Городская научно-практическая конференция школьников

«Содружество»

Секция: ИНФОРМАТИКА

Номинация: реферативно-исследовательская работа

**Изучение метода длинная арифметика**

Автор: Погула Андрей,

10 класс, МБОУ ФМЛ №174,  
г. Зеленогорск, Красноярский край

Руководители: Снегирева В.С.,

УчительМБОУ «Лицей №174»,  
г. Зеленогорск, Красноярский край

г. Зеленогорск, 2014/2015 учебный год

**Оглавление**

[Введение 3](#_Toc251193725)

1. Теоретическая часть……………………………………………………………………………..4

[1.1. Основные понятия 4](#_Toc251193727)

[1.2. Стандартные типы данных. 5](#_Toc251193727)

[1.3. Смысл и идеи реализации Длинной арифметики 5](#_Toc251193728)

2. Решение задач………..…………………………………………………………………………..7

[2.1. Задача №1. 7](#_Toc251193727)

[2.2. Задача №2. 8](#_Toc251193727)

[2.3. Задача №3.](#_Toc251193727) 10

[2.4. Задача №4.](#_Toc251193727) 11

3. Заключение......……..…………………………………………………………………………..12

4.Приложение..………..…………………………………………………………………………..13

5.Список литературы....…………………………………………………………………………..18

**Введение**

Данная исследовательская работа посвящена «Длинной арифметике и её реализации на языке С++».

**Проблема**: существует класс задач, которые нельзя решить с помощью стандартных типов данных.

**Гипотеза**: предполагаем, что есть специальный метод решения данных задач.

**Актуальность** и значимость этой темы для меня, определяется тем, что в некоторых олимпиадных, астрономических и жизненных задачах приходится работать с данными, а именно числами, которые не помещаются не в один тип языка С++. В этих случаях применяется «Длинная арифметика».

**Цель**: изучение метода длинная арифметика.

**Задачи**:

1. Изучить метод длинная арифметика.

2. Найти класс задач решаемых данным методом.

3. Научиться решать задачи данного типа.

Объектом исследования является метод длинная арифметика, а предметом задачи решаемые данным методом.

Методы, которые помогали в решении поставленной задачи:

1. Теоретический
2. Эмпирический
3. Эксперимент

**Глава 1**

**Основные понятия**

Для достижения поставленных целей нам придётся оперировать некоторыми понятиями, и выяснить их значение.

Длинная арифметика - раздел олимпиадного программирования, в котором рассматривается реализация действий с большими числами, не умещающихся в стандартных типах данных. На сегодняшний день одним из языков, используемым для решения олимпиадных задач и поддерживающим длинную арифметику, является Java, где все необходимые функции для работы с длинными числами встроены и можно обойтись без трудоемких реализаций.

Тип данных- Допустимое множество значений.

Строка-(String)-Массив символов, где каждый отдельно взятый символ это переменная символьного типа Char.

Символьный тип данных-(Char)- Простой тип данных, предназначенный для хранения одного символа в определённой кодировке.

**Стандартные типы данных**

Предлагаю рассмотреть основные целочисленные типы данных языка С++и диапазон их значений:

|  |  |
| --- | --- |
| Short | от -32 768 до 32 767 |
| UnsignedShort | От 0 до 65 535 |
| Int | от -2 147 483 648 до 2 147 483 647 |
| UnsignedInt | От 0 до 4 294 967 295 |
| Long | от -2 147 483 648 до 2 147 483 647 |
| UnsignedLong | От 0 до 4 294 967 295 |
| LongLong | От -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807 |
| UnsignedLongLong | От 0 до 18 446 744 073 709 551 615 |

Из этой таблицы следует, что наибольшее число которым мы можем оперировать это 18 446 744 073 709 551 615 (18 квадриллионов 446 триллиардов 744 триллиона 73 миллиарда 709 миллионов 551 тысяча 615) или просто 2 в 64 степени -1. Но число 2 в 64 уже не помещается не в один из типов. Для расчёта больших чисел нам потребуется другой метод.

