Министерство науки и высшего образования РФ

ФГБОУ ВПО «Пермский государственный национальный исследовательский университет»

Кафедра прикладной математики и информатики

**Индивидуальное задание №22**

***по дисциплине «Алгоритмизация и программирование»***

|  |  |
| --- | --- |
|  | **Выполнил:** |
|  | Студент 1-го курса механико-математического ф-та  \_\_\_\_\_\_\_\_\_\_\_ Ипатов К.И.,  группа ФИТ-1,2. |
|  | **Проверил:** |
|  | Старший преподаватель кафедры прикладной математики и информатики ПГУ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кнутова Н.С. |
|  | «11»декабрь\_ 2019 г. |

Пермь 2019

**Постановка задачи**

По заданному натуральному числу A требуется найти наибольшее число B такое, что B2 <= A.

Входные данные:

Во входном файле INPUT.TXT записано натуральное число A (A <= 103000).

Выходные данные:

В выходной файл OUTPUT.TXT выведите максимальное натуральное число B, квадрат которого не превосходит A. Число B следует выводить без лидирующих нулей.

*Пример файла входных данных:*

17

*Пример файла выходных данных:*  
4

**Определение идеи алгоритма, выбор методов решения и структур данных**

Очевидно, что ни один стандартный числовой тип данных нам не подойдет – даже переменная типа unsigned long long int поместит в себя лишь 20-значное число. Поэтому для представления числа A и его корня будем использовать строковую переменную типа string. Операции сложения, вычитания и сложения реализуем в виде отдельных функций.

Чтобы вычислить корень из исходного числа A, воспользуемся методом вычитания нечетных чисел. Алгоритм следующий:

1. Удаляем незначащие нули в начале числа, если таковые имеются.
2. Разбиваем число A на группы из двух цифр справа налево. В том случае, если A состоит из нечетного количества цифр, старший разряд оставляем без пары.
3. Теперь начинаем идти слева направо. Берем первую группу цифр числа A -получившееся "подчисло" обозначим как Sub. Введем также число V, которым будем обозначать вычитаемые из Sub нечетные числа.
4. Из Sub последовательно вычитаем нечетные числа 1, 3, 5, 7, … , (2n – 1) (то есть V на каждом шаге увеличивается на 2) до тех пор, пока не получится отрицательное число. Количество вычитаний, получившееся при этом, будет являться первой цифрой корня числа A.
5. Обнуляем счетчик количества вычитаний. К последней положительной разности приписываем следующую группу цифр числа A. К последнему вычитаемому V прибавляем 1, умножаем его на 10 и снова прибавляем 1. Снова вычитаем V из Sub до тех пор, пока разница больше нуля (не забывая после каждого вычитания увеличивать V на 2).
6. Получаем очередную цифру корня – получившееся количество вычитаний. Повторяем пункты 5 и 6 до тех пор, пока в числе A остаются группы цифр, которые можно приписать к Sub.

В основной программе нам понадобятся следующие переменные:

**S** – переменная типа string, в которой будет храниться число A. Будет считываться из файла "input.txt".

**N** – переменная типа int, куда будет заноситься количество цифр числа A (в том числе после удаления незначащих нулей).

**Sub** – переменная типа string, представляет собой "подчисло" числа A, из которого будут вычитаться нечетные числа, а затем добавляться группы цифр числа A.

**V** – переменная типа string, в которой будут храниться вычитаемые нечетные числа.

**One** – переменная типа string, в которой будет храниться число 1. Нужна она вот для чего: так как после каждого цикла вычитаний V лишний раз увеличивается на 2 (а после каждого цикла нам надо прибавить к нему 1, умножить на 10 и снова прибавить 1) , то из V мы просто вычитаем One, тем самым как бы "добавляя" единицу.

**kor** – переменная типа string, куда будут заноситься цифры корня. По завершению всех вычислений в этой переменной будет храниться итоговый результат – искомый целочисленный корень числа A, который будет заноситься в файл "output.txt".

**k** – переменная типа int, представляет собой счетчик количества вычитаний.

Также понадобятся следующие функции:

**del\_zero()** – функция типа void, удаляет незначащие нули в числе A.

**oborona()** – функция типа string, выполняет защиту "от дурака" и выводит сообщение об ошибке и в чем она заключается, если таковая есть.

**sravnenie()** – функция типа int, поразрядно сравнивает числа Sub и V, возвращая значение 0, если Sub >= V, И -1 – в противном случае.

**minus\_Sub()** – функция типа string, вычитает из Sub нечетное число V и возвращает новое значение Sub. Также эта функция нужна для вычитания из V единицы после каждого цикла вычитаний.

**plus\_Two()** – функция типа string, возвращает значение V, увеличенное на 2.

Немного подробнее о каждой функции:

**del\_zero()** – в эту функцию по ссылке передается переменная S. Для начала проверяется, состоит ли число A более чем из одной цифры и начинается ли оно с нуля. Если да, в таком случае в цикле с помощью функции **erase()** удаляем незначащие нули.

**oborona() –** принимает в качестве аргумента переменную S. Объявляем переменную строкового типа P, в которую будет заноситься сообщение об ошибке. Если S представляет собой пустую строку, выводится сообщение о том, что файл пуст. Если в S содержатся лишние символы (буквы, знак подчеркивания, дефис, др.), выводится сообщение с предупреждением о том, что файл, в котором хранится S, должен содержать только натуральное число. Если S – 3001-значное число, то выполняется проверка на то, чтобы оно не превышало 10^3000. Если превышает – выводится сообщение об ошибке. В случае, если S состоит более чем из 3001 цифры, выводится сообщение о превышении допустимого количества символов.

