guardar el valor tecleado en la variable *seg*.

Compilar, enlazar y ejecutar el proyecto. Comprobar la salida que se obtiene para las entradas 60, 129 y 230 y escribirla en el cuadro siguiente:

EJERCICIO 4: Escribir el siguiente programa en Dev-C++:

```
main()
{
      int x = 336;
      int y = 598745215458745698;
      printf("%d\n", y);
      printf("%d  %o  %x\n", x, x, x);
      printf("%d  %u\n", -336, -336);
      printf("%c  %d\n", 'A', 'A');
      printf("%d  %c\n", 80, 80);
      printf("%d  %c\n", 336, 336);
}
```

El significado de los códigos de formato es el siguiente:

Código de formato	Significado
%d	Número entero con signo expresado en decimal (base 10)
%o	Número entero con signo expresado en octal (base 8)
%x	Número entero con signo expresado en hexadecimal (base 16)
%u	Número entero sin signo expresado en base 10
%c	Carácter

Compilar, enlazar y ejecutar el proyecto. Comprobar la salida que se obtiene y escribirla en el cuadro siguiente, explicando el resultado obtenido.

EJERCICIO 5: Escribir el siguiente programa en Dev-C++:

```
main()
{
      int x = 0X1D;
      int y = 012;
      int z = 43;
      printf("%d  %d  %d\n", x, x, x);
      printf("%o  %o  %o\n", x, x, x);
      printf("%x  %x  %x\n", x, x, x);
}
```

Cuando el valor de una variable entera se define poniendo delante *0X*, dicho valor viene expresado en hexadecimal, y cuando se antepone al valor *0*, viene expresado en octal.

Compilar, enlazar y ejecutar el proyecto. Comprobar la salida que se obtiene y escribirla en el cuadro siguiente, explicando el resultado obtenido.

EJERCICIO 6: Escribir el siguiente programa en Dev-C++:

```
main()
{
      printf("%2d\n",336);
      printf("%5d\n",336);
      printf("%f\n",134.745);
      printf("%3.2f\n",134.745);
      printf("%7.1f\n",134.745);
      printf("%7.1f\n",-134.745);
      printf("%10.3e\n",1234.56);
}
```

El significado de los códigos de formato es el siguiente:

Código de formato	Significado
%A.Bf	número en coma flotante, donde A es el número total de caracteres y B es el número de decimales
%Ad	número entero con signo
%A.Be	número en coma flotante, donde A es el número total de caracteres y B es el número de decimales, expresado en notación científica

Compilar, enlazar y ejecutar el proyecto. Comprobar la salida que se obtiene y escribirla en el cuadro siguiente:

EJERCICIO 7: Escribir un programa que lea una cantidad de euros (entera) y diga cuantos billetes de 50, 20, 10 y 5 euros, y cuantas monedas de 2 y 1 euro componen esa cantidad de dinero.
A continuación se muestra un ejemplo de ejecución del programa. En negrita se muestra el dato introducido por el usuario.

```
Introduzca una cantidad de euros: 48
Dicha cantidad consta de:
0 billetes de 50 euros
2 billetes de 20 euros
0 billetes de 10 euros
1 billetes de 5 euros
1 monedas de 2 euro
1 monedas de 1 euro
```

Programa:

EJERCICIO 8: Escribir un programa que pida las notas de un alumno correspondientes a los tres trimestres del curso. A continuación, debe mostrar las notas introducidas y la nota media, con una precisión de 1 decimal.
A continuación se muestra un ejemplo de ejecución del programa. En negrita se muestran los datos introducidos por el usuario.