**Смысл и идеи реализации Длинной арифметики**

Из вышеизложенного мы понимаем, что нужно увеличить программно тип данных или создать свой. Второй вариант намного проще. И эту реализацию можно упростить используя строку(String).Мы можем хранить в виде строки наше число, где каждый i-тый символ строки соответствует i-той цифре исходного числа, а это значит, что можно свободно выполнять операции посимвольно, такие как сложение, умножение и сравнение больших чисел.

Используя строку вместо какого-либо целочисленного типа, мы практически снимаем ограничения на размер числа, единственным ограничением для нас будет являться, объём оперативной памяти.

Разобравшись с идеей, предлагаю перейти к возможностям при данной реализации.

В теории можно реализовывать алгоритмы, например:

* Сравнение

Данный алгоритм является самым простым за счёт строчной реализации. В языке С++ разрешено сравнивать строки как числа, но ответ будет верным тольковслучае равной длинны сравниваемых чисел. Отсюда следует, что при написании программы придётся добавить в начало меньшего числа лидирующих нулей, так чтобы размеры числе были равными.

* Сложение
* Умножение длинного числа на короткое
* Вычитание

Реализация сводится к посимвольной работе с числами и переносом в следующий разряд, как при решении в столбик.

* Целочисленное деление

Данный алгоритм несколько сложнее, так как мы вынуждены будем не просто работать с числом посимвольно, а ещё и проверять можем ли мы выполнить эту операцию.

* Нецелочисленное деление

Сложность заключается в вычислении дробной части, так как она может быть бесконечна (периодична) и в определении момента когда начинается дробная часть.

* Умножение длинного числа на длинное

В рамках обычных вычислений это операция не сложна, но в олимпиадном программировании, где на задачу отводится определённое количество времени, очень важна скорость работы, простота написания и выполнения алгоритма. Поэтому простое решение, как с перемножением длинного на короткое нам не подходит. Асимптотика данного решения будет O(A\*B), где A- длина первого числа, а В - длина второго.

Решение есть и для этой задачи, но оно требует другого подхода. Потребуется производить не посимвольное умножение, а разбивать большое число на меньшие, такие чтобы они вмещались в стандартные типы данных и производить умножение маленьких чисел.

Асимптотика данного решения будет O(A\*B)/C, где A- длина первого числа, В - длина второго, а C - длина меньшего числа, на которое мы разбивали.

В данном случае очевидно, что чем больше С , тем быстрее будет работать программа.

* Поиск факториала данного числа

Данный алгоритм в теории прост, но мы сталкиваемся с той же проблемой, что и в предыдущей реализации. На этот раз проблема будет заключатся не в длине числа, а в том что нам придётся произвести как минимум Nумножений для каждой их цифр.

Отсюда асимптотика будет равной O(N\*A), где A- длина числа и N данное нам число.

