

## Лабораторная 9

Сдать до 09.11

### Задание 1. 100 000 000 строк (5 баллов)

Создайте файл (\*.txt), содержащий 100 000 000 строк с целыми числами в диапазоне от 0 до 99999999. Формат чисел – всегда 8 цифр. Например: 1 = 00000001, 65535 = 00065535

Т.е. каждая строка имеет равную длину в 8 символов.

Все числа в файле идут в случайном порядке без повторов (каждая строка - уникальное число):

00306453

99645283

70000021

06847127

Число 100 000 000 для количества строк в файле дано для примера. Это число может быть сколь угодно большим – настолько большим, что загрузить в память массив из целых чисел такой длины окажется невозможным.

Предложите решение, в котором **все данные для файла не будут загружены в память одновременно**.

### Задание 2. Сериализация/десериализация студенческих групп (3 балла)

Сериализация — это процесс преобразования структуры данных или объекта в последовательность битов, чтобы их можно было хранить в буфере файла или памяти или передавать по каналу сетевого подключения для последующего восстановления в той же или другой компьютерной среде.

Пусть у вас есть два класса: студент (**Student**) и студенческая группа (**Group**), описанные ниже:

```
class Group
{
    public decimal GroupId { get; set; }

    public string Name { get; set; }

    public List<Student> Students { get; set; }

    // no need to serialize this
    public int StudentsCount { get; set; }
}

class Student
{
    public decimal StudentId { get; set; }

    public string FirstName { get; set; }

    public string LastName { get; set; }

    public int Age { get; set; }

    public Group Group { get; set; }
}
```

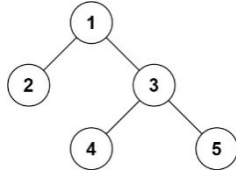
Вам необходимо сериализовать студенческую группу в файл и десериализовать ее обратно, используя бинарный сериализатор.

### Задание 3. Сериализация/десериализация двоичного дерева (4 балла)

Реализуйте свой алгоритм сериализации и десериализации для двоичного дерева. Нет никаких ограничений на то, как должен работать алгоритм. Вам просто нужно убедиться, что двоичное дерево может быть сериализовано, например, в строку, и эта строка может быть десериализована обратно в исходное дерево.

Предложите свои методы для сериализации и десериализации **без использования стандартных методов**.

Пример:



«2,1,4,3,5»

### Задание 4. Поиск файла (3 балла)

Реализуйте метод для рекурсивного поиска файла на диске по его имени. Метод получает имя файла (опционально можно задать расширение для файла) и возвращает полный путь до первого файла, имя которого совпадает с заданным. Предложите свое решение.

Например

`findFile("MyFile.csv") => "C:\Temp\MyStorage\MyFile.csv"`

### Задание 5. Рациональное число (4 балла)

Создайте функцию `Rational()`, которая получает два натуральных числа ( $a$ ,  $b$ ) и возвращает строку, содержащую запись рационального числа  $a/b$  в виде десятичной дроби, возможно, **периодической**. Всегда верно неравенство:  $a < b$ . Длина периодической дроби может быть больше 20 цифр.

Примеры

`Rational(2, 5) => "0.4"`

`Rational(1, 6) => "0.1(6)"`

`Rational(1, 3) => "0.(3)"`

`Rational(1, 7) => "0.(142857)"`

`Rational(1, 77) => "0.(012987)"`