

# PRACTICA-6-COMPLETA-RESUELTA.pdf



**user\_2716437**



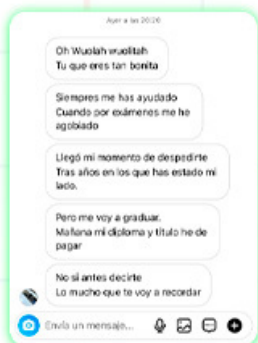
**Fundamentos de la Programación**



**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**

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



WUOLAH

(a nosotros por suerte nos pasa)



UNIVERSIDAD DE MÁLAGA  
DPTO. DE LENGUAJES Y CC. DE LA COMPUTACIÓN  
E.T.S. DE INGENIERÍA INFORMÁTICA

FUNDAMENTOS DE LA PROGRAMACIÓN  
E.T.S.I. Informática. Curso 1º

## Práctica Nº 6 Registros y arrays

### Ejercicios de clase.

1. Desarrolla una función `mayor` que tome como parámetro de entrada un array totalmente relleno de valores enteros y devuelva como resultado el valor mayor contenido en dicho array. Como tamaño del array puede tomarse el valor 10 (define una constante `TAM` para ello). Diseña también un procedimiento `leer` para rellenar el array con 10 números enteros que el usuario introducirá por teclado. Finalmente, crea una función `main` para comprobar el correcto funcionamiento de la función y el procedimiento diseñados. Ejemplo de ejecución:

```
Introduzca 10 numeros enteros: 7 -12 4 53 -8 4 1 65 12 7
```

```
El mayor del array es: 65
```

2. Diseña una función `media` para calcular la media de las estaturas de una clase. La función recibe como parámetro de entrada una estructura de tipo `TEstaturas`, con las estaturas (en centímetros) de los alumnos de una determinada clase. Define dicho tipo de forma que pueda almacenar hasta un máximo de `MAX = 50` estaturas. El tipo debe definirse de forma que se pueda controlar en todo momento la cantidad de estaturas almacenadas. Crea después otras dos funciones para determinar cuántos alumnos son más altos y cuántos más bajos que la media. El nombre de las funciones queda a criterio del alumno. Diseña también un procedimiento `leer` para tomar de teclado las estaturas que el usuario desee introducir. Finalmente, crea una función `main` para comprobar que todo se ha codificado correctamente. Ejemplo de ejecución:

```
Cuantas estaturas va a introducir (maximo 50): 5
```

```
Introduzca las 5 estaturas: 183 172 156 195 185
```

```
La media es: 178.2
```

```
Numero de alumnos mas altos que la media: 3
```

```
Numero de alumnos mas bajos que la media: 2
```

No si antes decirte  
Mañana mi diploma y título he de  
pagar

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 wuoliah  
Tu que eres tan bonita

WUOLAH

3. Diseña un algoritmo que lea de teclado una secuencia de números entre 0 y 9 y cuente el número de veces que se repite cada dígito. La secuencia de números de entrada se da por finalizada al leer un número negativo y a priori no se puede determinar cuántos números contiene. Ejemplo de ejecución:

```
Introduzca una secuencia de digitos (negativo termina):  
1 8 7 3 4 8 5 9 5 0 0 4 8 4 5 3 2 8 -1
```

```
La frecuencia de cada digito es:
```

```
0: 2  
1: 1  
2: 1  
3: 2  
4: 3  
5: 3  
6: 0  
7: 1  
8: 4  
9: 1
```

### Ejercicios de refuerzo.

4. Un histograma es una gráfica que muestra la frecuencia con que aparecen en una lista dada los distintos valores que la pudieran formar. Por ejemplo, si los valores de una lista pueden estar comprendidos entre 0 y 9, y la lista es:

6 4 4 1 9 7 5 6 4 2 3 9 5 6 4

entonces su histograma es:

```
          *  
          *      *  
          * * *      *  
      * * * * * * * *  
0 1 2 3 4 5 6 7 8 9
```