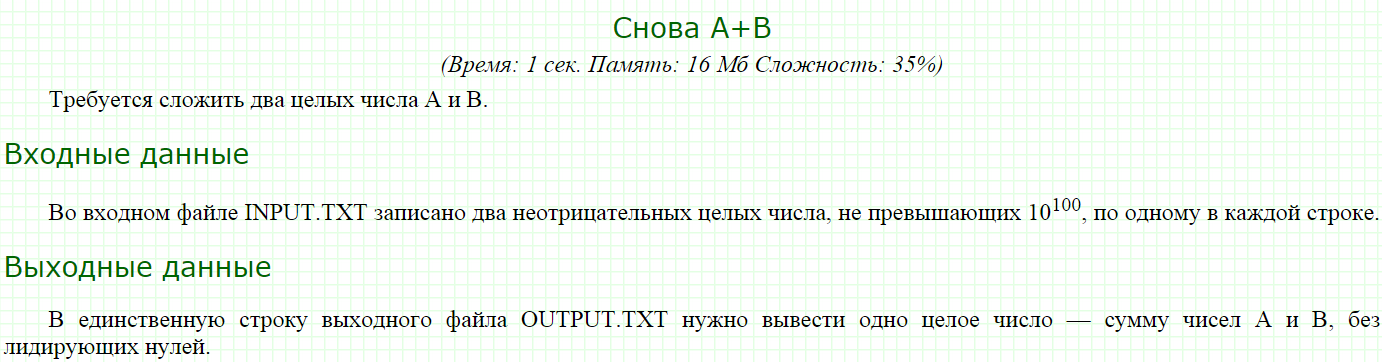
* Возведение числа в степень

Для начала предлагаю рассмотреть асимптотику при строчной реализации. О(А^2\*В), где A- длина числа, а В степень, в которую требуется возвести данное число.

Отсюда следует что при увеличении длины данного числа на 1, количество операций возрастает в раз. Что при больших значения А будет сильно влиять на скорость работы.

**Глава 2**

**Задача №1**



В задаче от нас требуется найти сумму двух чисел А и В,но ограничения до 10 в 100 степени говорят нам, что не в один целочисленный тип данных языка С++ не поместятся наши числа.

Тут нам и придётся использовать Длинную арифметику.

Суть данного алгоритма состоит в превращении числа в строку и работу с не с целым(большим числом), а с каждым символом отдельно,как это делается при сложении в столбик.

1. Сначала мы добавляем в наши числа незначащие нули,так чтобы их длина

была равна. Это сделано для упрощения алгоритма и исключения выхода за границы строки.

2) Далее, мы создаём целочисленную переменную ff в которой будем хранить число переходящие в следующий разряд.

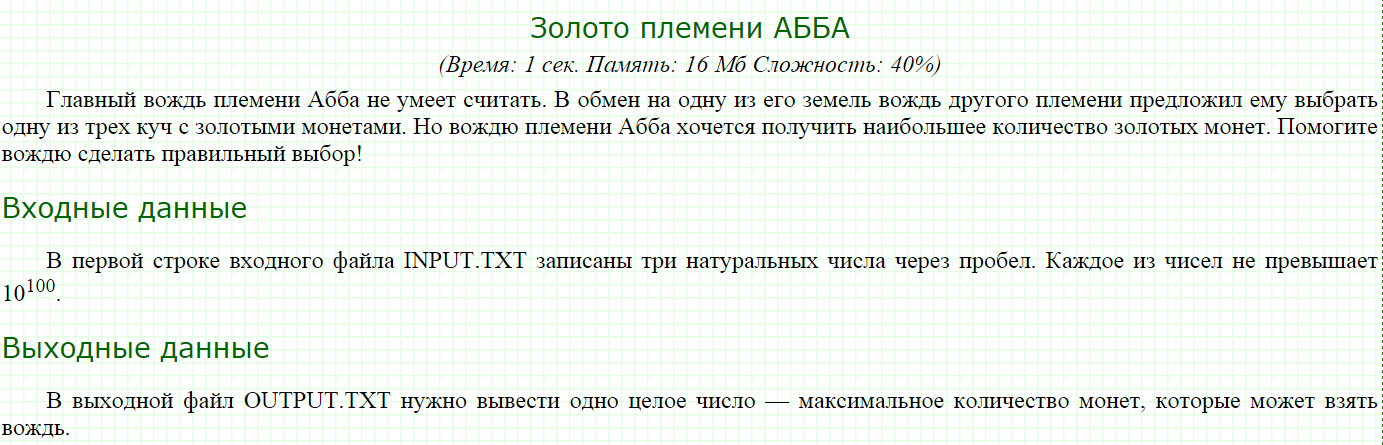
3) Затем, запускаем цикл с конца числа, как это делается при сложении в столбик. В переменную P кладём сумму двух цифр стоящих на i-той позиции в числе А и В. В строку strмы добавляем остаток от деления на 10 числа P.

4) После этого мы проверяем, нужно ли нам переносить в следующий разряд единичку или нет и заходим на следующую итерацию цикла.