**sravnenie() –** в качестве аргументов принимает переменные Sub и V и поразрядно их сравнивает. Если Sub >= V, возвращает 0, в противном случае - -1.

**minus\_Sub()** – в качестве аргументов принимает переменные Sub и V. Объявляем переменные **n1** и **n2** типа int, равные длинам Sub и V соответственно. Объявляем переменную строкового типа **RaznS**. Переворачиваем строки Sub и V задом наперед с помощью функции **reverse()**. Если у V меньше разрядов, чем у Sub, дополняем его конец незначащими нулями для корректного вычитания. Вводим переменную **mns** типа int, изначально равную нулю. Далее в цикле от i = 0 до i < n1:

1. Присваиваем переменной mns значение mns + (Sub[i] – '0') – (V[i] – '0') + 10. Вычитание '0' необходимо для того чтобы перевести цифру из символьного типа данных в числовой.
2. С помощью функции **push.back()** добавляем в конец строки RaznS значение mns % 10 + '0'.
3. Если mns <= 9, то присваиваем ему значение -1, в противном случае – 0.

После завершения цикла переворачиваем RaznS с помощью функции reverse() – получаем новое значение Sub.

Также эта функция будет необходима для того, чтобы вычесть из V единицу. В таком случае функция в качестве аргументов примет переменные V и One.

**plus\_Two()** – принимает в качестве аргумента переменную V. Объявляем переменную **n1**, равную длине числа V. Объявляем переменную **SumV** типа string. Переворачиваем V. Объявляем переменную **perenos** типа int, равную нулю, и переменную **sum** типа int, которой присвоим значение (V[0] – '0') + 2 + perenos. В конец SumV с помощью функции push.back() заносим остаток от деления sum на 10, а переменной perenos присваиваем значение sum / 10. Далее в цикле от i = 1 до i < n1:

1. Присваиваем переменной sum значение ((V[i] – '0') + perenos).
2. Остаток от деления sum на 10 заносим в конец SumV.
3. Переменной perenos присваиваем значение sum / 10.

После завершения цикла, если переменная perenos не равна нулю, то записываем ее в конец SumV. Переворачиваем SumV – получаем значение V, увеличенное на 2.

**Основная программа:**

Из файла "input.txt" считываем строку S, в которой хранится число A. Сразу удаляем незначащие нули, если они есть, с помощью функции del\_zero().

Далее включается защита "от дурака". Если функция oborona() возвращает пустую строку, значит, число в файле корректно, поэтому программа продолжает работу. В ином случае программа выводит сообщение об ошибке как на консоль, так и в файл "output.txt".

В переменную N заносится количество цифр числа A. Объявляется переменная Sub. Если в числе A четное количество цифр, то берем в качестве значения Sub два старших разряда, если нечетное – один.

Объявляем переменные V, One, kor, k.

В цикле, пока Sub >= V (это условие проверяем с помощью функции sravnenie()):

1. С помощью функции minus\_Sub() вычитаем из Sub нечетное число V
2. Увеличиваем V на 2 с помощью функции plus\_Two().
3. Увеличиваем значение k на 1.

После завершения цикла в конец строки kor (а она пока пустая) заносим значение k + '0' и обнуляем k.

Далее, так как V на последнем шаге лишний раз увеличилось на 2, а нам надо, чтобы оно увеличилось лишь на единицу, с помощью функции minus\_Sub() вычитаем из него 1. Процедуру умножения на 10 и прибавления 1 реализуем очень просто – просто с помощью функции push.back() добавим 1 в конец V.

Объявляем переменную i и обнуляем ее. Далее, если число A состоит более чем из двух цифр, в цикле от i = 1 или 2 (в зависимости от того, нечетное или четное количество цифр в числе A) до i < N с шагом i+=2 дописываем к Sub очередные две цифры числа A и повторяем все вышеописанные действия, начиная с цикла вычитаний.

Итоговый результат выводим как на консоль, так и в файл "output.txt".

**Тестирование программы**

Проведем 17 различных тестов, чтобы доказать работоспособность программы. Протестируем программу как на корректных, так и на различных некорректных значениях.

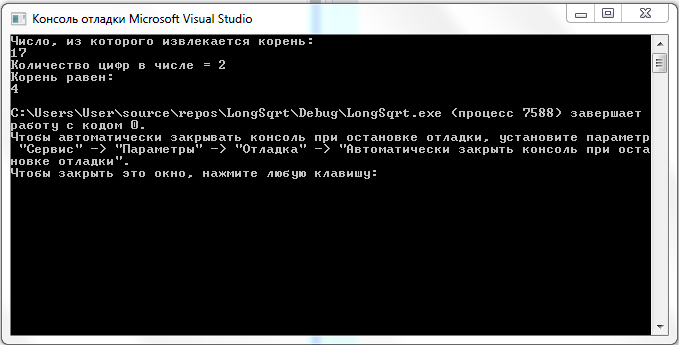
1. *Входной файл "input1.txt"*:

17

*Ожидаемый результат:*

4

*Результат работы программы (результирующий файл прилагается к отчету под именем "output1.txt"):*



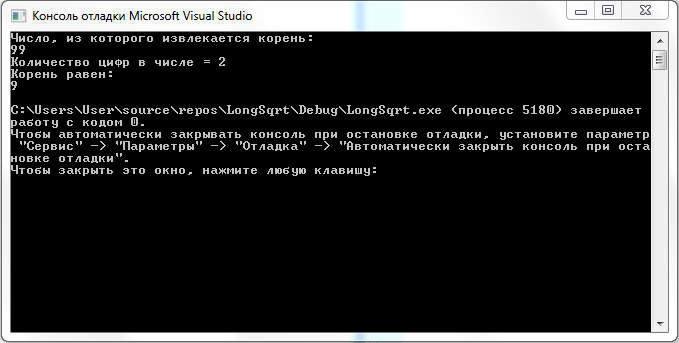
1. *Входной файл "input2.txt"*:

99

*Ожидаемый результат:*

9

*Результат работы программы (результирующий файл прилагается к отчету под именем "output2.txt"):*



1. Проверим работоспособность программы, если число A содержит незначащие нули.