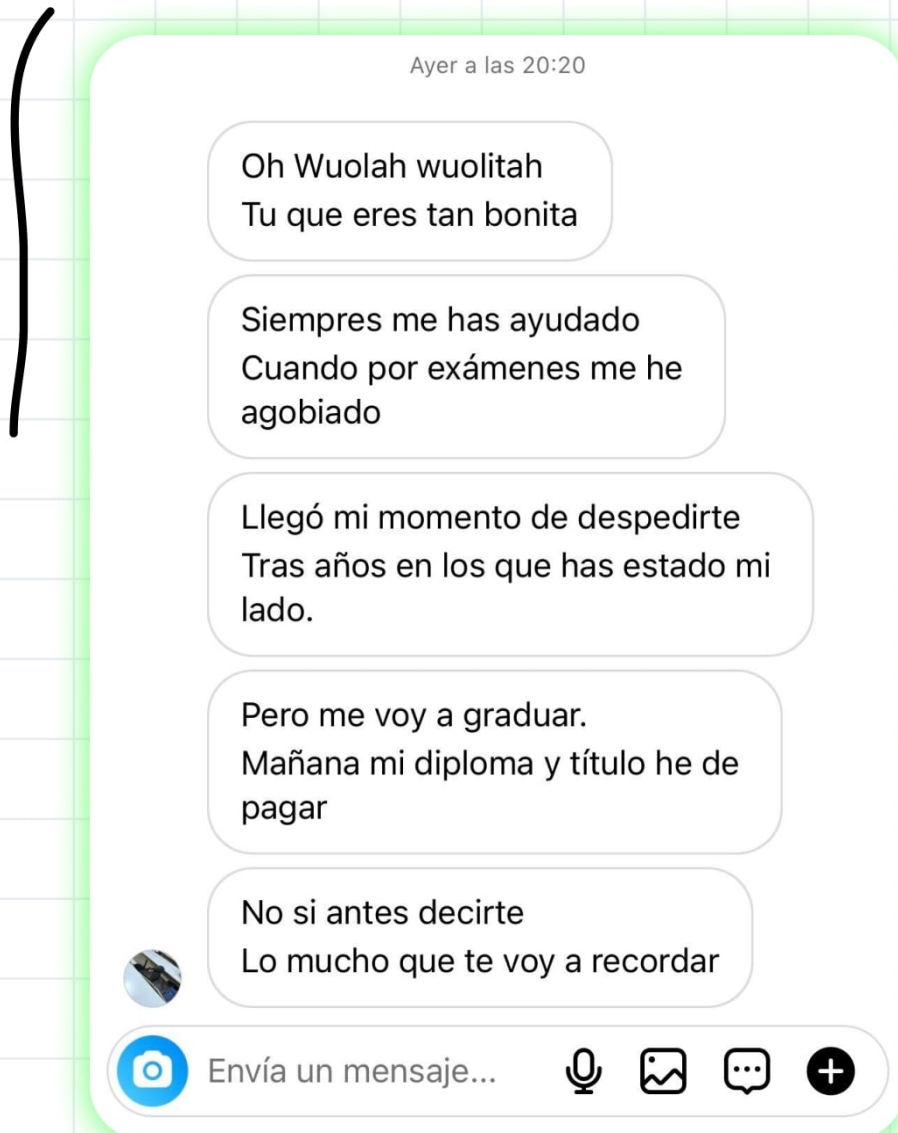
Esto indica que los dígitos 0 y 8 no aparecen ninguna vez, que 1, 2, 3 y 7 aparecen una vez, 5 y 9 dos veces, etc. Escribe un programa que lea una lista de números comprendidos entre 0 y 9 separados por espacios (la lista acabará cuando se lea un número negativo y a priori no se puede determinar cuántos números contiene) e imprima por pantalla un histograma como el anterior.

5. La denominada *Criba de Eratóstenes* es un método para determinar los números primos entre 1 y N, siguiendo los siguientes pasos:

- Se escriben los números naturales entre 1 y N.
- Se tacha el 1.
- Se deja el 2 y se tachan todos los demás números pares.
- Se deja el 3 y se tachan todos sus múltiplos.
- Como el 4 ya está tachado, pasamos al 5, que se deja y se tachan todos sus múltiplos (los del 5).
- Etc.

Así, cuando pasemos del 13, estarán tachados 14, 15 y 16, con lo que seguimos el proceso en el 17. El proceso acaba cuando llegamos a la raíz cuadrada de N. Los números que queden sin tachar, serán primos. La siguiente figura muestra este método aplicado del 1 al 100. Aparecen en blanco los números no marcados y que, por tanto, son primos.

