

WUOLAH



jemonra

www.wuolah.com/student/jemonra



9429

Ejercicio3BPractica4.pdf

Práctica 4 - EXPLICADA



1º Fundamentos de la Programación



Grado en Ingeniería Informática



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



academias

UNITEC

Especializada en Ingenierías

PRIMER CUATRIMESTRE en Unitec

Presencial y online ¡tú eliges!

Del 5 de octubre al 12 de febrero



Síguenos en Instagram para mantenerte informado.
academias_unitec



www.academiasunitec.com



info@academiasunitec.com



95 2345678



C/ Eolo, 3. 29010



C/ Teseo, 9. 29010

EJERCICIO 3B PRÁCTICA 4

EJERCICIO SIN COMENTARIOS

```
/* EJERCICIO 3B DE LA PRÁCTICA 4
```

Dos números a y b se dice que son amigos si la suma de los divisores de a (salvo él mismo)

coincide con b y viceversa (la suma de los divisores de b coincide con a). Por ejemplo, los

números 220 y 284 son amigos.

B. Diseña un programa que tenga como entrada de teclado dos números naturales (n y m) y que muestre en la pantalla todas las parejas de números amigos que existan en el intervalo determinado por n y m . El programa debe asegurarse de que n sea menor que m . Por ejemplo, en el intervalo 1 - 2000 sólo existen dos parejas de amigos: el 220 es amigo del 284, y el 1184 es amigo del 1210.

Autor: Jesús Moncada Ramírez

Fecha: 13/11/2019

```
*/
```

```
#include <iostream>
```

```
using namespace std;
```

```
void leerNumeros(int&, int&);
```

```
int sumaDivisores(int);
```

```
bool sonAmigos(int, int);
```

```
int main()
```

```
{
```

```
    int n, m;
```

```
    leerNumeros(n, m);
```

```
    for (int x = n; x<=m; x++)
```

```
    {
```

```
        int y = sumaDivisores(x);
```

```
        if (y>x)
```

```
        {
```

```
            if(x==sumaDivisores(y))
```

```

        {
            cout << "Numeros amigos encontrados: " << x << ", " << y << endl;
        }
    }
}

return 0;
}

void leerNumeros(int& n, int& m)
{
    do{
        cout << "Introduce incio del intervalo, n= ";
        cin >> n;
        cout << "Introduce final del intervalo, (> n) m= ";
        cin >> m;

        } while(m<=n);
    }

int sumaDivisores(int num)
{
    int suma = 0;
    for(int x = 1; x<num; x++)
    {
        if(num%x == 0)
        {
            suma += x;
        }
    }
    return suma;
}

```

EJERCICIO CON COMENTARIOS

```
/* EJERCICIO 3B DE LA PRÁCTICA 4
```

Dos números a y b se dice que son amigos si la suma de los divisores de a (salvo él mismo)

coincide con b y viceversa (la suma de los divisores de b coincide con a). Por ejemplo, los

números 220 y 284 son amigos.

B. Diseña un programa que tenga como entrada de teclado dos números naturales (n y m) y que muestre en la pantalla todas las parejas de números amigos que existan en el intervalo determinado por n y m . El programa debe asegurarse de que n sea menor que m . Por ejemplo, en el intervalo 1 - 2000 sólo existen dos parejas de amigos: el 220 es amigo del 284, y el 1184 es amigo del 1210.

Autor: Jesús Moncada Ramírez

Fecha: 13/11/2019

```
*/
```

```
#include <iostream>
```

```
using namespace std;
```

```
void leerNumeros(int&, int&);
```

```
int sumaDivisores(int);
```

```
bool sonAmigos(int, int);
```

```
int main()
```

```
{
```

```
    int n, m;
```

```
    leerNumeros(n, m);
```

```
    for (int x = n; x<=m; x++) //Desde n hasta m
```

```
    {
```

```
        int y = sumaDivisores(x);
```

```
        if (y>x) //Si y es mayor que x
```

```
        {
```

```
            if(x==sumaDivisores(y)) //Si la suma de divisores de y es igual a x --> si "x" e "y" son amigos
```

```

        {
            cout << "Numeros amigos encontrados: " << x << ", " << y << endl;
        }
    }

    return 0;
}

/*
    Lee el inicio ("n") y el final("m") del intervalo introducido por teclado.
*/
void leerNumeros(int& n, int& m)
{
    do{
        cout << "Introduce incio del intervalo, n= ";
        cin >> n;
        cout << "Introduce final del intervalo, (> n) m= ";
        cin >> m;

    } while(m<=n); //Controlamos que n sea mayor que m
}

/*
    Devuelve la suma de todos los divisores de num exceptuando a él mismo.
*/
int sumaDivisores(int num)
{
    int suma = 0;
    for(int x = 1; x<num; x++) //Por cada número inferior de num
    {
        if(num%x == 0) //Si x es divisor de num
        {
            suma += x;
        }
    }
    return suma;
}

```