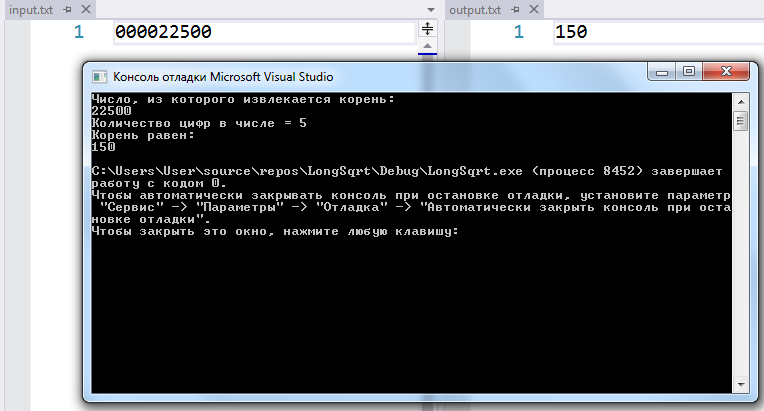
*Входной файл "input3.txt":*

000022500

*Ожидаемый результат:*

150

*Результат работы программы (результирующий файл прилагается к отчету под именем "output3.txt"):*



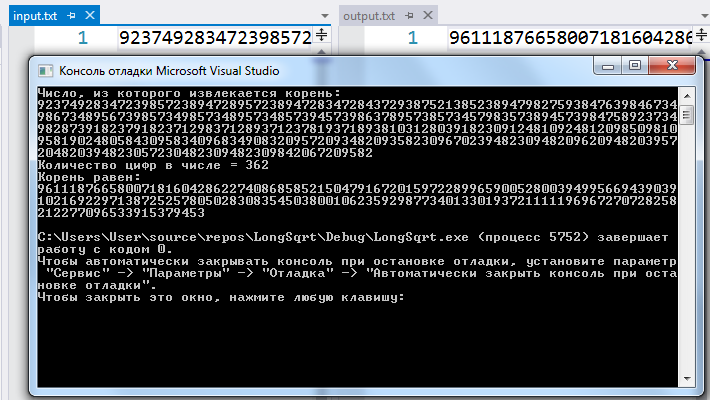
1. *Входной файл "input4.txt":*

92374928347239857238947289572389472834728437293875213852389479827593847639846734986734895673985734985734895734857394573986378957385734579835738945739847589237349828739182379182371298371289371237819371893810312803918230912481092481209850981095819024805843095834096834908320957209348209358230967023948230948209620948203957204820394823057230482309482309842067209582

*Ожидаемый результат:*

9611187665800718160428622740868585215047916720159722899659005280039499566943903910216922971387252578050283083545038001062359298773401330193721111196967270728258212277096533915379453

*Результат работы программы (результирующий файл прилагается к отчету под именем "output4.txt"):*



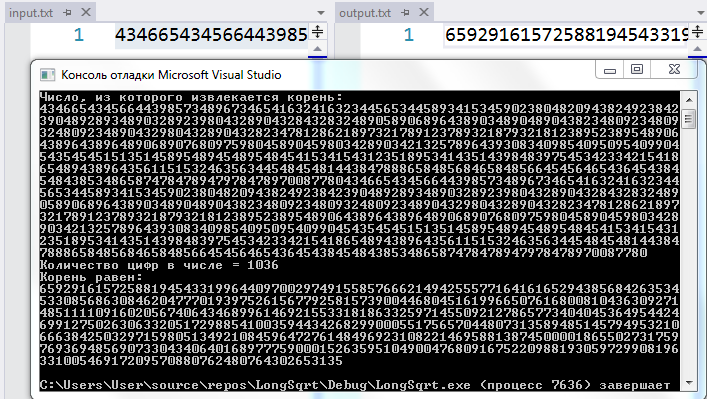
1. *Входной файл "input5.txt":*

4346654345664439857348967346541632416323445653445893415345902380482094382492384239048928934890328923980432890432843283248905890689643890348904890438234809234809324809234890432980432890432823478128621897321789123789321879321812389523895489064389643896489068907680975980458904598034289034213257896439308340985409509540990454354545151351458954894548954845415341543123518953414351439848397545342334215418654894389643561151532463563445484548144384788865848568465848566454564654364543845484385348658747847894797847897008778043466543456644398573489673465416324163234456534458934153459023804820943824923842390489289348903289239804328904328432832489058906896438903489048904382348092348093248092348904329804328904328234781286218973217891237893218793218123895238954890643896438964890689076809759804589045980342890342132578964393083409854095095409904543545451513514589548945489548454153415431235189534143514398483975453423342154186548943896435611515324635634454845481443847888658485684658485664545646543645438454843853486587478478947978478970087780

*Ожидаемый результат:*

65929161572588194543319964409700297491558576662149425557716416165294385684263534533085686308462047770193975261567792581573900446804516199665076168008104363092714851111091602056740643468996146921553318186332597145509212786577340404536495442469912750263063320517298854100359443426829900055175657044807313589485145794953210666384250329715980513492108459647276148496923108221469588138745000018655027317597693694856907330434064016897775900015263595104900476809167522098819305972990819633100546917209570880762480764302653135

*Результат работы программы (результирующий файл прилагается к отчету под именем "output5.txt"):*



1. Проверим некорректное значение. A должно быть натуральным числом, значит, не должно равняться нулю.