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



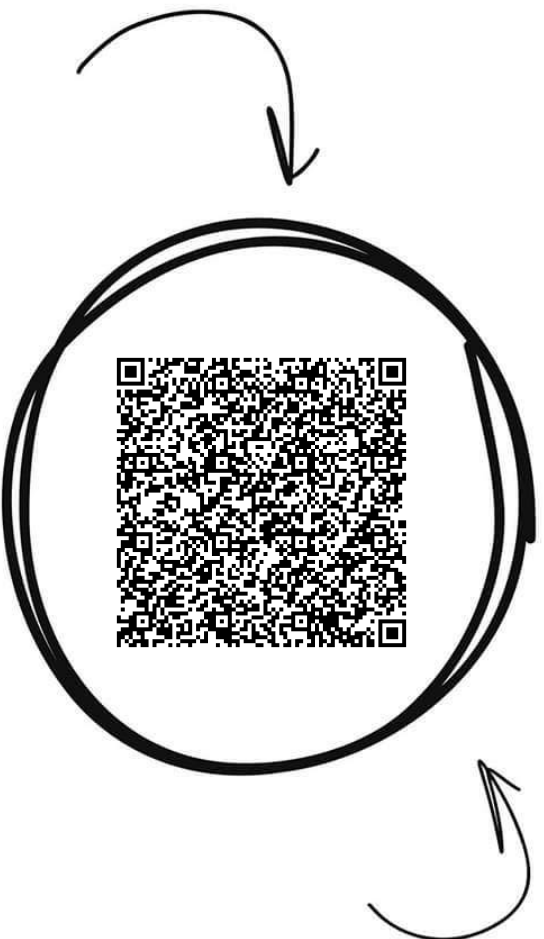
**WUOLAH**



# 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 escanear y acceder a apuntes

- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR



~~1~~ 2 3 4 5 6 7 8 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 20 21 22 23 24 25 26 27 28  
 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52  
 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76  
 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99  
~~100~~

Se pide crear un procedimiento denominado `eratostenes` que, mediante la criba descrita y haciendo uso de arrays, debe tomar como parámetro un natural `N` (mayor que 0 y menor o igual que una constante dada `MAX`) e imprimir por pantalla todos los números primos del 1 al `N`. Crea también una función `main` para comprobar que el procedimiento se ha codificado correctamente. Ejemplo de ejecución:

```
Introduzca el limite para calcular los primos (> 0 y <= 100): 50
```

```
Los numeros primos menores o iguales que 50 son:
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
```

- Realizar los ejercicios 10 y 11 de la Tercera relación de problemas.

```

/*
Desarrolla una función mayor que tome como parámetro de entrada un array
totalmente
relleno de valores enteros y devuelva como resultado el valor mayor
contenido en dicho
array. Como tamaño del array puede tomarse el valor 10 (define una
constante TAM para
ello). Diseña también un procedimiento leer para rellenar el array con 10
números enteros
que el usuario introducirá por teclado. Finalmente, crea una función main
para comprobar
el correcto funcionamiento de la función y el procedimiento diseñados.
Ejemplo de
ejecución:
Introduzca 10 numeros enteros: 7 -12 4 53 -8 4 1 65 12 7
El mayor del array es: 65
*/

```

```

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

const int MAX_TAM = 10;
typedef array<int,MAX_TAM> TArray;

void leer(TArray& a);
int mayor(TArray& a);

int main(){
    TArray a = {};
    leer(a);
    cout<<mayor(a);
    return 0;
}

void leer(TArray& a){
    cout<<"Introduzca "<<MAX_TAM<<" numeros enteros: ";
    for(int i=0; i<MAX_TAM; i++){
        cin>>a[i];
    }
}

int mayor(TArray& a){
    int mayor = a[0];
    for(int i=1; i<MAX_TAM; i++){
        if(mayor<a[i]){
            mayor = a[i];
        }
    }
    return mayor;
}

