

Recopilacion-1o-Parcial-Anos-ant...



pablofa02



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)

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



UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y Ciencias de la Computación
E.T.S.I. Informática

Fundamentos de la Programación
Examen 1ª Convocatoria Ordinaria

31/01/20

Apellidos, Nombre:

Titulación:

Grupo:

Código PC usado:

NOTAS PARA LA REALIZACIÓN DEL EXAMEN:

- La solución se almacenará en la carpeta **EXAMENFEBFP**, dentro de **Documentos**. Si la carpeta ya existe, debe borrarse todo su contenido. En otro caso, debe crearse.
- Los nombres de los ficheros con la solución para los ejercicios 1, 2, 3 y 4 serán **ejercicio1.cpp**, **ejercicio2.cpp**, **ejercicio3.cpp** y **ejercicio4.cpp**, respectivamente.
- Al inicio del contenido de cada fichero deberá aparecer un comentario con **el nombre del alumno, titulación, grupo y código del equipo** que se está utilizando (cada dato en una línea diferente).
- Debe consultarse** el documento “**Obligaciones y Recomendaciones Estilo de Programación**”, disponible en el Campus Virtual de la asignatura, con objeto de tener en cuenta los puntos allí señalados en las soluciones a los ejercicios.
- Una vez terminado el examen, se subirán los ficheros ***.cpp** a la tarea creada en el **campus virtual** para ello.
- No está permitido:**
 - Utilizar documentación electrónica o impresa.
 - Intercambiar documentación con otros compañeros.

(1 pto) 1.- **Diseña un algoritmo** que lea de teclado una secuencia de longitud arbitraria de números enteros positivos separados por un espacio y terminada en 0, y muestre por pantalla el mayor de los números primos leídos. Un número primo es un número entero positivo mayor que 1 que es divisible únicamente por él mismo y por la unidad. En caso de que no haya ningún primo en la secuencia de entrada, se mostrará un mensaje indicándolo. Puedes suponer que la entrada es correcta.

Importante: La puntuación de este problema será de 1 punto sólo en el caso de que el algoritmo funcione correctamente. En otro caso, la puntuación será de 0 puntos.

Dos ejemplos de ejecución:

Introduzca una secuencia de enteros positivos acabada
en 0: 1 3 5 7 2 9 17 13 18 0

El mayor primo de la secuencia es: 17

Introduzca una secuencia de enteros positivos acabada
en 0: 1 9 15 8 12 0

No hay ningun primo en la secuencia

WUOLAH



UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y Ciencias de la Computación
E.T.S.I. Informática

Fundamentos de la Programación
Examen 2ª Convocatoria Ordinaria

02/09/20

Apellidos, Nombre:

Titulación:

Grupo:

Código PC usado:

NOTAS PARA LA REALIZACIÓN DEL EXAMEN:

- La solución se almacenará en la carpeta **EXAMENSEPFP**, dentro de **Documentos**. Si la carpeta ya existe, debe borrarse todo su contenido. En otro caso, debe crearse.
- Los nombres de los ficheros con la solución para los ejercicios 1, 2, 3 y 4 serán **ejercicio1.cpp**, **ejercicio2.cpp**, **ejercicio3.cpp** y **ejercicio4.cpp**, respectivamente.
- Al inicio del contenido de cada fichero deberá aparecer un comentario con **el nombre del alumno, titulación, grupo y código del equipo** que se está utilizando (cada dato en una línea diferente).
- **Debe consultarse** el documento “**Obligaciones y Recomendaciones Estilo de Programación**”, disponible en el Campus Virtual de la asignatura, con objeto de tener en cuenta los puntos allí señalados en las soluciones a los ejercicios.
- Una vez terminado el examen, se subirán los ficheros ***.cpp** a la tarea creada en el **campus virtual** para ello.
- **No está permitido:**
 - Utilizar documentación electrónica o impresa.
 - Intercambiar documentación con otros compañeros.

