

En esta práctica se va a utilizar matrices, ficheros, operadores de bits, argumentos del main y compilación por separado.

Básicamente consiste en rotar los bits de los elementos de una matriz de MxN letras mayúsculas (siendo M y N constantes definidas previamente): los dos bits de menor peso (más a la derecha) pasarán a ser los dos de mayor peso (más a la izda). Para realizar la práctica se deberán implementar las siguientes funciones:

- **Imprimir\_mat**, que recibiendo como parámetro una matriz como la mencionada, la imprima en pantalla.
- **Rellenar\_mat**, que recibiendo como parámetro una matriz de MxN letras mayúsculas, la rellene con valores generados aleatoriamente.
- **Rotar\_mat**, que pasándole como parámetro una matriz como la mencionada, rote los dos bits de más a la dcha de cada entero de la matriz, de forma que estos dos pasen a ser los de más a la izda después de desplazar el resto dos posiciones hacia la dcha.
- **Desrotar\_mat**, que pasándole como parámetro una matriz como la mencionada, haga lo contrario que la función anterior, es decir, devuelva los dos bits movidos a su posición original.

Todas estas funciones se implementarán en un fichero .c de compilación por separado. Además se implementará un fichero .h en el que se definirán las constantes M y N y se incluirán los prototipos de las funciones descritas.

Por último, se realizará un programa principal que recibirá como parámetro el nombre de un fichero y realizará las siguientes acciones:

- a) Abrir el fichero.
- b) Rellenar la matriz.
- c) Imprimir la matriz.
- d) Grabar la matriz en el fichero.
- e) Rotar los bits de cada elemento de la matriz como se ha explicado anteriormente.
- f) Imprimir la matriz modificada.
- g) Grabar la matriz en el fichero.
- h) Desrotar los bits de cada elemento de la matriz como para devolverlos a su estado inicial.
- i) Imprimir la matriz.
- j) Cerrar el fichero.

Una posible ejecución para una matriz de 15x10 sería la siguiente:

```

P L U N Z P S Q Z S X D O V N
H N S Q U G T P N R T P E G P
Q L C D S G M E I E L O J P L
G F X U R Y M U M N S O U O J
H D G X T R C I K Z G E U E U
U X G W K P Y K K K V F P Y R
M F B F J M S E A V P X U L U
E I W M P B V F S C S Z A F I
A R K F R U V F U X F B B P P
Y C F O E V C M W F A C O B Y

U ð æ Q U ð S 1 æ P ð E É U
P ð E æ ð ð U æ ð - æ É É ¶ ¶
Q R 1 ð ¶ ð U æ É ð É ð P æ R
S æ E æ æ ð É Q P U ¶ - ð !! ð
U - ð 1 æ ¶ U É É É É U æ ¶ U ð
t ^ ð 1 æ S ð S ð R É É ð Q ð Q ð
D æ - U ð - 1 ð S ð É É U É æ
I !! ð ^ É ð S ð R Q !! É É ¶ !!
t ð É I ð ð S ¶ ð ð S ¶ Q ð ¶
¶ !! ð ð ð ¶ É T G É - ^ É U ð

P L U N Z P S Q Z S X D O V N
H N S Q U G T P N R T P E G P
Q L C D S G M E I E L O J P L
G F X U R Y M U M N S O U O J
H D G X T R C I K Z G E U E U
U X G W K P Y K K K V F P Y R
M F B F J M S E A V P X U L U
E I W M P B V F S C S Z A F I
A R K F R U V F U X F B B P P
Y C F O E V C M W F A C O B Y

```

## Solución:

Fichero .h

```
#ifndef ROTAR_H
#define ROTAR_H

#define N 10
#define M 15

void mat_imprimir (char[M][N]);
void mat_rellenar (char[M][N]);
void mat_rotar    (char[M][N]);
void mat_desrotar (char[M][N]);
void mat_toFile   (char[M][N], FILE*);
int  random       (int, int);

#endif // ROTAR_H
```