```

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

```
/*
Diseña una función media para calcular la media de las estaturas de una
clase. La función
recibe como parámetro de entrada una estructura de tipo TEstaturas, con
las estaturas
(en centímetros) de los alumnos de una determinada clase. Define dicho
tipo de forma que
pueda almacenar hasta un máximo de MAX = 50 estaturas. El tipo debe
definirse de forma
que se pueda controlar en todo momento la cantidad de estaturas
almacenadas. Crea después
otras dos funciones para determinar cuántos alumnos son más altos y
cuántos más bajos que
la media. El nombre de las funciones queda a criterio del alumno. Diseña
también un
procedimiento leer para tomar de teclado las estaturas que el usuario
desea introducir.
Finalmente, crea una función main para comprobar que todo se ha
codificado
correctamente. Ejemplo de ejecución:
Cuántas estaturas va a introducir (maximo 50): 5
Introduzca las 5 estaturas: 183 172 156 195 185
La media es: 178.2
Numero de alumnos mas altos que la media: 3
Numero de alumnos mas bajos que la media: 2
*/

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

const int MAX = 50;
typedef array<int,MAX> TArray;

struct TEstaturas{
    TArray estaturas;
    int ind;
};

void leerDatos(TEstaturas& e);
double media(const TEstaturas& e);
int masBajos(const TEstaturas& e, double media);
int masAltos(const TEstaturas& e, double media);

int main(){
    TEstaturas e = {};
    leerDatos(e);
    double med = media(e);
    cout<<"La media es: "<<med<<endl;
    cout<<"Numero de alumnos mas altos que la media:
"<<masAltos(e,med)<<endl;
    cout<<"Numero de alumnos mas bajos que la media: "<<masBajos(e,med);
    return 0;
}
```

WUOLAH



```

void leerDatos(TEstaturas& e){
    cout<<"Cuántas estaturas va a introducir (máximo "<<MAX<<" ): ";
    cin>>e.ind;
    while(e.ind>50 || e.ind<0){
        cout<<"Cuántas estaturas va a introducir (máximo "<<MAX<<" ): ";
        cin>>e.ind;
    }
    cout<<"Introduzca las "<<e.ind<<" estaturas: ";
    for(int i=0; i<e.ind; i++){
        cin>>e.estaturas[i];
    }
}

double media(const TEstaturas& e){
    double suma = 0;

    for(int i=0; i<e.ind; i++){
        suma += e.estaturas[i];
    }

    return suma/e.ind;
}

int masBajos(const TEstaturas& e, double media){
    int sol = 0;
    for(int i=0; i<e.ind; i++){
        if(media>e.estaturas[i]){
            sol++;
        }
    }
    return sol;
}

int masAltos(const TEstaturas& e, double media){
    int sol = 0;
    for(int i=0; i<e.ind; i++){
        if(media<e.estaturas[i]){
            sol++;
        }
    }
    return sol;
}

```

```

/*
Diseña un algoritmo que lea de teclado una secuencia de números entre 0 y
9 y cuente el
número de veces que se repite cada dígito. La secuencia de números de
entrada se da por
finalizada al leer un número negativo y a priori no se puede determinar
cuántos números
contiene. Ejemplo de ejecución:
Introduzca una secuencia de digitos (negativo termina):
1 8 7 3 4 8 5 9 5 0 0 4 8 4 5 3 2 8 -1
La frecuencia de cada digito es:
0: 2
1: 1
2: 1
3: 2
4: 3
5: 3
6: 0
7: 1
8: 4
9: 1
*/

```

```

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

const int RANGO = 10;
typedef array<int,RANGO> TArray;

void calcularFrec(TArray& a);
void imprimir(const TArray& a);

int main(){
    TArray a = {0};
    calcularFrec(a);
    imprimir(a);
    return 0;
}

void calcularFrec(TArray& a){
    int n;
    cout<<"Introduzca una secuencia de digitos (negativo termina):
"<<endl;
    cin>>n;
    while(n>=0){
        if(n>=0 && n<=9){
            a[n]++;
        }
        cin>>n;
    }
}

void imprimir(const TArray& a){

```

```
cout<<"La frecuencia de cada digito es: "<<endl;
for(int i=0; i<RANGO; i++){
    cout<<i<<": "<<a[i]<<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

```
/*
Un histograma es una gráfica que muestra la frecuencia con que aparecen
en una lista dada
los distintos valores que la pudieran formar. Por ejemplo, si los valores
de una lista pueden
estar comprendidos entre 0 y 9, y la lista es:
6 4 4 1 9 7 5 6 4 2 3 9 5 6 4
entonces su histograma es:
```