(1 pto) 1.- Diseña un algoritmo que lea de teclado una secuencia de longitud indefinida de números enteros positivos separados por un espacio y terminada en 0, y muestre por pantalla el *mayor de los números perfectos* leídos. Un *número perfecto* es aquel cuyo valor coincide con la suma de sus divisores (excepto él mismo). Por ejemplo, el 28 es un número perfecto porque $28 = 1+2+4+7+14$. En caso de que no haya ningún número perfecto en la secuencia de entrada, se mostrará un mensaje indicándolo. Puedes suponer que la entrada es correcta.

Importante: La puntuación de este problema será de 1 punto sólo en el caso de que el algoritmo funcione correctamente. En otro caso, la puntuación será de 0 puntos.

Dos ejemplos de ejecución:

Introduzca una secuencia de enteros positivos acabada en 0: 2 12 6 15 28 42 9 0

El mayor numero perfecto de la secuencia es: 28

Introduzca una secuencia de enteros positivos acabada en 0: 1 2 15 8 25 36 9 0

No hay ningun numero perfecto en la secuencia

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

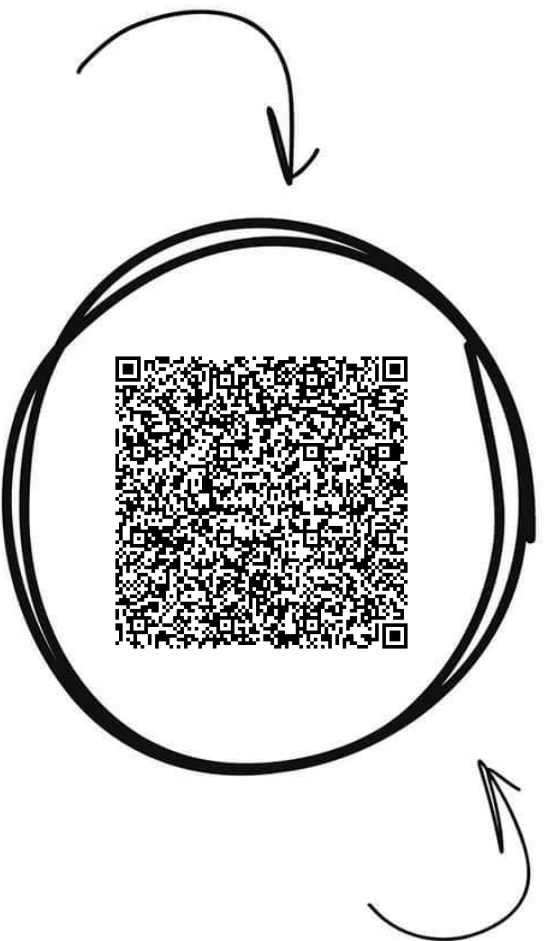


WUOLAH

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





UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y CC. Computación
E.T.S.I. Informática

Fundamentos de la Programación
Control de Evaluación
Problema Laboratorio

20/11/19

Apellidos, Nombre:

Grupo:

NOTAS PARA LA REALIZACIÓN DEL EXAMEN:

- La solución se almacenará en la carpeta CONTROL1_FP_GIIA, dentro de Documentos. Si la carpeta ya existe, debe borrarse todo su contenido. En otro caso, debe crearse.
- Los nombres de los ficheros con la solución para los ejercicios serán Ej1.cpp y Ej2.cpp.
- Al inicio del contenido de cada fichero deberá aparecer un comentario con el nombre del alumno, titulación, grupo y código del equipo que se está utilizando (cada dato en una línea diferente).
- Debe consultarse el documento "Obligaciones y Recomendaciones Estilo de Programación", disponible en el Campus Virtual de la asignatura, con objeto de tener en cuenta los puntos allí señalados en las soluciones a los ejercicios.
- Una vez terminado el examen, se subirán los ficheros *.cpp a la tarea creada en el campus virtual para ello.

No está permitido:

- Utilizar documentación electrónica o impresa.
- Intercambiar documentación con otros compañeros.
- Utilizar soportes de almacenamiento.
- Utilizar dispositivos electrónicos (móviles, tablets, ...)