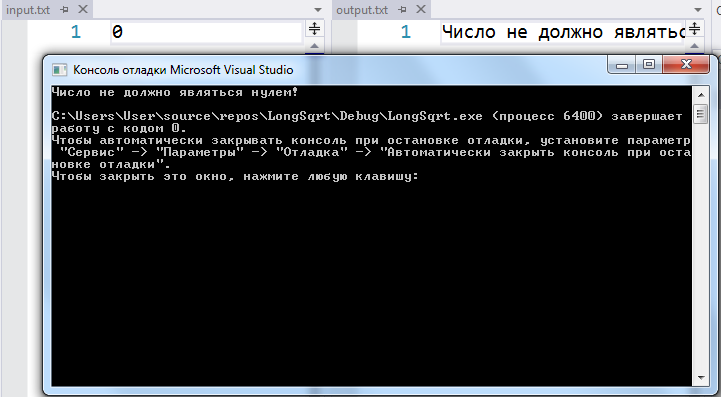
*Входной файл "input6.txt":*

0

*Ожидаемый результат:*

Число не должно являться нулем!

*Результат работы программы (результирующий файл прилагается к отчету под именем "output6.txt"):*



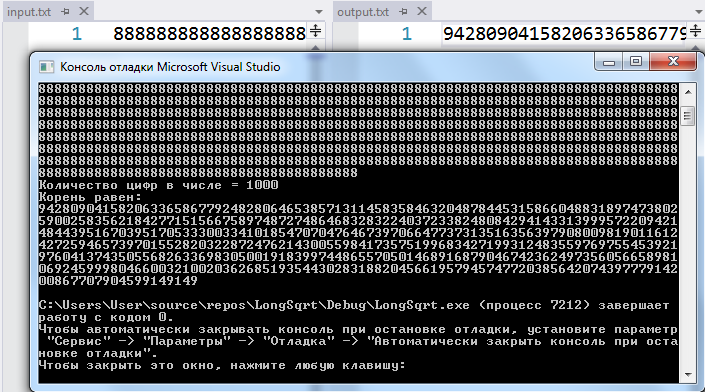
1. *Входной файл "input7.txt":*

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

*Ожидаемый результат:*

94280904158206336586779248280646538571311458358463204878445315866048831897473802590025835621842771515667589748727486468328322403723382480842941433139995722094214844395167039517053330033410185470704764673970664773731351635639790800981901161242725946573970155282032287247621430055984173575199683427199312483559769755453921976041374350556826336983050019183997448655705014689168790467423624973560566589810692459998046600321002036268519354430283188204566195794574772038564207439777914200867707904599149149

*Результат работы программы (результирующий файл прилагается к отчету под именем "output7.txt"):*



1. Проверим работоспособность программы в случае, если в файле "input.txt" содержатся лишние символы.

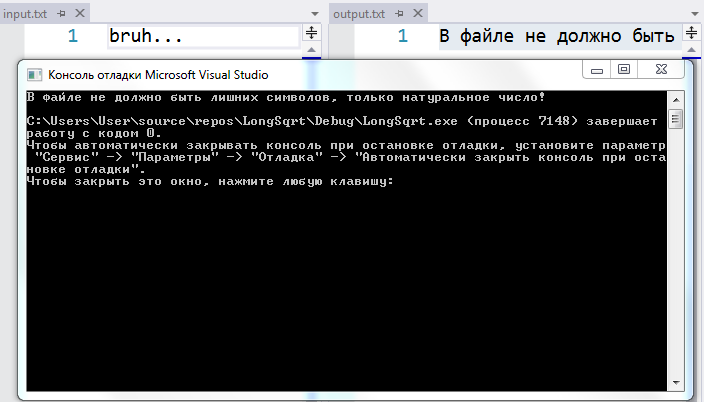
*Входной файл "input8.txt":*

bruh...

*Ожидаемый результат:*

В файле не должно быть лишних символов, только натуральное число!

*Результат работы программы (результирующий файл прилагается к отчету под именем "output8.txt"):*



1. Экстремальное тестирование – число A состоит из 3000 девяток.

