



ARQUITECTURA DE REDES
Laboratorio

**Práctica 1. Revisión y ampliación de la programación en
Lenguaje C**

OBJETIVOS

1. Revisar los principios de la programación en lenguaje ANSI C
2. Realizar con soltura programas de manipulación de flujos de entrada/salida en lenguaje C

CONTENIDOS

- Introducción al a programación
- Tipos de datos
- Constantes
- Operadores, expresiones y sentencias
- Sentencias de control de flujo
- Funciones y programación estructurada
- Vectores y punteros
- Estructuras y uniones
- Entrada y salida estándar
- Entrada y salida por archivos
- Asignación dinámica de memoria

ACTIVIDADES

Para la realización de los programas puede utilizarse cualquier editor de texto o, incluso, cualquier IDE que permita la programación en ANSI C. En caso de utilizarse el compilador GCC, presente en todos los entornos Unix y Linux, las órdenes básicas a utilizar serán

- Para obtener ayuda: `gcc --help`
- Para compilar: `gcc ArchivoFuente.c -o ArchivoSalida`

Enunciados propuestos:

1. Programa que, tras recibir por teclado el número de filas a mostrar (con el límite adecuado), presenta en pantalla un triángulo como el siguiente:

```

          1
         1 2
        1 2 3
       1 2 3 4
      1 2 3 4 5
     1 2 3 4 5 6
    1 2 3 4 5 6 7
   1 2 3 4 5 6 7 8
  1 2 3 4 5 6 7 8 9
 1 2 3 4 5 6 7 8 9 10
```

2. Programa que recibe del teclado un conjunto de números y, al finalizar, los muestra en pantalla ordenados según un criterio previamente seleccionado (ascendente/descendente) y permitiendo elegir el algoritmo de ordenación de entre un mínimo de tres algoritmos distintos.

3. Programa que realiza la suma de dos matrices de las mismas dimensiones, cuyos valores se introducen desde el teclado, y muestra el resultado en pantalla.
4. Programa que multiplica dos matrices cuyos elementos se introducen por teclado y muestra el resultado en pantalla
5. Programa que muestra en pantalla el llamado *Triángulo de Pascal* en el que cada número resulta de sumar los dos que hay encima de él, tal como se muestra en el siguiente gráfico

```

                1
              1 1
            1 2 1
          1 3 3 1
        1 4 6 4 1
       1 6 10 6 1
      1 10 15 10 4 1
     1 15 22 15 10 4 1
    1 21 30 21 15 10 4 1
   1 28 36 28 21 15 10 4 1
  1 36 46 36 28 21 15 10 4 1
 1 46 56 46 36 28 21 15 10 4 1

```

6. Programa que cuenta los bytes contenidos en un archivo.
7. Programa que muestra por pantalla el contenido de un archivo esperando la pulsación de una tecla cada vez que se hayan mostrado un número de líneas previamente introducido por teclado.
8. Programa que obtiene datos enteros del teclado a discreción del usuario y, al finalizar, almacene en un archivo de texto todos los valores introducidos así como un informe que indique el número de datos introducidos, el mayor valor, el menor y el número de repeticiones (en caso de que existan) de cada dato.
9. Programa que procesa el contenido de un archivo de texto existente y uno nuevo con el mismo texto pero habiendo eliminando todas las vocales.
10. Programa que lee un archivo de texto cuyas líneas tienen el formato *Operando1 Operando2 Operador* y, tras realizar las operaciones indicadas por el término *Operador*, crea un nuevo archivo con el mismo número líneas y un formato de línea similar al siguiente *Operando1 Operando2 Operador = Resultado*, considerando que todos los datos son números reales.
11. Programa que busca en un archivo el número de veces que se repite en el archivo una determinada secuencia de caracteres previamente introducida por teclado.
12. Programa que lee el contenido del archivo en paquetes de 64 bytes y en otro archivo cada paquete de 64 bytes en una línea que finaliza con la suma de comprobación (*checksum*) de esos 64 bytes.

EJEMPLOS Y ACTIVIDADES RESUELTAS

Programa que cuenta los bytes contenidos en un archivo (Propuesta nº 6)

```
/* CUENTA LOS BYTES QUE SE ALMACENAN DE UN ARCHIVO CUYO NOMBRE SE RECIBE
COMO ARGUMENTO. */

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

int main(int argc, char *argv[])
{
    FILE *fich;
    long cont = 0;

    if (argc != 2)
        /*Dos argumentos: Nombre del programa y nombre del archivo */
        {
            puts("\nLlamada incorrecta. Debe teclear:");
            puts("NOMBRE_PROGRAMA nombre_archivo");
            puts("\nFin del programa");
            getc(stdin);
            exit(1);
        }

    if ((fich = fopen(argv[1], "rb")) == NULL)
        /* Apertura del archivo origen para lectura binaria */
        {
            puts("No es posible abrir el archivo.");
            puts("\nFIN DEL PROGRAMA");
            exit(1);
        }

    while((fgetc(fich))!= EOF ) cont++; /* Cuenta bytes hasta llegar al
                                         fin de fichero */

    fclose(fich);          /* Cierre de los archivos abiertos */

    printf("\nEl archivo \"%s\" almacena %ld bytes.", argv[1], cont);
    puts("No se cuenta el EOF pero se incluye un byte por línea.\n\n");
    return(0);
}
```