

# PRACTICA-7-RESUELTA-COMPLETA.pdf



**user\_2716437**



**Fundamentos de la Programación**



**1º Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingeniería Informática  
Universidad de Málaga**



**Que no te escriban poemas de amor  
cuando terminen la carrera**



*(a nosotros por  
suerte nos pasa)*

**WUOLAH**

# WUOLAH

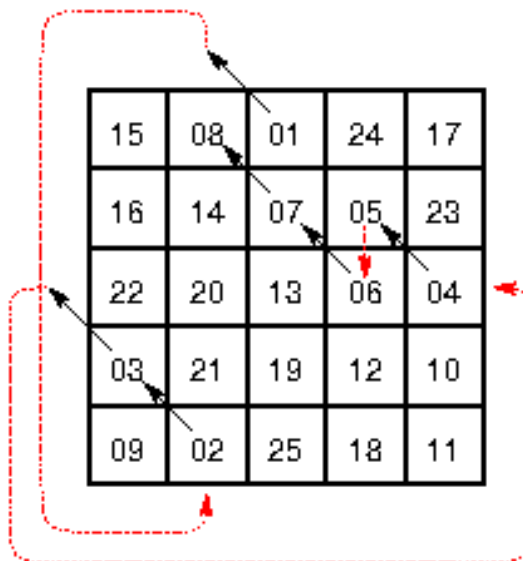
Oh Wuolah wuolithah  
Tu que eres tan bonita

15	08	01	24	17
16	14	07	05	23
22	20	13	06	04
03	21	19	12	10
09	02	25	18	11

## Ejercicios de Refuerzo

4. La regla para generar el cuadrado mágico (si  $N$  es impar) del ejercicio 3 es la siguiente:

*Se empieza poniendo un 1 en la casilla situada en la mitad de la primera fila. A partir de ese momento se inicia un proceso que visitará las diferentes casillas subiendo a la izquierda diagonalmente para ir poniendo en ellas los siguientes números de forma consecutiva hasta poner el número final  $N^2$ . Si en este proceso se pasa el borde superior o izquierdo del cuadrado, se continúa por el extremo opuesto (ejemplos: en la figura los pasos marcados desde 1 a 2 y desde 3 a 4). Por otro lado, si en este proceso se alcanza una casilla ya rellena, se baja una posición desde el último valor depositado (ejemplo: en la figura el paso desde 5 a 6).*



Implementa un programa que construya un cuadrado mágico para una constante N impar dada (por ejemplo, N = 5) y lo visualice por pantalla. Ejemplo de ejecución:

El cuadrado magico para  $N = 5$  es:

15	8	1	24	17
16	14	7	5	23
22	20	13	6	4
3	21	19	12	10
9	2	25	18	11

**Que no te escriban poemas de amor  
cuando terminen la carrera ▶▶▶▶▶▶▶▶**  
(a nosotros por suerte nos pasa) 😊

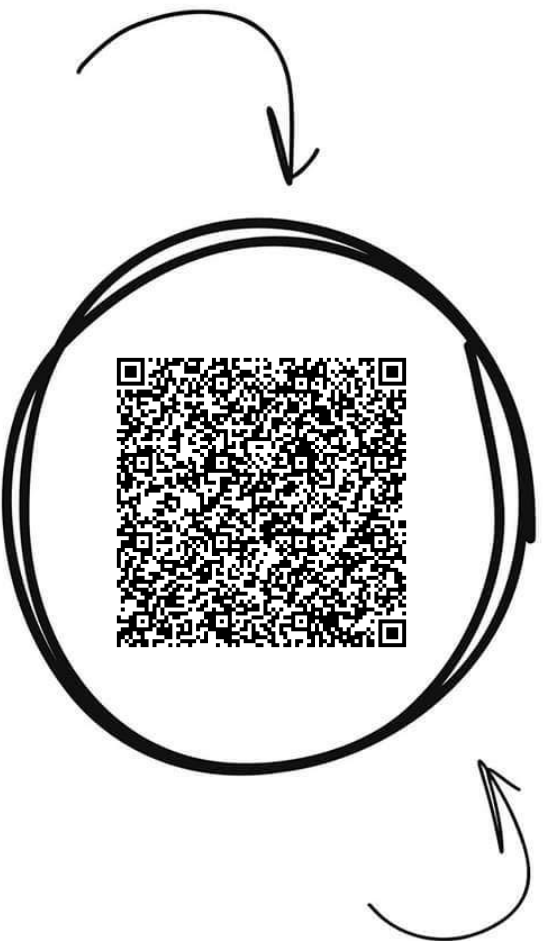


**WUOLAH**

# Fundamentos de la Programación



**Comparte estos flyers en tu clase y consigue más dinero y recompensas**



**Note bank of the**

**WUOLAH**



- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes
- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR

5. Escribe un programa que lea una sucesión de 10 números naturales, encuentre el valor máximo y lo imprima junto con el número de veces que aparece y las posiciones en que esto ocurre. El proceso se repite con el resto de la sucesión hasta que no quede ningún elemento por tratar.

Ejemplo de entrada: 7 10 143 10 52 143 72 10 143 7

Salida generada: 143 aparece 3 veces, en posiciones 3 6 9.  
72 se repite 1 vez, en posición 7.  
...  
7 aparece 2 veces, en posiciones 1 10.

6. Realiza los ejercicios 18 y 19 de la Tercera relación de problemas.

```

/*
Implementa un programa que lea por teclado una matriz bidimensional de
N×M elementos de
tipo entero (N y M son dos constantes definidas) y visualice por pantalla
el mayor elemento
de la misma junto con la posición (fila y columna) en la que se
encuentra. Si el elemento
mayor aparece varias veces, se deja a criterio del alumno cuál
seleccionar. Ejemplo de
ejecución:
Introduzca por filas una matriz 4 x 5:
3 5 -2 7 5
4 7 -4 9 8
3 2 5 9 12
8 4 12 11 8
El mayor de la matriz es: 12 que aparece en la posicion: [2][4]
*/

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

//fila
const int N = 4;
//columna
const int M = 5;

typedef array<int,N>TFilas;
typedef array<TFilas,M>TMatriz;

void leerMatriz(TMatriz& a);
void posicionMayor(TMatriz& a);

int main(){
    TMatriz a;
    cout<<"Introduzca por filas una matriz "<<N<<" x "<<M<<": "<<endl;
    leerMatriz(a);
    posicionMayor(a);
    return 0;
}

void leerMatriz(TMatriz& a){
    for(int f=0; f<N; f++){
        for(int c=0; c<M; c++){
            cin>>a[f][c];
        }
    }
}

void posicionMayor(TMatriz& a){
    int mayor = a[0][0];
    int fila = 0;
    int columna = 0;
    for(int f=0; f<N; f++){

```



(a nosotros por suerte nos pasa)

Oh Wuolah wuolithah  
Tu que eres tan bonita



```

/*
Implementa un programa que lea por teclado una matriz bidimensional de
N×N elementos de
tipo entero (N es una constante definida) y que indique por pantalla si
esa matriz es simétrica.
Una matriz es simétrica si  $a_{ij} = a_{ji}$  para todo i y j con valores 1, 2, 3,
4, ..., N. Ejemplo de
ejecución:
Introduzca por filas una matriz 4 x 4:
1 2 3 4
2 7 6 2
3 6 8 4
4 2 4 3
SI es simetrica
*/

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

const int N = 4;

typedef array<int,N>TFilas;
typedef array<TFilas,N>TMatriz;

void leerMatriz(TMatriz& m);
bool matrizSimetrica(TMatriz& m);

int main(){
    TMatriz m;
    cout<<"Introduzca por filas una matriz "<<N<<" x "<<N<<":"<<endl;
    leerMatriz(m);
    if(matrizSimetrica(m)){
        cout<<"SI es simetrica.";
    }else{
        cout<<"NO es simetrica.";
    }
    return 0;
}

void leerMatriz(TMatriz& m){
    for(int f=0; f<N; f++){
        for(int c=0; c<N; c++){
            cin>>m[f][c];
        }
    }
}

bool matrizSimetrica(TMatriz& m){
    bool ok = true;
    int i,j;
    i = 0;
    while (ok && (i < N)) {
        j = i+1;