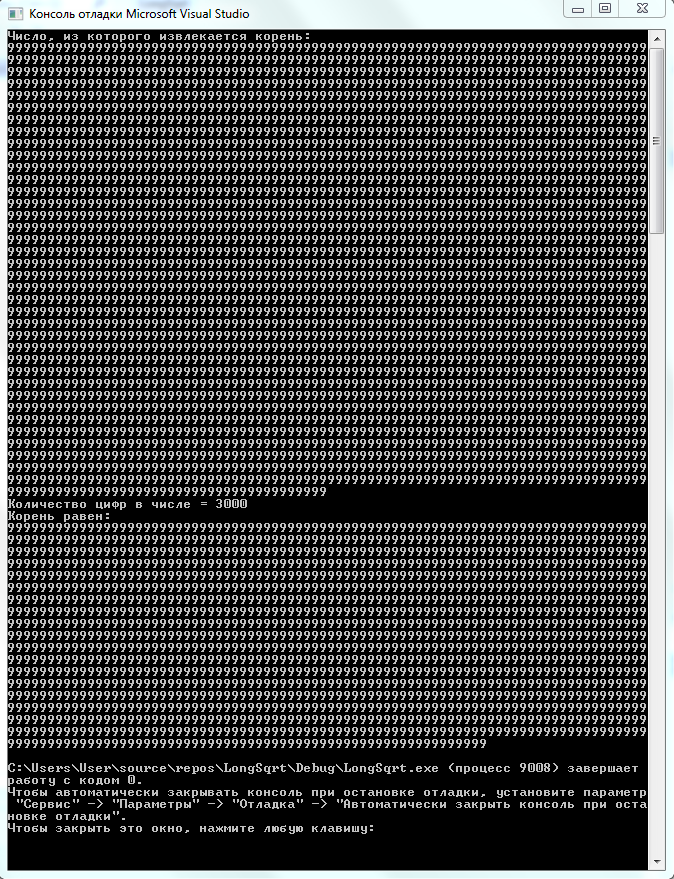
*Входной файл "input9.txt":*

999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999

*Ожидаемый результат:*

999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999

*Результат работы программы (результирующий файл прилагается к отчету под именем "output9.txt"):*



1. В файле хранится максимально возможное число – 10^3000.

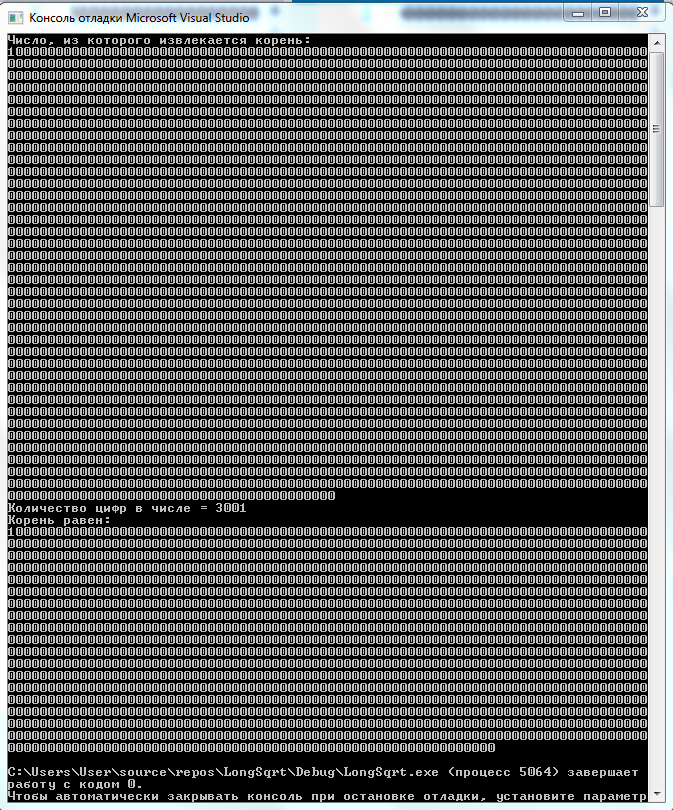
*Входной файл "input10.txt":*

1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

*Ожидаемый результат:*

1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

*Результат работы программы (результирующий файл прилагается к отчету под именем "output10.txt"):*



1. Некорректное значение – число превышает максимально возможное на 1.

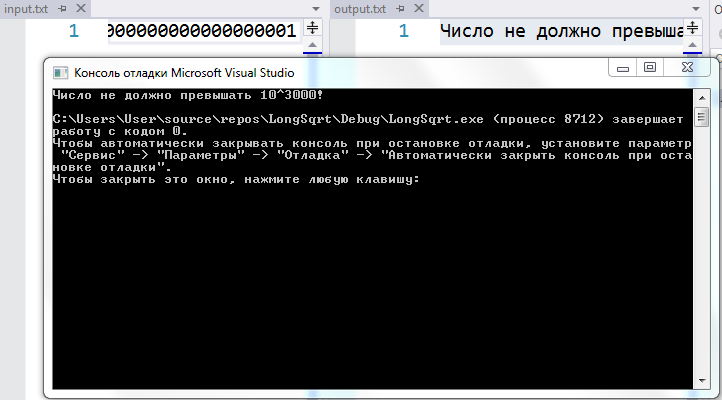
*Входной файл "input11.txt":*

1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001

*Ожидаемый результат:*

Число не должно превышать 10^3000!

*Результат работы программы (результирующий файл прилагается к отчету под именем "output11.txt"):*



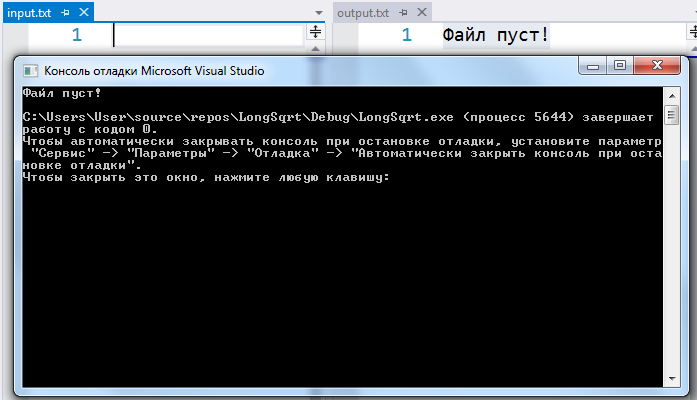
1. Входной файл не содержит никаких символов.

