



PRACTICA 6: NUMEROS PERFECTOS

ESTUDIANTE: EFRAIN ROBLES PULIDO

CODIGO: 221350095

**NOMBRE DE LA MATERIA: SEMINARIO DE SOLUCION DE PROBLEMAS DE
PROGRAMACION**

SECCIÓN: D67

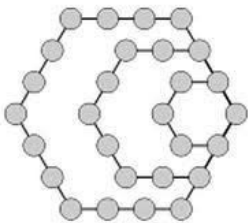
Descripción:

Objetivo de la practica

Hacer un programa para encontrar los números perfectos que hay hasta el número 10,000.

Fundamentación teórica

Un número perfecto es aquel número natural que es igual a la suma de sus divisores propios positivos. Por ejemplo: 6 se puede dividir por 1, 2 y 3, y cuando sumas esos números, el resultado es 6.



Los números perfectos son números hexagonales; los más grandes siempre se pueden disponer en torno a un collar hexagonal. Siendo la forma más eficiente de cubrir el plano sin dejar huecos, rellenando el plano de la forma más eficaz significa que a igualdad de área con otro "rellenando" el perímetro total será menor.

La historia de los números perfectos forma parte de una de las más antiguas y fascinantes ramas de las matemáticas: la teoría de los números.

El primero en referirse a ellos fue nada menos que Euclides, en su influyente obra "Los elementos", publicada en el año 300 a.C.

Había descubierto cuatro números perfectos, y en su libro revelaba una forma segura de hallar otros. Segura, aunque difícil y laboriosa.

						$6 = 1 \times 6$
						$6 = 2 \times 3$
						$6 = 3 \times 2$
						$6 = 6 \times 1$

						$1 + 2 + 3$
						$= 6$

Análisis del problema

Se desea determinar los números perfectos del 2 al 10 000, en donde no se introducirá ningún valor, y se utilizará los ciclos "for" para hacer recorrer los números en determinados rangos, y también se deberá de hacer un arreglo para analizar los números que estarán en los ciclos para que puedan hacer la sumatoria de los divisores que componen el numero a analizar, si dicha suma es lo mismo que el número que se está analizando significa que es un numero perfecto.

Datos de entrada y precondiciones

Los números del 2 al 10,000 los cuales se autogeneran del propio programa.

Datos o elementos de salida

Los números perfectos.

Desarrollo:

Procedimiento en lenguaje natural

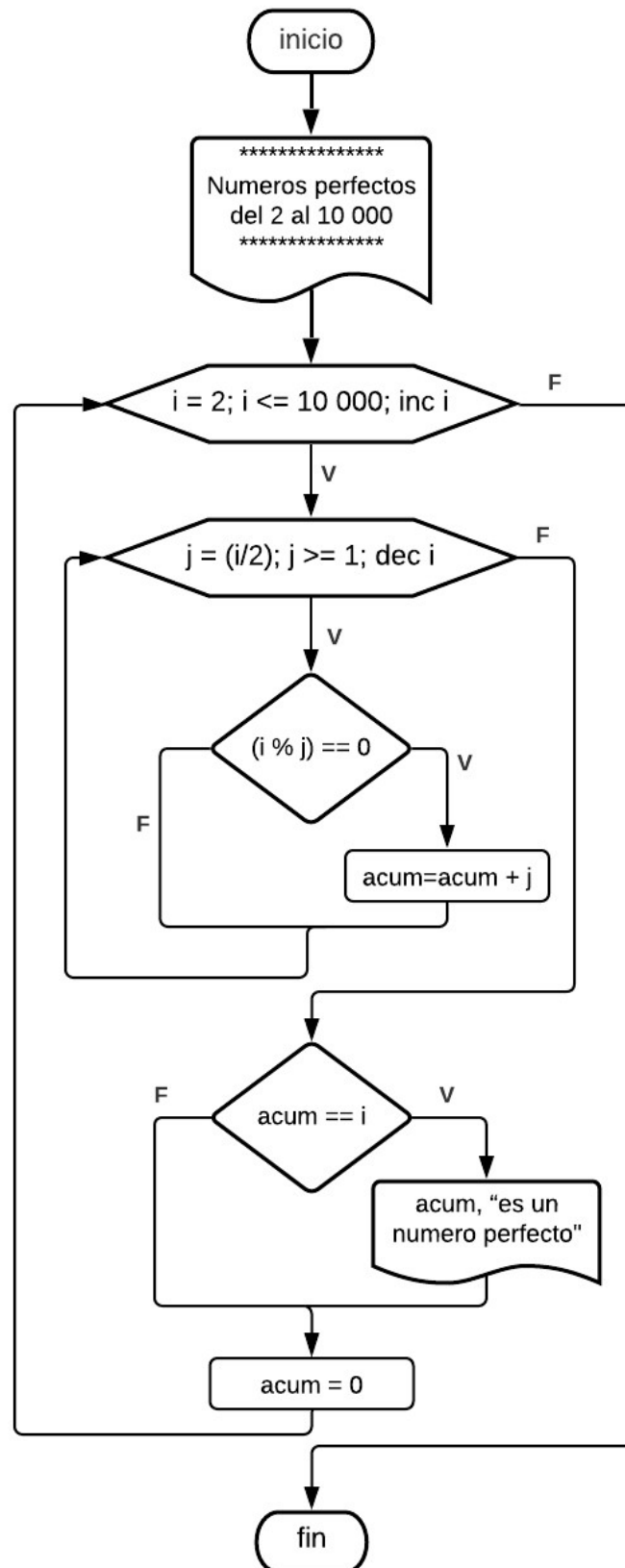
Para determinar los números perfectos del 2 al 10 000 utilizaremos un ciclo “for” para hacer recorrer los números del rango dicho, y dentro de ese ciclo se le añadirá otro ciclo para recorrer los números desde la mitad del número anterior hasta el cero, y dentro de ese ciclo se comprueba si el residuo de la división de los números que componen los ciclos es igual a cero. Si se cumple se acumulará como una sumatoria de los números divisores. Después se comprueba si la sumatoria de los divisores es igual al número que se analiza, si se cumple se mostrará que número es perfecto. El programa se finalizará con la limpieza del acumulador para volver a recorrer o analizar el resto de los números del rango hasta el 10 000.