```
      *
      * *
    * * *
  * * * * *
0 1 2 3 4 5 6 7 8 9
```

```
Esto indica que los dígitos 0 y 8 no aparecen ninguna vez, que 1, 2, 3 y
7 aparecen una vez, 5 y
9 dos veces, etc. Escribe un programa que lea una lista de números
comprendidos entre 0 y 9
separados por espacios (la lista acabará cuando se lea un número negativo
y a priori no se
puede determinar cuántos números contiene) e imprima por pantalla un
histograma como el
anterior.
*/
```

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

const int RANGO = 10;
typedef array<int,RANGO> TArray;

void calcularFrec(TArray& a);
int mayorFrec(const TArray& a);
void imprimir(TArray& a);
void asteriscos(TArray& a, int linea);

int main(){
    TArray a = {0};
    calcularFrec(a);
    imprimir(a);
    return 0;
}

void calcularFrec(TArray& a){
    int n;
    cout<<"Introduzca una secuencia de digitos (negativo termina):
"<<endl;
    cin>>n;
    while(n>=0){
        if(n>=0 && n<=9){
            a[n]++;
        }
        cin>>n;
    }
}
```

WUOLAH

```

}

int mayorFrec(const TArray& a){
    int mayor = a[0];
    for(int i=1; i<RANGO; i++){
        if(mayor<a[i]){
            mayor = a[i];
        }
    }
    return mayor;
}

void imprimir(TArray& a){
    int frecM = mayorFrec(a);
    for(int i=frecM; i>0; i--){
        astericos(a,i);
    }
    for(int i=0; i<RANGO; i++){
        cout<<i;
    }
}

void astericos(TArray& a, int linea){
    for(int i=0; i<RANGO; i++){
        if(a[i]>=linea){
            cout<<"*";
        }else{
            cout<<" ";
        }
    }
    cout<<endl;
}

```

```

/*
La denominada Criba de Eratóstenes es un método para determinar los
números primos
entre 1 y N, siguiendo los siguientes pasos:
□ Se escriben los números naturales entre 1 y N.
□ Se tacha el 1.
□ Se deja el 2 y se tachan todos los demás números pares.
□ Se deja el 3 y se tachan todos sus múltiplos.
□ Como el 4 ya está tachado, pasamos al 5, que se deja y se tachan todos
sus múltiplos
(los del 5).
□ Etc.
Así, cuando pasemos del 13, estarán tachados 14, 15 y 16, con lo que
seguimos el proceso
en el 17. El proceso acaba cuando llegamos a la raíz cuadrada de N. Los
números que
queden sin tachar, serán primos. La siguiente figura muestra este método
aplicado del 1 al
100. Aparecen en blanco los números no marcados y que, por tanto, son
primos.
x 2 3 x 5 x 7 x x x 11 x 13 x x x 17 x 19 x x x 23 x x x x x 29
x 31 x x x x x 37 x x x 41 x 43 x x x 47 x x x x x 53 x x x x x
59 x 61 x x x x x 67 x x x 71 x 73 x x x x x 79 x x x 83 x x x
x x 89 x x x x x x 97 x x x
Se pide crear un procedimiento denominado eratostenes que, mediante la
criba descrita
y haciendo uso de arrays, debe tomar como parámetro un natural N (mayor
que 0 y menor o
igual que una constante dada MAX) e imprimir por pantalla todos los
números primos del 1 al
N. Crea también una función main para comprobar que el procedimiento se
ha codificado
correctamente.
Ejemplo de ejecución:
Introduzca el limite para calcular los primos (> 0 y <= 100): 50
Los numeros primos menores o iguales que 50 son:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
*/

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

const int MAX = 100;
typedef array<bool,MAX+1> TArray;
struct TPrimos{
    TArray nums;
    int ind;
};

void inicializar(TPrimos& a, int n);
void multiplos(TPrimos& a, int primo);
void primo(TPrimos& a);
void eratostenes(int n);

