

Лабораторная работа №1

Шикунов Максим Евгеньевич Р3333
Проблемы номер 5 и 26

Описание проблем

Проблема 5

Наименьшее кратное 2520 - самое маленькое число, которое делится без остатка на все числа от 1 до 10.
Какое самое маленькое число делится нацело на все числа от 1 до 20?

Проблема 26

Обратные циклы Единичная дробь имеет 1 в числителе. Десятичные представления единичных дробей со знаменателями от 2 до 10 даны ниже:

1/2 = 0.5
1/3 = 0.(3)
1/4 = 0.25
1/5 = 0.2
1/6 = 0.1(6)
1/7 = 0.(142857)
1/8 = 0.125
1/9 = 0.(1)
1/10 = 0.1

Где 0.1(6) значит 0.166666..., и имеет повторяющуюся последовательность из одной цифры. Заметим, что 1/7 имеет повторяющуюся последовательность из 6 цифр.

Найдите значение $d < 1000$, для которого $1/d$ в десятичном виде содержит самую длинную повторяющуюся последовательность цифр.

Решение проблем

Решение проблемы 5 представлено в файле [SmallestMultiple.hs](#)
Для данной проблемы были написаны две основные вспомогательные функции: **findGcd** и **findLcm** (Нахождение НОД и НОК соответственно)
Они нам нужны для того, чтобы решить проблему. Идея заключается в том, чтобы найти наименьшее общее кратное для всех чисел с 1 по 20.
Дальше, используя данные вспомогательные функции, проблема была решена 4 способами:

- Рекурсией (функция **recursionMethod**)
- Хвостовой рекурсией (функция **tailRecursionMethod**)
- Модульным со сверсткой (функция **moduleMethod**)
- Используя бесконечный массив (функция **infinityListMethod**)

Решение данной проблемы на c++:

```
#include
using namespace std;

long gcd(long a, long b) {
    while (b != 0) {
        long temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}
```

```

}

long lcm(long a, long b) {
    return (a * b) / gcd(a, b);
}

int main() {
    long ans = 1;
    for (long i = 2; i <= 20; i++) {
        ans = lcm(ans, i);
    }
    cout << ans;
    return 0;
}

```

Решение проблемы 26 представлено в файле [ReciprocalCycles.hs](#)

В данной решение была написана одна вспомогательная функция: **findCycle**. Которая вычисляет повторяющуюся последовательность после запятой.

Мы проходимся по всем числам от 1 до 1000, смотрим у каждого какая длина данной последовательности и выводим то число, у которого она наибольшая.

Решено было также 4 способами:

- Рекурсией (функция **recursionMethod**)
- Хвостовой рекурсией (функция **tailRecursionMethod**)
- Модульным со сверсткой и замыканием (функция **moduleMethod**)
- Используя кастомный map (функция **mapMethod**)

Решение на языке c++:

```

#include
#include
using namespace std;

long findCycle(int divider) {
    int remainder = 1;
    int number = 1;
    int len = 0;
    unordered_map myMap;
    while (remainder != 0) {
        remainder = number % divider;
        number = remainder * 10;
        if (myMap.find(remainder) != myMap.end()) {
            return len - myMap[remainder];
        }
        myMap[remainder] = len;
        len++;
    }
    return 0;
}

int main() {
    int maxLen = 0, maxNumber = 0, newLen;
    for (int i = 1000; i > 0; i--) {
        newLen = findCycle(i);
        if (newLen > maxLen) {
            maxLen = newLen;
            maxNumber = i;
        }
        if (maxLen >= i) {
            break;
        }
    }
    cout << maxNumber;
    return 0;
}

```