В конечном итоге мы избавляемся от незначащих нулей и функция возвращает сумму 2 чисел в виде строки.

Реализация задачи А+В на языке С++ находится в приложении

**Задача №2**



Суть задачи состоит в том, чтобы среди 3-ёх чисел найти максимальное. Входные числа

вновь намного больше стандартных целочисленных типов языка С++, а значит что мы опять должны использовать длинную арифметику.

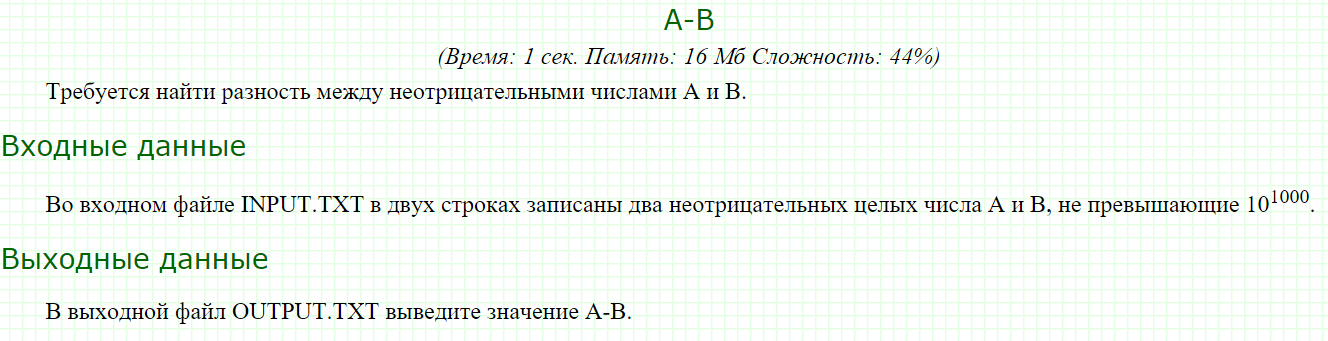
Несмотря на большую заявленную сложность, на данном языке задача решается проще предыдущей.

Алгоритм решения:

1. Считываем три числа в строки добавляем в начало каждого ведущих нулей, так чтобыдлинна каждого из чисел была одинаковой ,в данном случае это 200 символов.
2. Далее следует три вложенных условия где мы просто сравниваем строки как обычные числа и выводим наибольшее.

Решение задачи «Золото племени АББА» на языке С++ находится в приложении

**Задача №3**



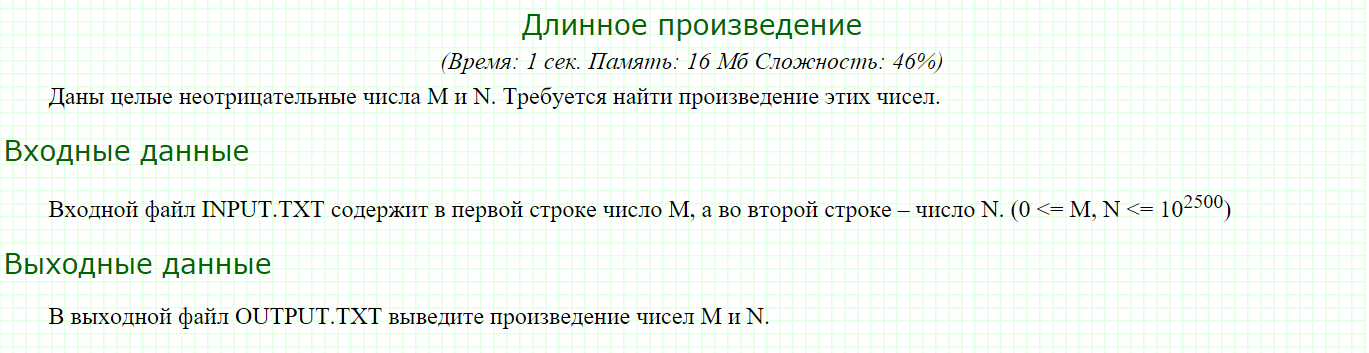
Данная задача чуть сложнее предыдущих в реализации. Это происходит из-за того ,что мы должны занимать из большего разряда. И именно поэтому код задачи становится объёмнее.

