

Лабораторная работа №4  
по дисциплине «Информатика»

Арифметические задачи

Группа: **АВТ-010**

Студенты: **Антонов Н.А., Довиденко Р.Е.**

Преподаватель: **Романов Е.Л.**

НОВОСИБИРСК 2010

## Задание

Число называется совершенным, если оно равно сумме всех своих делителей, например,  $6=1+2+3$ ,  $28=1+2+4+7+14$ . Найти все совершенные числа в заданном интервале.

## Теоретические сведения

Совершенным называется число, равное сумме всех своих делителей (включая 1, но исключая само число). Наименьшее из совершенных чисел 6 равно сумме трех своих делителей 1, 2 и 3. Следующее совершенное число  $28=1+2+4+7+14$ . Ранние комментаторы Ветхого завета, пишет в своей книге «Математические новеллы» Мартин Гарднер, усматривали в совершенстве чисел 6 и 28 особый смысл. Разве не за 6 дней был сотворен мир, восклицали они, и разве Луна обновляется не за 28 суток? Первым крупным достижением теории совершенных чисел была теорема Евклида о том, что число  $2^n - 1 * (2^n)$  - четное и совершенное, если число  $2^n - 1$  - простое. Лишь две тысячи лет спустя Эйлер доказал, что формула Евклида содержит все четные совершенные числа. Поскольку не известно ни одного нечетного совершенного числа (у читателей есть шанс найти его и прославить свое имя), то обычно, говоря о совершенных числах, имеют в виду четное совершенное число.

## Проектирование программы

### Обсуждение основных идей алгоритма

Идея: Для нахождения данных чисел можно использовать перебор всех чисел, что займёт большое количество времени, но, зная формулу  $2^k * (2^{k+1} - 1)$ , можно сократить время их нахождения. Во втором случае при изменении  $k$  получается множество чисел, которые проверяются на «совершенство»

### «Составные части» программы

- Инициализация таймера  
`long t=clock();`
- Цикл перебора чисел для нахождения совершенного.  
`for (s=0,i=1;i<j;i++)  
    {if (j%i==0) s+=i;}`
- Формула вычисления числа для отбора на совершенное.  
`v=1<<k;  
j=v*(2*v-1);`
- Условие определения совершенного числа (сохраняется в массиве).  
`if (s==j) C[p++]=j`
- Вывод совершенного числа, степени двойки (на данный момент работы программы) и времени работы (в миллисекундах)  
`cout << j << endl;  
cout << "k=" << k << " t=" << clock()-t << endl;`

Переменные:

- `i` - счётчик делителей
- `v` -  $2^k$
- `C[100]` - массив совершенных чисел
- `k` - счётчик степеней двойки
- `j` - счётчик делимых

p – счётчик совершенных чисел  
s – сумма делителей

### **Текст программы с комментариями**

```
using namespace std;
void main(){
    _int64 i,v,C[100],k,j,p,s;
    v=2000000000;
    k=0;
    p=0;
    for (k=1; k<30; k++){ // запуск основного цикла программы, увеличивающего степень двойки
        long t=clock(); // запоминание текущего времени для таймера
        v=1<<k; // сдвиг степени двойки на единицу (в машинном коде)
        j=v*(2*v-1); // применяем формулу
        for (s=0,i=1;i<j;i++) // запуск цикла для подсчёта суммы делителей
        {
            if (j%i==0) s+=i; // Если счётчик является делителем числа, то он складывается с
            // другими делителями
            .....
        }
        for (i=0;i<p;i++) // цикл для вывода массива
            cout << C[i] << endl; // вывод массива
    }
}
```

### **Пример работы программы**

```
6
k=1 t=0
28
k=2 t=0
k=3 t=0
496
k=4 t=0
k=5 t=0
8128
k=6 t=0
k=7 t=0
k=8 t=0
k=9 t=0
k=10 t=31
k=11 t=78
33550336
k=12 t=344
k=13 t=1312
k=14 t=5219
k=15 t=20845
```

### **Ошибки и неточности**

#### **Выводы**

Способ перебора всех чисел не дает эффективного результата (времени тратится значительно больше нежели в отборе по формуле). Для решения данной задачи предпочтительнее использование формулы. Но даже при использовании формулы на 16, 17 степенях двойки подсчёты занимают уже десятки секунд.