Introduzca la primera nota: **4.8**
Introduzca la segunda nota: **7.56**
Introduzca la tercera nota: **9**

Nota 1: 4.8
Nota 2: 7.6
Nota 3: 9.0

La nota media es: 7.1

Programa:

EJERCICIO 9: Escribir un programa que lea por teclado los coeficientes de un polinomio de segundo grado $ax^2 + bx + c$ y que saque por pantalla todas sus raices (suponer que las soluciones del polinomio serán números reales).
Para calcular la raiz cuadrada de un número, existe dentro de la librería math.h una función llamada sqrt, a la que se le pasa como parámetro el número y devuelve como resultado su raiz cuadrada del siguiente modo:

```
float x, y=10;
x = sqrt(y);
```

En la variable *x* se almacenará la raiz cuadrada de *y*.
A continuación se muestra un ejemplo de ejecución del programa. En negrita se muestran los datos introducidos por el usuario.

```
Introduzca el coeficiente a: 1
Introduzca el coeficiente b: 4
Introduzca el coeficiente c: 3

Las soluciones son  -1  y  -3
```

Programa:

Práctica 2. Sesión 2.

EJERCICIO 1: Después de ejecutar el siguiente fragmento de programa, ¿Cuál será el valor de la variable x?

```
int x = 0;
int n = 16;
while(n % 2 == 0)
{
     x += n;
     n /= 2;
}
```

EJERCICIO 2: ¿Cuántas veces se ejecuta la función printf en el siguiente fragmento de código?

```
a = 9;
for(i=0; i<100; i++)
     if((a%4 == 0) || (i%2 == 0))
          printf("%d, %d\n", a, i);
```

```
for(i=0; i<100; i++)
{
     if ((i%4) == 0)
          continue;
     printf("%d\n", i);
}
```

```
for(i=0; i<100; i+=3)
{
     if ((i%9) == 0)
          break;
     printf("%d\n", i);
}
```

EJERCICIO 3: Escribir cual sería la salida por pantalla del siguiente programa:

```
int i, j;

for(i=1; i<4; i++)
     for(j=i; j<=5; j++)
          printf("i = %d, j = %d\n", i, j);
printf("i = %d, j = %d\n", i, j);
```

EJERCICIO 4: Escribir cual sería la salida por pantalla del siguiente programa:

```
#include <stdio.h>

void main(void)
{
  int i=0, x=0;

  for(i=1; i<10; i*=2)
  {
    if (i%5 == 0)
    {
      x++;
      printf("%d ", x);
    }
    ++i;
  }
  printf("\nx = %d, i = %d\n", x, i);
}
```

EJERCICIO 5: Escribir cual sería la salida por pantalla del siguiente programa:

```
#include <stdio.h>

void main(void)
{
  int i,j,k,x=0;

  for(i=0; i<5; i++)
    for (j=0; j<i; j++) {
      k = i+j-1;
      if (k%2 == 0)
        x += k;
      else if (k%3 == 0)
          x += k-2;
    printf("%d ", x);
  }
  printf("\nx = %d, i = %d, j = %d\n", x, i, j);
}
```

EJERCICIO 6: Escribir un programa que lea números enteros hasta que se introduzca un número <0. El programa debe devolver los siguientes datos:

- La media aritmética de los valores introducidos (precisión 3 decimales).
- El número de valores introducidos que son múltiplos de 3.
- El máximo y el mínimo número introducidos.

El número negativo no se debe tener en cuenta a la hora de calcular estos resultados. A continuación se muestra un ejemplo de ejecución. En negrita se muestran los valores introducidos por teclado.

```
Valor 1:  10
Valor 2:  5
Valor 3:  9
Valor 4:  6
Valor 5:  13
Valor 6:  12
Valor 7:  13
Valor 8:  4
Valor 9:  2
Valor 10:  -5

El valor medio de los datos introducidos es:  8.222
Se han introducido 3 múltiplos de 3
El maximo es 13 y el minimo 2
```

EJERCICIO 7: Escribir un programa que lea un número entero y saque por pantalla todos sus divisores.

```
Introduzca un numero:  2546

Los divisores de 2546 son:
1 2 19 38 67 134 1273 2546
```

EJERCICIO 8: Escribir un programa que calcule el factorial de un número. A continuación se muestra un ejemplo de ejecución.

```
Introduzca un número:  8
El factorial de 8 es:
8*7*6*5*4*3*2*1
Resultado = 40320
```

Modificar el programa anterior para que se calcule el factorial sólo si el número introducido es mayor a cero. En caso de que el número sea cero o negativo, se debe volver a pedir (el número de veces necesarias) hasta que se introduzca uno positivo. A continuación se muestra un ejemplo de ejecución.

```
Introduzca un numero:  -2
El número introducido debe ser mayor a 0
Introduzca un numero:  0
El número introducido debe ser mayor a 0
Introduzca un numero:  10
El factorial de 10 es:
10*9*8*7*6*5*4*3*2*1
Resultado = 3628800
```

EJERCICIO 9: Escribir un programa que lea un número entero y diga si el número leído es primo o no primo.