Алгоритм:

1. Проверка на частные случаи с нулём и на равенство чисел.
2. Добавляем в оба числа ведущих нулей.
3. Проверяем какое из чисел больше и если нужно, запоминаем наличие минуса в ответе.
4. Запускаем вычитание с конца, но при этом мы поступаем жадно, т.е. не смотрим на впереди идущее число и можем ли мы из него занять.
5. Дальше запускается цикл проверяющий, всё ли в порядке у нас дальше. Так как мы используем строки цифра 0 имеет код 48, и если мы жадно вычтем, то можем получить «**/**» с кодом 47. Это проверяет и исправляет следующий цикл.
6. И в конце программы избавляемся от лидирующих нулей, выводим минус, если это требуется и следом само число.

Решение задачи «А-В» на языке С++ находится в приложении

**Задача №4**

****

Реализовывать данную задачу мы будем как при умножении в столбик.

Алгоритм:

1. Сначала подготовим строки для работы, в С добавим нулей, а строки А и В развернём.
2. Затем производим посимвольное умножение.
3. Разворачиваем строку С и избавляемся от незначащих нулей.

Решение задачи «Длинное произведение» на языке С++ находится в приложении

**Заключение**

На примере рассмотренных в работе задач, мы можем убедиться, что для решения задач на длинную арифметику, строчный метод подходит, и в некоторых случаях проще и короче в реализации.

Поставленная мною цель, изучение метода длинная арифметика, достигнута, задачи:

1. Изучить метод длинная арифметика.

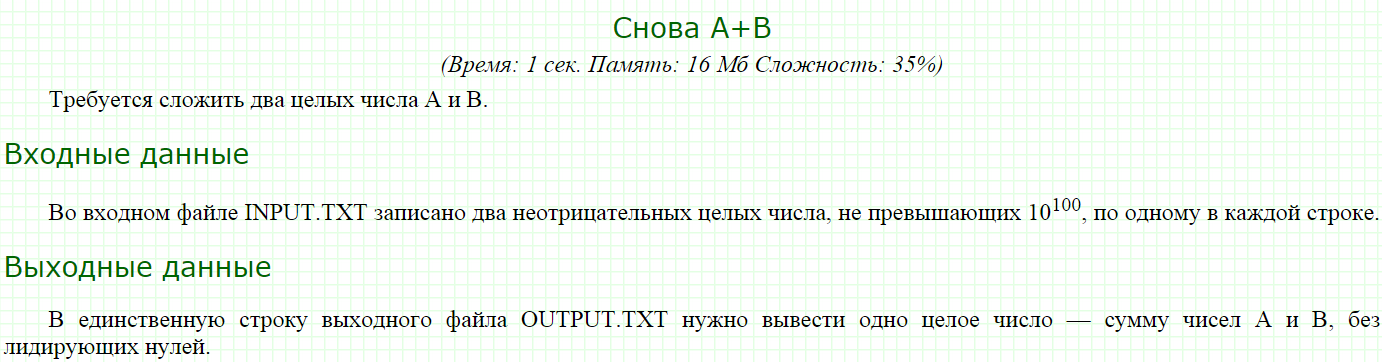
2. Найти класс задач решаемых данным методом.

3. Научиться решать задачи данного типа.