*Входной файл "input12.txt":*

*Ожидаемый результат:*

Файл пуст!

*Результат работы программы (результирующий файл прилагается к отчету под именем "output12.txt"):*



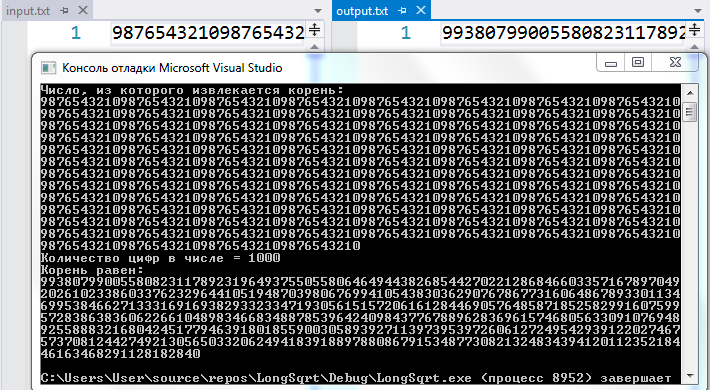
1. *Входной файл "input13.txt":*

9876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210

*Ожидаемый результат:*

99380799005580823117892319649375505580646494438268544270221286846603357167897049202610233860337623296441051948703980676994105438303629076786773160648678933011346995384662713331691693829332334719305615157206161284469057648587185258299160759957283863836062266104898346683488785396424098437767889628369615746805633091076948925588832168042451779463918018559003058939271139739539726061272495429391220274675737081244274921305650332062494183918897880867915348773082132483439412011235218446163468291128182840

*Результат работы программы (результирующий файл прилагается к отчету под именем "output13.txt"):*



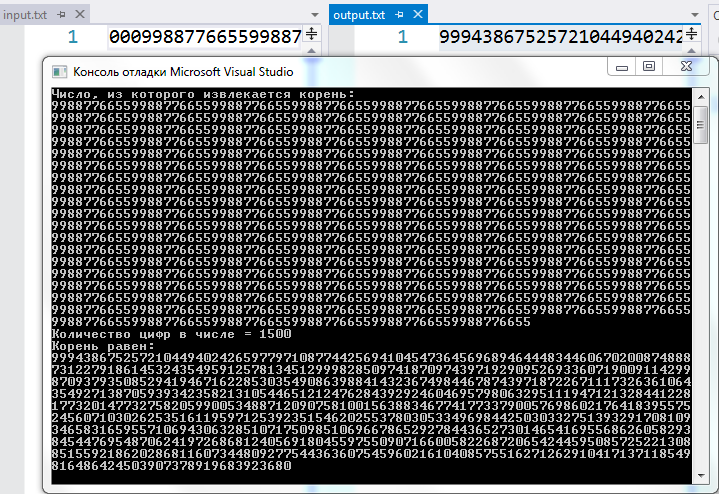
1. *Входной файл "input14.txt":*

000998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655998877665599887766559988776655

*Ожидаемый результат:*

999438675257210449402426597797108774425694104547364569689464448344606702008748887312279186145324354959125781345129998285097418709743971929095269336071900911429987093793508529419467162285303549086398841432367498446787439718722671117326361064354927138705939342358213105446512124762843929246046957980632951119471213284412281773201477327582059900534887120907581001563883467741773379005769860217641839557524560710302625351611959712539235154620255378030533496984425030332751393291708109346583165955710694306328510717509851069667865292784436527301465416955686260582938454476954870624197268681240569180455975509071660058226872065424459508572522130885155921862028681160734480927754436360754596021610408575516271262910417137118549816486424503907378919683923680

*Результат работы программы (результирующий файл прилагается к отчету под именем "output14.txt"):*



1. Некорректное значение - превышено допустимое количество цифр в числе A.

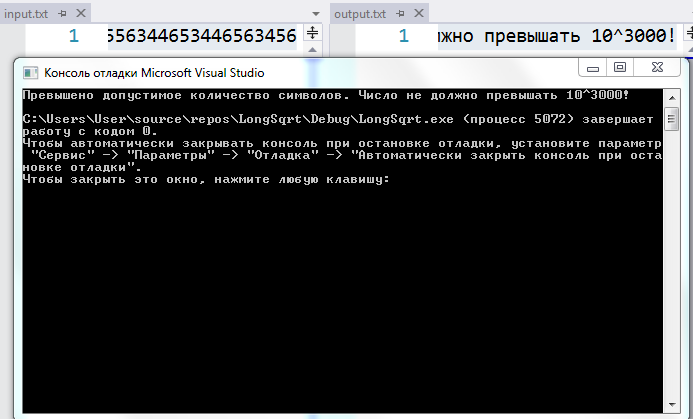
*Входной файл "input15.txt":*

897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456897865561515344556344653446563456344653446563456897865589786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345689786556151534455634465344656345634465344656345689786556151534455634465344656345661515344556344653446563456

*Ожидаемый результат:*

Превышено допустимое количество символов. Число не должно превышать 10^3000!