Práctica 2. Sesión 3.

EJERCICIO 1: Un número perfecto es un entero positivo igual a la suma de sus divisores propios. Un divisor propio es un divisor del número distinto del número en si mismo. Por ejemplo, 6 es un número perfecto porque la suma de sus divisores propios 1, 2 y 3 es igual a 6. El número 8 no es perfecto porque la suma de sus divisores propios 1 + 2 + 4 es distinta de 8.

Se pide escribir un programa que pida por pantalla un número y a continuación indique si dicho número es o no un número perfecto. A continuación debe volver a pedir otro número y repetir el proceso hasta que el usuario introduzca un número 0 o negativo.

EJERCICIO 2: Escribir un programa que lea un número entero por teclado, y a continuación saque por pantalla un triángulo de base igual al número introducido, con el siguiente formato:

```
Introduzca n:  7
1
12
123
1234
12345
123456
1234567
```

EJERCICIO 3: Se puede calcular la exponencial de x en puntos cercanos a 0 de forma aproximada, sumando los n primeros términos de la serie infinita siguiente:

$$e^x = \sum_{i=0}^{n} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \ldots$$

Escribir un programa que pida al usuario que introduzca el valor de x y la cantidad de términos a sumar (n), y calcule el valor de la exponencial.

```
Introduce el valor de x: 1
Introduce el limite de la suma: 4
La aproximacion de la exponencial es: 2.70833
```

Modificar el anterior programa para que únicamente se pida el valor de *x* y la exponencial se calcule con un error menor a 0.0000001.

EJERCICIO 4: Se puede calcular el seno de x en puntos cercanos a 0 de forma aproximada, sumando los n primeros términos de la serie infinita siguiente:

$$senx = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \ldots$$

Escribir un programa que pida al usuario que introduzca el valor de x y el valor de n (siendo n el exponente del numerador del último término que se debe sumar), y calcule el valor del seno.

```
Introduce el valor de x: 1
Introduce el limite de la suma: 4
La aproximacion del seno es: 0.83333
```

EJERCICIO 5: Escribir un programa que lea un número entero positivo y escriba todos los números primos menores o iguales al número introducido.

```
Introduce valor: 20
Los numeros primos menores o iguales a 20 son:
1, 2, 3, 5, 7, 11, 13, 17, 19
```

EJERCICIO 6: Escribir un programa que actúe como una calculadora elemental. Al principio, el programa debe mostrar un menú en el que el usuario elige la operación a realizar. A continuación, se deben pedir los operandos y se devuelve el resultado. Esta secuencia se repite hasta que el usuario elija la opción de Salir.

El programa se debe ejecutar correctamente independientemente de si se introducen letras mayúsculas o minúsculas.

Se debe utilizar la sentencia switch para resolver el problema.

```
Opciones:
a) Suma
b) Resta
c) Multiplicación
d) Division
e) Salir
Introduce opcion: a

Introduce operando 1: 5
Introduce operando 2: 3
5 + 3 = 8

Pulse una tecla para continuar

Opciones:
a) Suma
b) Resta
c) Multiplicación
d) Division
e) Salir
Introduce opcion: g

Opcion desconocida

Pulse una tecla para continuar

a) Suma
b) Resta
c) Multiplicación
d) Division
e) Salir
Introduce opcion: C

Introduce operando 1: 7
Introduce operando 2: 4
7 x 4 = 28

Pulse una tecla para continuar

a) Suma
b) Resta
c) Multiplicación
d) Division
e) Salir
Introduce opcion: e

Fin del programa
```