

Hallar parejas de numeros amigos

Escrito por adrianvaca
Martes, 22 de Marzo de 2011 23:59 -

Los numeros amigos son aquellos cuya suma de divisores nos da el otro numero.

Por ejemplo los numeros: 220 y 284

Los divisores de 220 son: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110

Y la suma: $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$

Los divisores de 284 son: 1, 2, 4, 71 y 142.

Y la suma: $1 + 2 + 4 + 71 + 142 = 220$

Bien, ahora contruiremos un programa que nos ayude a encontrar parejas de numeros amigos y escriba los resultados en un archivo de texto...

Las cabeceras serian:

```
// #include <stdlib.h>
#include <iostream.h>
#include <fstream.h>
#include <math.h>
```

Ahora una funcion que nos devuelve la suma de los divisores de un numero cualquiera:

```
// int suma(int n)
{
    int j = (int)sqrt((double) n);
    int suma = 1;
    if(n < j*j) j--;
    for(int i=2; i < j; i++)
    {
        if(n%i==0)
            suma = suma + i + n/i;
    }
    if(n & j == 0)
        suma = suma + j;
    return suma;
}
```

Hallar parejas de numeros amigos

Escrito por adrianvaca

Martes, 22 de Marzo de 2011 23:59 -

```
}
```

Y ahora la funcion principal main la cual hace el trabajo:

```
int main          ()
{
    ofstream myout ( "Salida.txt" );
    int min        = 1      ;
    int max        = 5      ;
    int a          = 0      ;
    int b          = 0      ;
    int k          = 1      ;

    for( a = min ;
    {
        b = suma (
            if( b > a &&
            {
                cout << k <<
                myout << k <<
                k ++;
            }
        }
    }
    myout . close ();
    return 0 ;
}
```

En este ejemplo se calcula las 5 primeras parejas de números amigos a partir del 1. Se puede cambiar los valores de max y min para alterar el comportamiento.

Por ultimo los resultados son escritos en un archivo llamado Salida.txt cuyo contenido es similar al siguiente:

Hallar parejas de numeros amigos

Escrito por adrianvaca
Martes, 22 de Marzo de 2011 23:59 -

1.	-	a	:	220
2.	-	a	:	1184
3.	-	a	:	2620
4.	-	a	:	5020
5.	-	a	:	6232