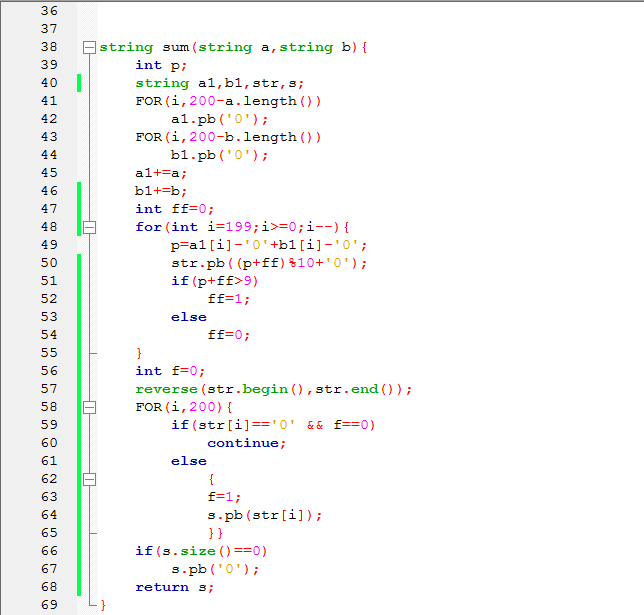
решены. Гипотеза о том, способ для решения таких задач существует и его можно реализовать используя строки, подтвердилась.