```

```
    while (ok && (j < N)) {  
        if (m[i][j] != m[j][i]) {  
            ok = false;  
        }  
        j++;  
    }  
    i++;  
}  
return ok;  
}
```

```

/*
Un cuadrado mágico de orden N es una ordenación de los números desde el 1
hasta el N^2
en una matriz cuadrada (N×N) de manera que la suma de cada fila y de cada
columna y de las
dos diagonales principales produce el mismo valor. Por ejemplo, la
siguiente figura muestra
un cuadrado mágico de orden 5 (los números almacenados son del 1 al 25)
en el que todas las
filas, columnas y las dos diagonales suman 65. Diseña un programa que lea
por teclado una
matriz bidimensional de N×N números naturales (N es una constante
definida) e indique por
pantalla si es un cuadrado mágico.
15 08 01 24 17
16 14 07 05 23
22 20 13 06 04
03 21 19 12 10
09 02 25 18 11
*/

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

const int N = 5;
typedef array<int,N> TFilas;
typedef array<TFilas,N> TMatriz;
typedef array<bool,N*N+1> TRango;

void leerMatriz(TMatriz& m);
bool matrizCorrecta(TMatriz& m);
int sumaFilas(TMatriz& m, int f);
bool sumaFilaIgual(TMatriz& m);
int sumaColumnas(TMatriz& m, int c);
bool sumaColumnaIgual(TMatriz& m);
int sumaDiagonal1(TMatriz& m);
int sumaDiagonal2(TMatriz& m);
bool sumaDiagonalIgual(TMatriz& m);
bool esMagico(TMatriz& m);

int main(){
    TMatriz m;
    cout<<"Introduzca una matriz "<<N<<"x"<<N<<": "<<endl;
    leerMatriz(m);
    if(esMagico(m)){
        cout<<"Es magico.";
    }else{
        cout<<"No es magico.";
    }
    return 0;
}

void leerMatriz(TMatriz& m){

```

Que no te escriban poemas de amor  
cuando terminen la carrera ▶▶▶▶▶▶▶▶



WUOLAH

(a nosotros por suerte nos pasa)

No si antes decirte  
Lo mucho que te voy a recordar

Pero me voy a graduar.  
Mañana mi diploma y título he de  
pagar

Llegó mi momento de despedirte  
Tras años en los que has estado mi  
lado.

Siempre me has ayudado  
Cuando por exámenes me he  
agobiado

Oh Wuolah wuolilah  
Tu que eres tan bonita

```
for(int f=0; f<N; f++){
    for(int c=0; c<N; c++){
        cin>>m[f][c];
    }
}

bool matrizCorrecta(TMatriz& m){
    bool ok = true;
    TRango r = {false};
    int i=0;
    while(i<N && ok){
        int j=0;
        while(j<N && ok){
            if(m[i][j]>=1 && m[i][j]<=25 && !r[m[i][j]]){
                r[m[i][j]] = true;
            }else{
                ok = false;
            }
            j++;
        }
        i++;
    }
    return ok;
}

int sumaFilas(TMatriz& m, int f){
    int sol = 0;
    for(int c=0; c<N; c++){
        sol += m[f][c];
    }
    return sol;
}

bool sumaFilaIgual(TMatriz& m){
    int sumal = sumaFilas(m,0);
    int i=0;
    while(i<N && sumal==sumaFilas(m,i)){
        i++;
    }
    return i>=N;
}

int sumaColumnas(TMatriz& m, int c){
    int sol = 0;
    for(int f=0; f<N; f++){
        sol += m[f][c];
    }
    return sol;
}

bool sumaColumnaIgual(TMatriz& m){
    int sumal = sumaColumnas(m,0);
    int i=0;
```