Fichero .c (de funciones)

```
#include <time.h>
#include <stdlib.h>
#include <stdio.h>
#include "rotar.h"

void mat_imprimir (char _matr[M][N])
{
    int i, j;
    printf("\n");
    for (i = 0; i < M; i++)
    {
        for (j = 0; j < N; j++)
            printf(" %c ", _matr[i][j]);
        printf("\n");
    }
}

void mat_rellenar (char _matr[M][N])
{
    int i, j;
    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            _matr[i][j] = random('A', 'Z');
}

void mat_rotar    (char _matr[M][N])
{
    /*
     * == Funcionamiento:
     *
     * - mask = 3:
     *    3 = 00000011
     */
}
```

```

* - char = 'A':
*   'A' = 01000001
*
* (1) ults = bits que queremos rotar (los 2 ultimos):
*   00000011 (mask)
*   & 01000001 (char)
*   -----
*   00000001 (ults)
*
* (2) Desplazamos ults a la izquierda:
*   00000001 << 6 = 01000000
*
* (3) Desplazamos char dos veces a la derecha:
*   01000001 >> 2 = 00010000
*
* (4) Ponemos los ultimos dos bits (ults) en las primeras dos posiciones de char
*   01000000 (ults)
*   | 00010000 (char)
*   -----
*   01010000 (char rotado)
*
*
* == Podemos calcularlo en una linea:
*
*   char = ( ( char & 3 ) << 6 ) | ( char >> 2 )
*           | |____(1)____|         | |____(3)____|
*           |_____(2)_____|
*           |_____(4)_____|
*
*/
int i, j;
for (i=0; i<M; i++)
    for(j=0; j<N; j++)
        _matr[i][j] = ( (_matr[i][j] & 3) << 6 ) | (_matr[i][j] >> 2);
}

void mat_desrotar (char _matr[M][N])
{
    /*
    * == Funcionamiento:
    *
    * - mask = 3 << 6 = 192:
    *   192 = 11000000
    *
    * - char rotado = 'A' rotado:
    *   'A' rotado = 01010000
    *
    * (1) prims = bits que queremos rotar (los 2 primeros):
    *   11000000 (mask)
    *   & 01010000 (char rotado)
    *   -----
    *   01000000 (prims)
    *
    * (2) Desplazamos prims a la derecha:
    *   01000000 >> 6 = 00000001
    *
    * (3) Desplazamos char rotado dos veces a la izquierda:
    *   01010000 << 2 = 01000000
    */

```

```

*
* (4) Ponemos los primeros dos bits (prims) en las ultimas dos posiciones de char rotado
*   00000001 (prims)
*   | 01000000 (char rotado)
*   -----
*   01000001 (char)
*
*
* == Podemos calcularlo en una linea:
*
*   char = ( ( char & 192 ) >> 6 ) | ( char << 2 )
*           | |____(1)_____| | |____(3)_____|
*           | |____(2)_____|
*           | |____(4)_____|
*
*/
int i, j;
for (i=0; i<M; i++)
    for(j=0; j<N; j++)
        _matr[i][j] = ( (_matr[i][j] & 192) >> 6 ) | (_matr[i][j] << 2);
}

void mat_toFile(char _matr[M][N], FILE* f_out)
{
    int i;
    for (i=0; i<M; i++)
        fputs(_matr[i], f_out);
}

int random (int _min, int _max)
{
    int temp;
    if (_min > _max)
    {
        temp = _max;
        _max = _min;
        _min = temp;
    }
    return (rand()%(_max - _min + 1)+_min);
}

```

Fichero .c (del main)

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

//Para mandar un nombre de fichero al programa, ejecutalo desde la terminal
int main(int argc, char* argv[])
{
    FILE* f_out;
    char mat[M][N];

    if (argc != 2)
    {
        printf("ERROR: Maximo dos argumentos");
        return 1;
    }

    if (!(f_out = fopen(argv[1], "w")))
    {
        printf("ERROR: No ha sido posible abrir/crear el fichero");
        return 2;
    }

    mat_rellenar (mat);
    mat_imprimir (mat);
    mat_toFile   (mat, f_out);
    mat_rotar    (mat);
    mat_imprimir (mat);
    mat_toFile   (mat, f_out);
    mat_desrotar (mat);
    mat_imprimir (mat);
    mat_toFile   (mat, f_out);

    fclose(f_out);

    return 0;
}
```