1. **(0,4 puntos)** Los babilonios calculaban la raíz cuadrada de un número N con el siguiente método:

Paso 1: Se parte de una primera aproximación al resultado. Por ejemplo, se parte de una aproximación igual al número dividido entre 2 (aproximación = $N/2$).

Paso 2: Se calcula la raíz como el resultado de la división entre el número N y la aproximación calculada (raíz = $N/\text{aproximación}$).

Paso 3: Se calcula una nueva aproximación en función de la aproximación anterior y de la raíz calculada en el paso 2 (aproximación = $(\text{aproximación} + \text{raíz}) / 2$).

Los pasos 2 y 3 se repitan tantas veces como sea necesario. Cuantas más veces se repitan los pasos 2 y 3 más cerca estaremos de que el valor almacenado en "raíz" sea la raíz cuadrada del número N.

Escriba un programa que pida un número entero N y se asegure que sea mayor que 0. A continuación, escriba por pantalla la raíz cuadrada de ese número calculada según el algoritmo babilonio descrito anteriormente. Salvo el número N, el resto de valores calculados se almacenará en variables tipo double. El algoritmo parará cuando la diferencia entre las dos últimas aproximaciones calculadas sea menor de 0.001. **Se valorará el uso de subprogramas.**

Ejemplos de distintas ejecuciones del programa:

Introduzca un número mayor que 0: -1

Introduzca un número mayor que 0: 16

La raíz cuadrada del número 16 por el método babilonio es: 4

Introduzca un número mayor que 0: 121

La raíz cuadrada del número 16 por el método babilonio es: 10.9998

Introduzca un número mayor que 0: 17

La raíz cuadrada del número 16 por el método babilonio es: 4.1222

2. **(0,6 puntos)** Escriba un programa que lea una secuencia de números de longitud indefinida que terminará cuando se introduzca el valor -1. El programa mostrará por pantalla aquellos números que sean capicúa. Un número es capicúa si se puede leer igual de derecha a izquierda que de izquierda a derecha. Se valorará el uso de subprogramas. Un ejemplo de ejecución del programa sería:

Introduzca una secuencia de números terminada en -1: 7 121 23 404 12521 37 1221 -1

Los números capicúas son: 7 121 9 404 12521 1221



UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y CC. Computación
E.T.S.I. Informática

Fundamentos de la Programación
Control de Evaluación Diciembre 2013
Problema Laboratorio

Apellidos, Nombre:

Titulación:

Código PC usado:

Grupo:

Diseña un programa que dado un número natural N mayor que 0 muestre por pantalla un mensaje indicando si el número es *perfecto* o no. Además, debe mostrar por pantalla todos los divisores propios del número. Un *número perfecto* es un número natural igual a la suma de sus divisores propios. Un *divisor propio* de un número N es un número natural distinto de N que lo divide de forma exacta (es decir, sin resto).

Si el usuario introduce un número menor o igual a 0 el programa volverá a pedirle que introduzca un nuevo valor hasta que se cumpla que N sea mayor que 0.

A continuación se muestran dos ejemplos de la entrada/salida de la ejecución del programa:

Ejemplo 1:

```
Introduce un número N > 0: 6
El número 6 es perfecto.
Sus divisores propios son: 1, 2 y 3
1+2+3 == 6
```

Ejemplo 2:

```
Introduce un número N > 0: 0
Introduce un número N > 0: -1
Introduce un número N > 0: 8
El número 8 no es perfecto.
Sus divisores propios son: 1, 2 y 4
1+2+4 != 8
```

WUOLAH

Oh Wuolah wuolita
Tu que eres tan bonita



UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y CC. Computación
E.T.S.I. Informática

Fundamentos de Programación
Control de Evaluación 23 de Noviembre 2017
Problema Laboratorio

Apellidos, Nombre:
Titulación:
Código PC usado:

Grupo:

NOTAS PARA LA REALIZACIÓN DEL EXAMEN:

- La solución se almacenará en el directorio **C:\FP\IIBCONT1**. Si el directorio ya existe, debe borrarse todo su contenido. En otro caso, debe crearse.
- El primer ejercicio se llamará **ej1.cpp** y el segundo ejercicio **ej2.cpp**.
- Al inicio del contenido del fichero deberá aparecer un comentario con **el nombre del alumno, titulación, grupo y código del equipo** que se está utilizando (en líneas diferentes).
- Una vez terminado el examen, se subirán los dos ficheros **ej1.cpp** y **ej2.cpp**, sin comprimir, a la tarea creada en el **campus virtual** para ello.
- **No está permitido:**
 - Utilizar documentación electrónica o impresa.
 - Intercambiar documentación con otros compañeros.
 - Utilizar soportes de almacenamiento.
 - Utilizar dispositivos electrónicos (móviles, tablets, ...)

1. **(0.25 puntos)** Escriba un programa que lea dos números naturales *a* y *b*, que representan un intervalo [*a*..*b*], y muestre la suma de los números pares en el intervalo y la suma de los números impares en el intervalo. El programa volverá a leer los números *a* y *b* hasta que se cumpla que *a* sea menor que *b*. Por ejemplo:

```
Introduzca el intervalo: 10 1
Introduzca el intervalo: 1 10
Suma de los numeros pares: 30
Suma de los numeros impares: 25
```

2. **(0.75 puntos)** Implemente un programa que lea un carácter de teclado, que representa un día de la semana (*l* es lunes, *m* es martes, *x* es miércoles, *j* es jueves, *v* es viernes, *s* es sábado y *d* es domingo) y a continuación muestre en pantalla un calendario para un mes de 31 días, suponiendo que el día 1 de dicho mes cae el día de la semana introducido. Si el día indicado no es válido no se hará nada. Por ejemplo:

```
Dia que cae el primero del mes: j

.. .. .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
```



UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y CC. Computación
E.T.S.I. Informática

1) Implementa un programa que calcule la raíz n-esima entera de un número natural. El programa pedirá al usuario el número y el valor de n e imprimirá por pantalla dicha raíz entera. Dado **número** y **n**, la raíz será un número entero, x, tal que x^n es el mayor valor **menor o igual** que **número**.

Ejemplo. Dado el número 31 y el valor 2, el resultado debería ser 5:

$$31 > 2^2$$

$$31 > 3^3$$

$$31 > 4^4$$

$$31 > 5^5$$

$$31 < 6^6$$

La ejecución del programa sería la siguiente:

Introduzca el numero: 31

Introduzca el valor n: 2

La raíz n-esima de 31 es: 5

Ejemplo. Dado el número 28 y el valor 3 el, el resultado debería ser 3.

$$28 > 2^2 \cdot 2$$

$$28 > 3^3 \cdot 3$$

$$28 < 4^4 \cdot 4$$

La ejecución del programa sería la siguiente:

Introduzca el numero: 28

Introduzca el valor de n: 3

La raíz n-esima de 28 es: 3

2) Un número *narcisista*, es aquel, en el que la suma de las potencias k-esimas de sus k dígitos es igual a dicho número. Por ejemplo, los números 153, 8 o 1634 son números narcisistas ya que:

$$153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$$

$$8 = 8^1 = 8$$

$$1634 = 1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 256 = 1634$$

Sin embargo, 231 no lo es, ya que:

$$231 = 2^3 + 3^3 + 1^3 = 8 + 27 + 1 = 36$$

Escribe por teclado un programa que pida un número entero mayor que cero (el programa deberá asegurarse que el dato es correcto), y muestre un mensaje por pantalla indicando si el número es o no narcisista. A continuación, se muestran varios ejemplos de la ejecución del programa.