```

```

void imprimir(const TPrimos& a);

int main(){
    int n;
    cout<<"Introduzca el limite para calcular los primos (>0 y <=
"<<MAX<<"): ";
    cin>>n;
    while(n<0 || n>MAX){
        cout<<"Introduzca el limite para calcular los primos (>0 y <=
"<<MAX<<"): ";
        cin>>n;
    }
    eratostenes(n);
    return 0;
}

void inicializar(TPrimos& a, int n){
    if(n>MAX){
        a.ind=MAX;
    }else{
        a.ind=n;
    }
    a.nums[1] = false;
    for(int i=2; i<MAX+1; i++){
        a.nums[i] = true;
    }
}

void multiplos(TPrimos& a, int primo){
    for(int i=primo*2; i<=a.ind; i+=primo){
        a.nums[i] = false;
    }
}

void primo(TPrimos& a){
    for(int i = 2; i<sqrt(a.ind); i++){
        if(a.nums[i]){
            multiplos(a,i);
        }
    }
}

void eratostenes(int n){
    TPrimos a;
    inicializar(a, n);
    primo(a);
    imprimir(a);
}

void imprimir(const TPrimos& a){
    cout<<"Los numeros primos menores o iguales que "<<a.ind<<" son: ";
    for(int i=0; i<a.ind; i++){
        if(a.nums[i]){
            cout<<i<<" ";
        }
    }
}

```



# WUOLAH

(a nosotros por suerte nos pasa)

} } }

«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 wuolithah  
Tu que eres tan bonita



```

/*
Vamos a trabajar con listas de números enteros de hasta un tamaño máximo
de MAX (una
constante cualquiera). Define el tipo de datos TLista para ello y diseña
un procedimiento
criba que recibe como parámetros de entrada una lista de números enteros
lista1 de tipo
TLista y un número natural x. El procedimiento devolverá como parámetro
de salida otra
lista lista2 de tipo TLista que contendrá sólo aquellos números de lista1
que están
repetidos x veces. En la lista lista2 no habrá elementos repetidos.
Diseña la función
principal (main) para probar el funcionamiento del procedimiento. Para
leer la lista de
números lista1, se le pedirá primero al usuario el número de valores que
va a introducir,
controlando que éste sea mayor que 0 y menor o igual que MAX. Ejemplo de
ejecución (MAX =
10):
Cuantos numeros desea introducir (maximo 10): 9
Introduzca 9 numeros: 1 3 4 3 1 3 0 -6 4
Introduzca el numero de repeticiones para realizar la criba: 2
La lista cribada es: 1 4
*/

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

const int MAX = 10;
typedef array<int,MAX> TArray;

struct TLista{
    TArray lista;
    int tam;
};

bool esta(const TLista& a, int b);
int repeticiones(const TLista& a, int b);
void criba(const TLista& l1, TLista& l2, int x);

int main(){
    TLista lista1, lista2;
    int x;

    cout<<"Cuantos numeros desea introducir (maximo "<<MAX<<"): ";
    cin>>lista1.tam;
    while(lista1.tam<0 || lista1.tam>MAX){
        cout<<"Cuantos numeros desea introducir (maximo "<<MAX<<"): ";
        cin>>lista1.tam;
    }

    cout<<"Introduzca "<<lista1.tam<<" numeros:";

```

```

        for(int i=0; i<lista1.tam; i++){
            cin>>lista1.lista[i];
        }

        cout<<"Introduzca el numero de repeticiones para realizar la criba:
";
        cin>>x;

        criba(lista1,lista2,x);
        cout<<"La lista cribada es: ";
        for(int i=0; i<lista2.tam; i++){
            cout<<lista2.lista[i]<<" ";
        }

    }

    bool esta(const TLista& a, int b){
        int cont = 0;
        while(cont<a.tam && b != a.lista[cont]){
            cont++;
        }
        return cont < a.tam;
    }

    int repeticiones(const TLista& a, int b){
        int sol = 0;
        for(int i=0; i<a.tam; i++){
            if(a.lista[i]==b){
                sol++;
            }
        }
        return sol;
    }

    void criba(const TLista& l1, TLista& l2, int x){
        l2.tam = 0;
        for(int i=0; i<l1.tam; i++){
            if(repeticiones(l1,l1.lista[i])==x && !esta(l2,l1.lista[i])){
                l2.lista[l2.tam] = l1.lista[i];
                l2.tam++;
            }
        }
    }
}

```

si lees esto me debes un besito

WUOLAH



# WUOLAH

(a nosotros por suerte nos pasa)

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 wuolithah  
Tu que eres tan bonita

si lees esto me debes un besito

WUOLAH

si lees esto me debes un besito

WUOLAH

si lees esto me debes un besito

WUOLAH