*Результат работы программы (результирующий файл прилагается к отчету под именем "output15.txt"):*



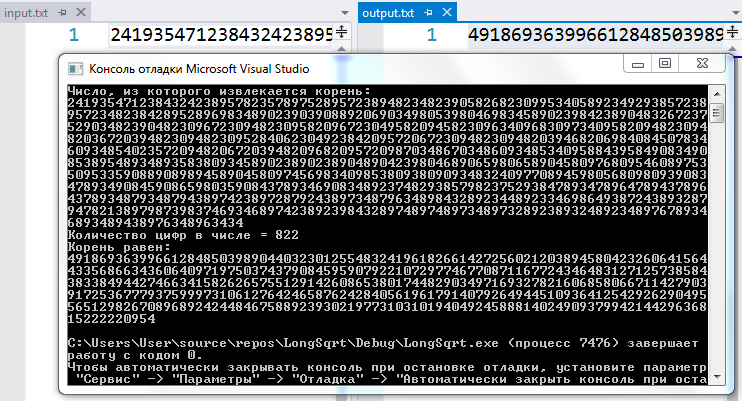
1. *Входной файл "input16.txt":*

241935471238432423895782357897528957238948234823905826823099534058923492938572389572348238428952896983489023903908892069034980539804698345890239842389048326723752903482390482309672309482309582096723049582094582309634096830973409582094823094820367203948230948230952840623049238420957206723094823094820394682069840845078346093485402357209482067203948209682095720987034867034860934853409588439584908349085389548934893583809345890238902389048904239804689065980658904580976809546089753509533590889089894589045809745698340985380938090934832409770894598056809809390834789349084590865980359084378934690834892374829385798237529384789347896478943789643789348793487943897423897287924389734879634898432892344892334698649387243893287947821389798739837469346897423892398432897489748973489732892389324892348976789346893489438976348963434

*Ожидаемый результат:*

491869363996612848503989044032301255483241961826614272560212038945804232606415644335686634360640971975037437908459590792210729774677087116772434648312712573858438338494427466341582626575512914260865380174482903497169327821606858066711427903917253677793759997310612764246587624284056196179140792649445109364125429262904955651298267089689242448467588923930219773103101940492458881402490937994214429636815222220954

*Результат работы программы (результирующий файл прилагается к отчету под именем "output16.txt"):*



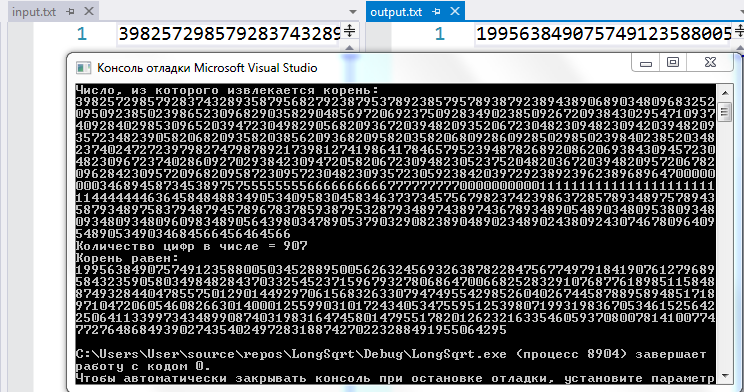
1. *Входной файл "input17.txt":*

3982572985792837432893587956827923879537892385795789387923894389068903480968325209509238502398652309682903582904856972069237509283490238509267209384302954710937409284029853096520394723049829056820936720394820935206723048230948230942039482093572348239058206820935820385620936820958203582068092860928502985023984023852034823740247272397982747987892173981274198641784657952394878268920862069384309457230482309672374028609270293842309472058206723094823052375204820367203948209572067820962842309572096820958723095723048230935723059238420397292389239623896896470000000034689458734538975755555555666666666677777777700000000001111111111111111111111114444444636458484883490534095830458346373734575679823742398637285789348975789435879348975837948794578967837859387953287934897438974367893489054890348095380934809348093480960983489056439803478905379032908238904890234890243809243074678096409548905349034684566456464566

*Ожидаемый результат:*

1995638490757491235880050345288950056263245693263878228475677497918419076127968958432359058034984828437033254523715967932780686470066825283291076877618985115848874932844047855750129014492970615683263307947495542985260402674458788958948517189710472060546082663014000125599031017243405347559512539807199319836705346152564225064113399734348990874031983164745801479551782012623216335460593708007814100774772764868493902743540249728318874270223288491955064295

*Результат работы программы (результирующий файл прилагается к отчету под именем "output17.txt"):*



**Код программы**

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

void del\_zero(string &S) // функция для удаления незначащих нулей из исходного числа

{

if (S.length() > 1 && S[0] == '0')

{

while (S[0] == '0')

S.erase(0, 1);

}

}

string oborona(string S)

{

string P = "";

if (S == "")

{

P = "Файл пуст!";

}

if (S[0] == '0' && S.length() == 1)

P = "Число не должно являться нулем!";

if (S.length() <= 3001)

{

for (int i = 0; i < S.length(); i++)

{

if (S[i] > '9' || S[i] < '0')

{

P = "В файле не должно быть лишних символов, только натуральное число!";

}

}

}

if (S.length() == 3001)

{

if (S[0] > '1' && S[0] <= '9')

P = "Число не должно превышать 10^3000!";

else if (S[0] == '1')

{

for (int i = 1; i < 3001; i++)

{

if (S[i] > '0' && S[i] <= '9')

P = "Число не должно превышать 10^3000!";

}

}

}

if (S.length() > 3001)

{

P = "Превышено допустимое количество символов. Число не должно превышать 10^3000!";

}

return P;

}

int sravnenie(string Sub, string V) // функция для сравнения

{

if (Sub.length() > V.length())