Introduzca un numero entero mayor que 0: 1634

El numero 1634 SI es un numero narcisista

Introduzca un numero entero mayor que 0: 231

El número 231 NO es un numero narcisista

```

#include <iostream>
using namespace std;

int main()
{
    int num, may=0, cont=0;

    cout << "Introduzca una secuencia de enteros positivos acabada en 0: ";
    cin>>num;

    while(num!=0)
    {
        cont=0;
        for(int i=2; i<=num; i++)
        {
            if(num%i==0)
            {
                cont++;
            }
        }

        if(cont==1)
        {
            if(num>may)
            {
                may=num;
            }
        }

        cin>>num;
    }

    if(may==0)
    {
        cout << "No hay ningun primo en la secuencia."<<endl;
    }
    else
    {
        cout << "El mayor primo de la secuencia es: "<<may<<endl;
    }

    return 0;
}

```

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 wuoliah
Tu que eres tan bonita

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int num, may=0, suma;
7
8      cout << "Introduzca una secuencia de enteros positivos acabada en 0: ";
9      cin>>num;
10
11     while(num!=0)
12     {
13
14         suma=0;
15
16         for(int i=1; num>i; i++)
17         {
18             if(num%i==0)
19             {
20                 suma=suma+i;
21             }
22         }
23
24         if(suma==num)
25         {
26             if(num>may)
27             {
28                 may=num;
29             }
30         }
31
32         cin>>num;
33     }
34
35     if (may==0)
36     {
37         cout << "No hay ningun numero perfecto en la secuencia."<<endl;
38     }
39     else
40     {
41         cout << "El mayor numero perfecto de la secuencia es: "<<may<<endl;
42     }
43
44     return 0;
45 }
```

WUOLAH

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6
7      int num;
8      double aprox, anterior, raiz;
9
10     do
11     {
12         cout << "Introduzca un numero mayor que 0: ";
13         cin>>num;
14     }
15     while (num<=0);
16
17     aprox=double(num)/2;
18     anterior=num;
19
20     while((anterior-aprox)>0.001)
21     {
22
23         anterior=aprox;
24         raiz=double(num)/aprox;
25         aprox=(aprox+raiz)/2;
26
27     }
28
29     cout << "La raiz cuadrada del numero "<<num<<" por el metodo babilonio es: "<<aprox<<endl;
30
31     return 0;
32 }
33

```



```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6
7      int num, numInv;
8      bool primer=true;
9
10     cout << "Introduzca una secuencia de numeros terminada en -1: ";
11     cin>>num;
12
13     while(num!=-1)
14     {
15         numInv=0;
16         for(int i=num; i>0; i=i/10)
17         {
18             numInv=(10*numInv)+(i%10);
19         }
20
21         if(num==numInv)
22         {
23             if(primer)
24             {
25                 primer=false;
26                 cout << "Los numeros capicuas son: ";
27             }
28             cout << num<<" ";
29         }
30
31         cin>>num;
32     }
33
34     return 0;
35 }
36
37

```

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int num, suma=0;
7      bool primer=true;
8
9      do
10     {
11         cout << "Introduce un numero N > 0: ";
12         cin>>num;
13     }
14     while (num<=0);
15
16     for(int i=1; num>i; i++)
17     {
18         if(num%i==0)
19         {
20             suma=suma+i;
21             if(primer)
22             {
23                 primer=false;
24                 cout << "Sus divisores propios son: ";
25             }
26             cout<<i<<" ";
27         }
28     }
29
30     cout<<endl<<endl;
31
32     if(suma==num)
33     {
34         cout << "El numero "<<num<<" es perfecto."<<endl;
35     }
36     else
37     {
38         cout << "El numero "<<num<<" no es perfecto." <<endl;
39     }
40
41     return 0;
42 }
43

```

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 wuoliah
Tu que eres tan bonita

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int num, sumarDigitos;
7      bool primer=true;
8
9      cout << "Introduzca secuencia terminada en -1: ";
10     cin>>num;
11
12     while(num!=-1){
13         sumarDigitos=0;
14         for(int i=num; i>0; i=i/10){
15             sumarDigitos=sumarDigitos+(i%10);
16         }
17
18         if(sumarDigitos==15){
19             if(primer){
20                 primer=false;
21                 cout << "Los numeros que cumplen la condicion son: ";
22             }
23             cout<<num<<" ";
24         }
25
26         cin>>num;
27     }
28
29
30
31     return 0;
32 }
33
```