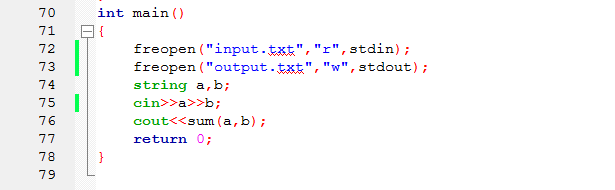
**Приложение**

**Задача «Снова А+В»**

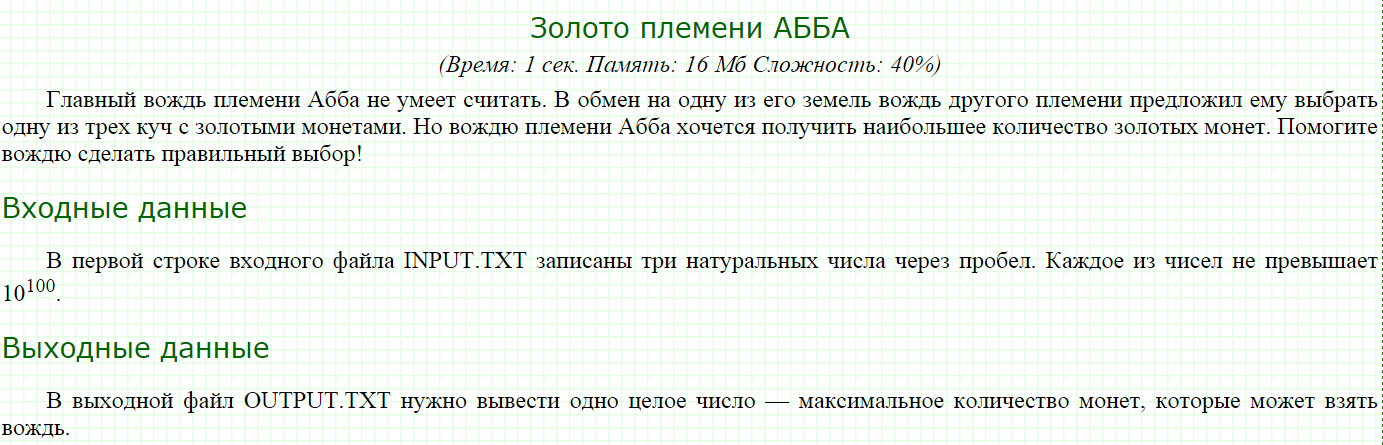
****

**Реализация задачи А+В на языке С++**

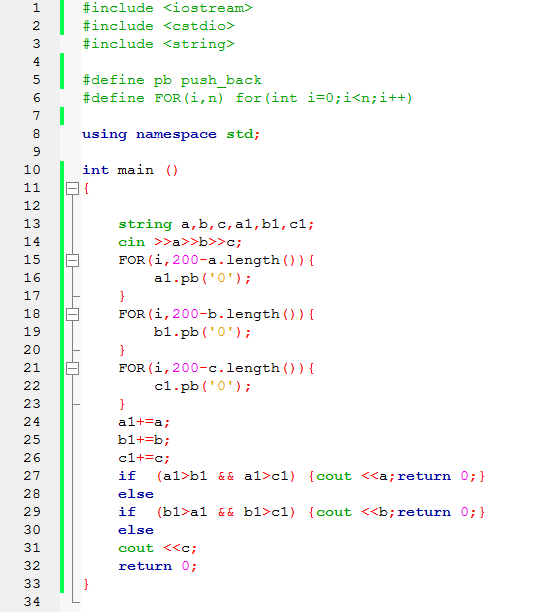
****

****

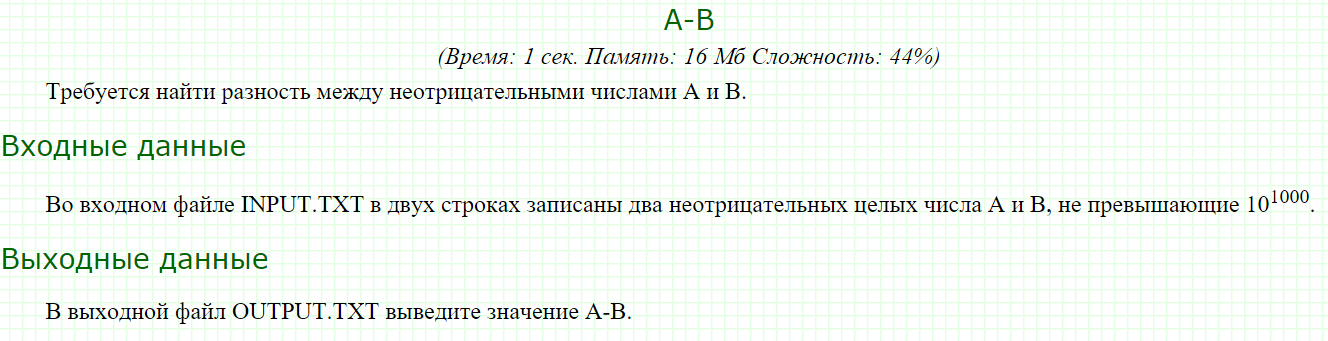
**Задача «Золото племени АББА»**

****

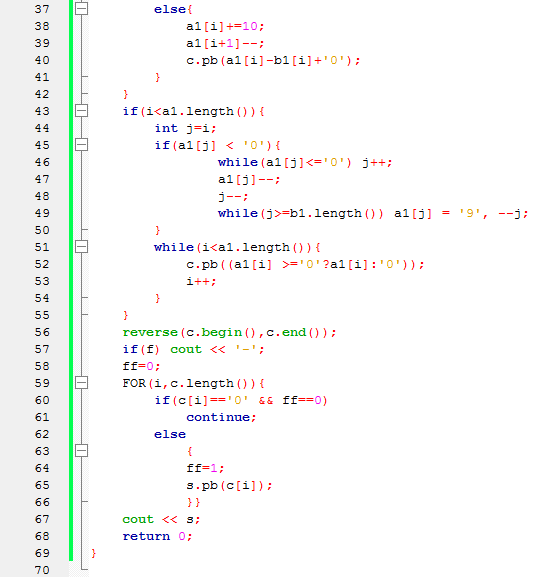
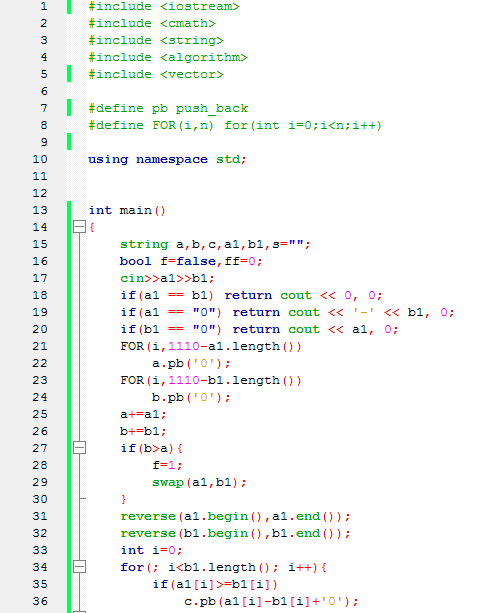
**Решение задачи «Золото племени АББА» на языке С++**