{

while (Sub.length() != V.length())

{

V = '0' + V; // дополняем вычитаемое незначащими нулями, если в нем меньше разрядов, чем в "подчисле"

}

}

if (Sub.length() == V.length())

{

for (int i = 0; i < Sub.length(); i++) // поразрядное сравнение

{

if (Sub[0] <= V[0])

{

if (Sub[i] > V[i])

{

return 0;

}

if (Sub[i] < V[i])

{

return -1;

}

}

}

}

return 0;

}

string minus\_Sub(string Sub, string V) // функция для вычитания

{

int n1 = Sub.length(), n2 = V.length();

string RaznS = "";

reverse(Sub.begin(), Sub.end()); // переворачиваем "подчисло"

reverse(V.begin(), V.end()); // переворачиваем вычитаемое

for (int i = n2; i < n1; i++) // дополняем вычитаемое незначащими нулями, если в нем меньше разрядов, чем в "подчисле"

V.push\_back('0');

int mns = 0; // остаток от деления на 10 этой переменной будет результатом разницы разрядов

for (int i = 0; i < n1; i++)

{

mns = mns + (Sub[i] - '0') - (V[i] - '0') + 10;

RaznS.push\_back(mns % 10 + '0');

if (mns <= 9)

mns = -1; // если разряд "подчисла" меньше разряда вычитаемого, то занимаем единицу у старшего разряда "подчисла"

else

mns = 0;

}

reverse(RaznS.begin(), RaznS.end()); // не забываем перевернуть строку, чтобы получить нужное число

return RaznS;

}

string plus\_Two(string V) // функция для прибавления двойки

{

int n1 = V.length();

string SumV = "";

reverse(V.begin(), V.end()); // переворачиваем число для сложения столбиком

int perenos = 0; // переменная, куда будет записываться остаток от сложения

int sum = (V[0] - '0') + 2 + perenos;

SumV.push\_back(sum % 10 + '0'); // от получившейся суммы берем остаток от деления на 10 и добавляем в конец строки

perenos = sum / 10; // делим сумму на 10 нацело - получаем остаток для переноса в следующий разряд

for (int i = 1; i < n1; i++) // в этом цикле, если остаток равен нулю, цифры в разрядах числа остаются без изменений

{

int sum = ((V[i] - '0') + perenos);

SumV.push\_back(sum % 10 + '0');

perenos = sum / 10;

}

if (perenos != 0) SumV.push\_back(perenos + '0'); // если остаток в итоге не равен нулю, то тоже добавляем его в конец строки

reverse(SumV.begin(), SumV.end()); // не забываем перевернуть строку, чтобы получить нужное число

return SumV;

}

int main()

{

setlocale(LC\_ALL, "Russian");

ifstream fin("input.txt");

ofstream fout("output.txt");

string S;

fin >> S; // считываем число из файла в виде строки

del\_zero(S); // удаляем незначащие нули, если такие имеются

if (oborona(S) != "")

{

cout << oborona(S) << endl;

fout << oborona(S);

}

else

{

cout << "Число, из которого извлекается корень:" << endl;

cout << S << endl;

int N = S.length(); // количество цифр в числе

cout << "Количество цифр в числе = " << N << endl;

string Sub;

if (N % 2 == 0) // если количество цифр чётное, берём за "подчисло" два старших разряда, если нет - один

{

Sub.push\_back(S[0]);

Sub.push\_back(S[1]);

}

else

{

Sub.push\_back(S[0]);

}

string V = "1"; // первое вычитаемое число

string One = "1"; // понадобится после каждого цикла вычитаний

string kor;

int k = 0; // счетчик количества вычитаний

while (V.length() <= Sub.length() && sravnenie(Sub, V) != -1) // в этом цикле вычитаем V из "подчисла" до тех пор, пока V не станет больше "подчисла"

{

Sub = minus\_Sub(Sub, V); // вычитаем из "подчисла" нечетное число

V = plus\_Two(V); // увеличиваем это нечетное число на 2

k++;

}

kor.push\_back(k + '0'); // Получившееся число вычитаний - первая цифра корня, добавляем её в строку

k = 0;

V = minus\_Sub(V, One); // Так как вычитаемое на последнем шаге лишний раз увеличивается на 2, а нам надо лишь на 1, вычитаем 1

V.push\_back('1'); // умножаем вычитаемое на 10 и прибавляем 1 (по сути, просто дописываем в конец единицу)

int i = 0;

if (N > 2)

{

if (N % 2 != 0) // определяем, от какой по счёту цифры шагаем после первого цикла вычитаний

i = 1;

else

i = 2;

for (i; i < N; i += 2)

{

Sub.push\_back(S[i]);

Sub.push\_back(S[i + 1]); // после каждого цикла вычитаний добавляем к "подчислу" по две цифры от исходного числа

while (V.length() <= Sub.length() && sravnenie(Sub, V) != -1)

{

Sub = minus\_Sub(Sub, V);

V = plus\_Two(V);

k++;

}

kor.push\_back(k + '0'); // получившееся число вычитаний - это следующая цифра корня, добавляем её в конец строки

k = 0;

V = minus\_Sub(V, One); // Так как вычитаемое на последнем шаге лишний раз увеличивается на 2, а нам надо лишь на 1, вычитаем 1

V.push\_back('1'); // умножаем вычитаемое на 10 и прибавляем 1 (по сути, просто дописываем 1 в конец)

}

}

cout << "Корень равен:" << endl << kor << endl;

fout << kor;

}

}