WUOLAH

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6
7      int a, b, sumaPar=0, sumaImpar=0;
8
9      do
10     {
11         cout << "Introduzca el intervalo: ";
12         cin>>a>>b;
13     } while (a>=b);
14
15     for(int i=b; a<=i; i--){
16         if(i%2==0){
17             sumaPar=sumaPar+i;
18         }else{
19             sumaImpar=sumaImpar+i;
20         }
21     }
22
23     cout << "Suma de los numeros pares: "<<sumaPar<<endl;
24     cout << "Suma de los numeros impares: "<<sumaImpar<<endl;
25
26     return 0;
27 }
28

```

C:\Users\pablo\OneDrive\Docur

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int num, impar=-1, cubo, cont;
7
8      do
9      {
10         cout << "Introduzca numero > 0: ";
11         cin>>num;
12     }
13     while (num<=0);
14
15     cout <<"Los " <<num<< " primeros cubos son: " <<endl;
16
17     for(int i=1; num>=i; i++)
18     {
19         cont=1;
20         cubo=0;
21         cout << i << " al cubo = ";
22
23         for(int x=impar+2; cont<=i; x=x+2)
24         {
25             cont++;
26             cubo=cubo+x;
27             impar=x;
28         }
29
30         cout << cubo<<endl;
31     }
32
33     return 0;
34 }
35
36

```



```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      char dia;
7      int primer=10;
8      bool primero=true;
9
10     cout << "Dia que cae el primero del mes: ";
11     cin>>dia;
12
13     switch (dia)
14     {
15     case 'l':
16         primer=0;
17         break;
18     case 'm':
19         primer=1;
20         break;
21     case 'x':
22         primer=2;
23         break;
24     case 'j':
25         primer=3;
26         break;
27     case 'v':
28         primer=4;
29         break;
30     case 's':
31         primer=5;
32         break;
33     case 'd':
34         primer=6;
35         break;
36     }
37
38     for(int i=1; i<=31; i++)
39     {
40         if(primer)
41         {
42             for(int x=1; primer>=x; x++)
43             {
44                 cout<< ".. ";
45             }
46         }
47
48         primero=false;
49
50         if(i<10)
51         {
52             cout<<". ";
53         }
54
55         cout<<i<<" ";
56
57         if((primer+i)%7==0)
58         {
59             cout << endl;
60         }
61     }
62
63     return 0;
64 }
65

```



WUOLAH

No si antes decirte
Lo mucho que te voy a recordar

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

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 wuolita
Tu que eres tan bonita

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int num, n, fin=0;
7      float raiz;
8
9      cout << "Introduzca el numero: ";
10     cin>>num;
11
12     cout << "Introduzca el valor n: ";
13     cin>>n;
14
15
16     for(int i=2; num>=raiz; i++)
17     {
18         raiz=i;
19
20         for (int x=1; x<n; x++)
21         {
22             raiz=raiz*i;
23         }
24
25         fin=i-1;
26     }
27
28     cout << "La raiz n-esima de "<<num<< " es: "<<fin<<endl;
29
30
31     return 0;
32 }
33
34

```

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int num, k=0, suma=0, numero, cifra;
7
8      do
9      {
10         cout << "Introduzca un numero entero mayor que 0: ";
11         cin>>num;
12     }
13     while (num<=0);
14
15     for(int i=num; i>0; i=i/10)
16     {
17         k++;
18     }
19
20     for(int x=num; x>0; x=x/10)
21     {
22         cifra=x%10;
23         numero=cifra;
24
25         for(int z=1; z<k; z++)
26         {
27             numero=numero*cifra;
28         }
29
30         suma=suma+numero;
31     }
32
33     if(suma==num)
34     {
35         cout << "El numero "<<num<<" SI es un numero narcisista."<<endl;
36     }
37     else
38     {
39         cout << "El numero "<<num<<" NO es un numero narcisista. "<<endl;
40     }
41
42
43
44
45     return 0;

```

si lees esto me debes un besito

WUOLAH