****

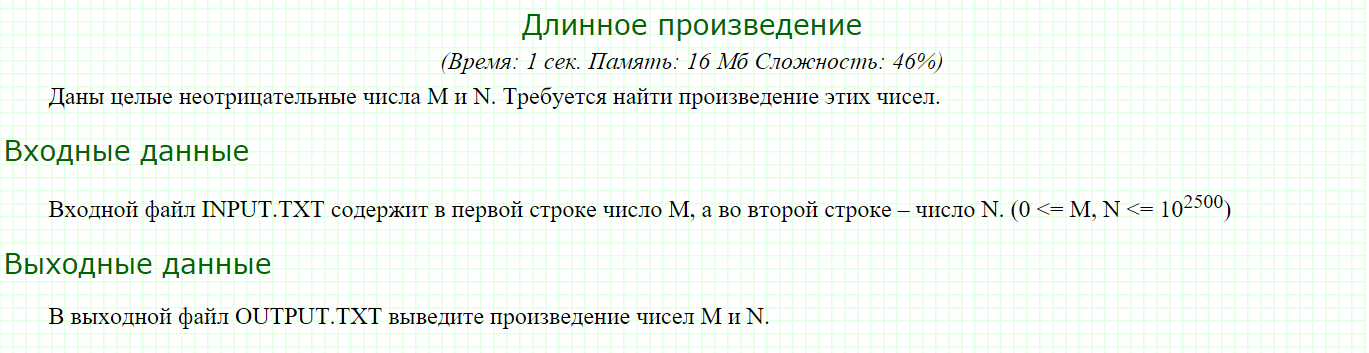
**Задача «А-В»**



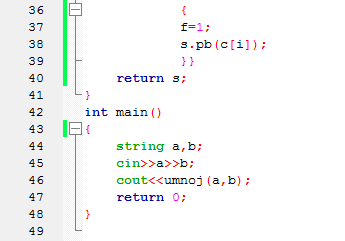
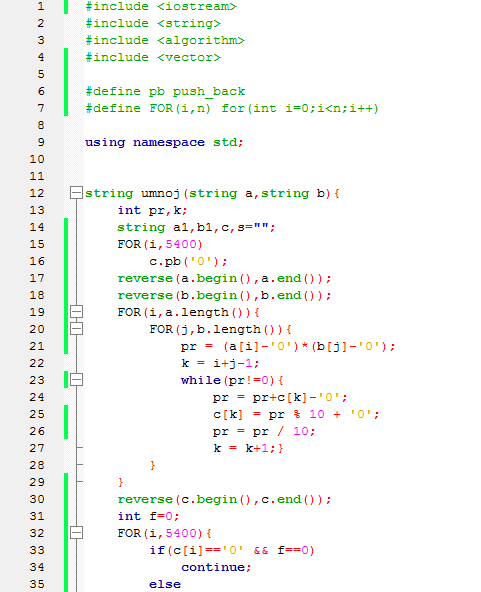
**Решение задачи «А-В» на языке С++**



**Задача «Длинное произведение»**

****

**Решение задачи «Длинное произведение» на языке С++**

****

**Список литературы:**

1. Информатика. Углубленный уровень. Для 11 класса К.Ю. Поляков, Е.А.Еремин
2. Справочник спортивного программиста. М.Д. Кормышов, В.А. Демиденко
3. Решение олимпиадных задач по программированию. С.Н. Беляев
4. <http://acmp.ru/>
5. <http://codeforces.ru/>