WUOLAH

```

        while(i<N && suma1==sumaColumnas(m,i)) {
            i++;
        }
        return i>=N;
    }

    int sumaDiagonal1(TMatriz& m){
        int sol = 0;
        for(int i=0; i<N; i++){
            sol += m[i][i];
        }
        return sol;
    }

    int sumaDiagonal2(TMatriz& m){
        int sol = 0;
        for(int i=0; i<N; i++){
            sol += m[i][N-1-i];
        }
        return sol;
    }

    bool sumaDiagonalIgual(TMatriz& m){
        return sumaDiagonal1(m) == sumaDiagonal2(m);
    }

    bool esMagico(TMatriz& m){
        return sumaDiagonalIgual(m) == sumaColumnaIgual(m) ==
        sumaFilaIgual(m) && matrizCorrecta(m);
    }

```

```

/*
La regla para generar el cuadrado mágico (si N es impar) del ejercicio 3
es la siguiente:
Se empieza poniendo un 1 en la casilla situada en la mitad de la primera
fila. A partir de
ese momento se inicia un proceso que visitará las diferentes casillas
subiendo a la
izquierda diagonalmente para ir poniendo en ellas los siguientes números
de forma
consecutiva hasta poner el número final  $N^2$ . Si en este proceso se pasa
el borde superior
o izquierdo del cuadrado, se continúa por el extremo opuesto (ejemplos:
en la figura los
pasos marcados desde 1 a 2 y desde 3 a 4). Por otro lado, si en este
proceso se alcanza
una casilla ya rellena, se baja una posición desde el último valor
depositado (ejemplo:
en la figura el paso desde 5 a 6).

```

```

Implementa un programa que construya un cuadrado mágico para una
constante N impar dada
(por ejemplo, N = 5) y lo visualice por pantalla. Ejemplo de ejecución:
El cuadrado magico para N = 5 es:

```

```

15 8 1 24 17
16 14 7 5 23
22 20 13 6 4
3 21 19 12 10
9 2 25 18 11

```

```

*/

```

```

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

```

```

const int N = 5;
typedef array<int,N> TFilas;
typedef array<TFilas,N> TMatriz;

```

```

void mas(int &x);
void menos(int &x);
void sigCoordenada(TMatriz& m, int& f, int &c);
void inicializarMatriz(TMatriz& m);
void cuadradoMagico(TMatriz& m);
void imprimirCuadrado(const TMatriz& m);

```

```

int main(){
    TMatriz m;
    cout<<"El cuadrado magico para N = "<<N<<" es:"<<endl;
    cuadradoMagico(m);
    imprimirCuadrado(m);
}

```

```

void mas(int &x){
    if(x==N-1){

```

```

        x = 0;
    }else{
        x++;
    }
}

void menos(int &x){
    if(x!=0){
        x--;
    }else{
        x = N-1;
    }
}

void sigCoordenada(TMatriz& m, int& f, int &c){
    menos(f);
    menos(c);
    if(m[f][c]!=0){
        mas(f);
        mas(f);
        mas(c);
    }
}

void inicializarMatriz(TMatriz& m){
    for(int f=0; f<N; f++){
        for(int c=0; c<N; c++){
            m[f][c] = 0;
        }
    }
    m[0][N/2] = 1;
}

void cuadradoMagico(TMatriz& m){
    int f = 0;
    int c = N/2;
    inicializarMatriz(m);
    for(int i=2; i<=(N*N); i++){
        sigCoordenada(m,f,c);
        m[f][c] = i;
    }
}

void imprimirCuadrado(const TMatriz& m){
    for(int f=0; f<N; f++){
        for(int c=0; c<N; c++){
            cout<<m[f][c]<<' ';
        }
        cout<<endl;
    }
}