Algoritmo

Pseudocódigo:

//Efrain Robles Pulido	inicio
//Practica 6: Numeros perfectos	acum=acum + j
Principal	fin
inicio	fin
entero acum, j, i	if (acum==i)
imprimir ("***** Numeros perfectos del 2 al 10 000 *****")	//comprobacion de la operación
for (i ← 2; i ≤ 10000; inc i)	inicio
//ciclo contador de 2 al 10 000	imprimir (acum, “es un numero perfecto”)
inicio	fin
for (j ← (i/2); j ≥ 1; j--)	acum ← 0
//ciclo contador para las sumas parciales	fin
inicio	fin
if ((i%j) == 0) //seleccion de numeros que son divisores	

Diagrama de flujo



Código fuente del programa en lenguaje C (textualmente)

```
//Efrain Robles Pulido
#include <stdio.h>
#include <stdlib.h>

int main(){//Practica 6: Numeros perfectos

    int acum, j, i;

    printf("***** Numeros perfectos del 2 al 10 000 *****\n\n");

    for(i=2;i<=10000;i++){//ciclo contador de 2 al 10 000

        for(j=(i/2);j>=1;j--){//ciclo contador para las sumas parciales

            if((i%j)==0){//seleccion de numeros que son divisores

                acum=acum+j;

            }

        }

        if (acum==i){//comprobacion de la operacion

            printf("\t%d es un numero perfecto\n",acum);

        }

        acum=0;

    }

}
```

Resultados obtenidos:

```
//Efrain Robles Pulido
#include <stdio.h>
#include <stdlib.h>

int main(){//Practica 6: Numeros perfectos
    int acum, j, i;
    printf("***** Numeros perfectos del 2 al 10 000 *****\n\n");
    for(i=2;i<=10000;i++){//ciclo contador de 2 al 10 000
        for(j=(i/2);j>=1;j--){//ciclo contador para las sumas parciales
            if((i%j)==0){//seleccion de numeros pares
                acum=acum+j;
            }
        }
        if (acum==i){//comprobacion de la operacion
            printf("\t%d es un numero perfecto\n",acum);
        }
        acum=0;
    }
}
```

```
***** Numeros perfectos del 2 al 10 000 *****  
  
6 es un numero perfecto  
28 es un numero perfecto  
496 es un numero perfecto  
8128 es un numero perfecto  
  
Process returned 0 (0x0)   execution time : 0.451 s  
Press any key to continue.
```

Conclusión

En esta practica pude aplicar los ciclos anidados para poder obtener los números perfectos mientras se va recorriendo los números enteros del 2 al 10 000. Aprendí como puedo obtener los números perfectos, en donde es una forma más eficiente de repartir o acomodar algo en un espacio como son los hexágonos. También aplique la estructura de control selectiva para comprobar o seleccionar los números que se componían cada ciclo, para poder obtener el resultado deseado. También aprendí que podemos hacer arreglos con los ciclos para ir generando números continuamente sin la necesidad de que el usuario deba introducir dichos numero manualmente.