```

Que no te escriban poemas de amor  
cuando terminen la carrera ▶▶▶▶▶▶▶▶▶▶



WUOLAH

(a nosotros por suerte nos pasa)

No si antes decirte  
Lo mucho que te voy a recordar

Pero me voy a graduar.  
Mañana mi diploma y título he de  
pagar

Llegó mi momento de despedirte  
Tras años en los que has estado mi  
lado.

Siempre me has ayudado  
Cuando por exámenes me he  
agobiado

Oh Wuolah wuolah  
Tu que eres tan bonita

```
/*
Escribe un programa que lea una sucesión de 10 números naturales,
encuentre el valor máximo
y lo imprima junto con el número de veces que aparece y las posiciones en
que esto ocurre.
El proceso se repite con el resto de la sucesión hasta que no quede
ningún elemento por tratar.
Ejemplo de entrada: 7 10 143 10 52 143 72 10 143 7
Salida generada: 143 aparece 3 veces, en posiciones 3 6 9.
72 se repite 1 vez, en posición 7.
...
7 aparece 2 veces, en posiciones 1 10.
*/

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

const int N = 10;
typedef array<int,N> TArray;
struct TNumeros{
    TArray lista;
    int indice;
};

void leerDatos(TNumeros& a);
int mayor(TNumeros& a);
void posicion(TNumeros& a, int num);
void eliminarNum(TNumeros& a, int num);
int repeticiones(TNumeros& a, int num);
bool fin(TNumeros& a);
void solucion(TNumeros& a);

int main(){
    TNumeros a;
    cout<<"Introduzca "<<N<<" numeros naturales: ";
    leerDatos(a);
    solucion(a);
}

void leerDatos(TNumeros& a){
    for(int i=0; i<N; i++){
        cin>>a.lista[i];
    }
}

int mayor(TNumeros& a){
    int mayor = a.lista[0];
    for(int i=1; i<N; i++){
        if(mayor<a.lista[i]){
            mayor = a.lista[i];
        }
    }
    return mayor;
}
```

WUOLAH



```

}

void posicion(TNumeros& a, int num){
    for(int i=0; i<N; i++){
        if(a.lista[i]==num){
            cout<<" "<<i+1;
        }
    }
}

void eliminarNum(TNumeros& a, int num){
    for(int i=0; i<N; i++){
        if(a.lista[i]==num){
            a.lista[i] = -1;
        }
    }
}

int repeticiones(TNumeros& a, int num){
    int sol = 0;
    for(int i=0; i<N; i++){
        if(a.lista[i]==num){
            sol++;
        }
    }
    return sol;
}

bool fin(TNumeros& a){
    int cont = 0;
    for(int i=0; i<N; i++){
        if(a.lista[i]==-1){
            cont++;
        }
    }
    return cont>=N;
}

void solucion(TNumeros& a){
    while(!fin(a)){
        cout<<mayor(a)<<" aparece "<<repeticiones(a,mayor(a));
        if(repeticiones(a,mayor(a))==1){
            cout<<" vez, en posicion";
            posicion(a,mayor(a));
        }else{
            cout<<" veces, en posiciones";
            posicion(a,mayor(a));
        }
        cout<<"."<<endl;
        eliminarNum(a,mayor(a));
    }
}

```

si lees esto me debes un besito

WUOLAH

si lees esto me debes un besito

WUOLAH

(a nosotros por suerte nos pasa)

Oh Wuolah wuolilah  
Tu que eres tan bonita

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

si lees esto me debes un besito

WUOLAH

si lees esto me debes un besito

WUOLAH

si lees esto me debes un besito

WUOLAH