

Задачи к лабораторной работе на тему «Программирование на языке C++ с использованием строк»

Цель работы:

Изучить особенности написания программ на языке C++ с использованием строк.

Указания к выполнению работы

При решении задач 1-3 необходимо использовать не менее трех различных методов работы со строками, соответствующих заданному представлению строки. Недопустимо использовать функции для работы с нуль-терминальными строками.

Для задач 1 и 2 необходимо представить строку как объект класса `string`.

Для задачи 3 необходимо представить строку как объект встроенного класса строк среды разработки (`QString` для Qt или `System::String` для Visual Studio). Недопустимо использовать способ представления и функции класса `string`.

Индивидуальные варианты заданий

Задача 1

1. Дан объект, в котором встречаются буквы «и» и «т». Определить, какая из них встречается позже (при просмотре текста слева направо). Если таких букв несколько, то должны учитываться последние из них.

2. Дан текст, в котором имеется одна буква «а» и одна буква «о». Определить, сколько символов находится между буквами «а» и «о».

3. Дан текст, в котором имеется одна буква «к» и как минимум одна буква «о». Вывести текст, расположенный между буквой «к» и крайней справа буквой «о».

4. Дан текст, предложения в котором разделены точкой. Определить сколько в тексте предложений.
5. Дан текст, слова в котором разделены одинарным символом пробела (символ «—» в тексте отсутствует). Верно ли, что число слов в предложении больше трех?
6. Дан текст. Определить верно ли, что букв «а» в тексте больше, чем букв «о»?
7. Дан текст, слова в котором разделены одинарным символом пробела (символ «—» в тексте отсутствует). Определить сколько в тексте слов.
8. Дан текст. Определить сколько раз в тексте встречается его последняя буква (учесть, что текст может содержать не только буквы).
9. Дан текст, предложения в котором разделены точкой. Определить, сколько раз в первом предложении текста встречается буква «а» (предусмотреть вариант, что в первом предложении такой буквы может не быть).
10. Дан текст. Напечатать все символы текста, расположенные между первой и второй запятой. Если второй запятой нет, то надо вывести все символы, расположенные после первой запятой. Если запятых нет – вывести весь текст.
11. Дан текст и символ `symbol`. Удвоить каждое вхождение символа `symbol` в тексте.
12. Дан текст и какая-то буква. Определить количество заданных букв в тексте, предшествующих первой запятой (если запятая отсутствует – определить общее количество заданных букв).
13. Дан текст. Определить количество цифр, встречающихся в тексте.
14. Дан текст. Определить верно ли, что в нем есть четыре подряд расположенных одинаковых символа.
15. Дан текст. Определить сколько в тексте одинаковых соседних букв.

16. Дан текст. Определить первую и последнюю пару одинаковых соседних символов в тексте.

17. Даны два слова, разделенные пробелом. Определить, сколько начальных букв первого слова совпадает с начальными буквами второго слова.

18. Дан текст, слова в котором разделены одинарным символом пробела. Вывести слово, содержащее удвоенную букву «н» (если подобных слов несколько – вывести последнее из них).

19. Дан текст, слова в котором разделены одинарным символом пробела (символ «—» в тексте отсутствует). Вывести все слова, в которых встречается буквосочетание «ее».

20. Дан текст. Проверить, правильно ли записаны буквосочетания «ча» и «ща» и исправить ошибки.

21. Дан текст. Определить частоту появления букв «а» в нем. Частота вычисляется как отношение количества данных символов в тексте к длине всего текста (пробелы учитываются, а символ конца строки не учитывается).

22. Дан текст. Определить, сколько в нем гласных букв.

23. Дан текст, слова в котором разделены произвольным количеством пробелов. Удалить повторяющиеся пробелы между отдельными словами (оставляя по одному пробелу).

24. Дан текст, слова в котором разделены одинарным символом пробела, и какая-то буква. Посчитать количество слов, начинающихся на заданную букву.

25. Дан текст, слова в котором разделены одинарным символом пробела. Вывести все слова текста, которые начинаются и заканчиваются на одну и ту же букву.

26. Дан текст, слова в котором разделены одинарным символом пробела, и какая-то буква. Вывести все слова, в которых встречается эта буква.

27. Дан текст, слова в котором разделены одинарным символом пробела (в тексте могут встречаться запятые), и какая-то буква. Вывести все слова текста, у которых последняя буква не совпадает с заданной.

28. Дан текст, слова в котором разделены одинарным символом пробела (знаки пунктуации отсутствуют). Определить сколько в тексте слов-перевертышей.

29. Дан текст, слова в котором разделены одинарным символом пробела (знаки пунктуации отсутствуют). Вывести все симметричные слова текста, отличные от первого.

30. Дан текст, слова в котором разделены одинарным символом пробела (знаки пунктуации отсутствуют), какая-то буква и целое n . Вывести все слова текста, в которых заданная буква встречается ровно n раз.

31. Описание химического вещества - строка, в которой входящие в него атомы химических элементов перечисляются в определенном порядке. При этом последовательности из двух и более одинаковых атомов, идущих подряд, группируются: записывается сокращенное название химического элемента и количество одинаковых элементов подряд. Например, вместо NN пишут N_2 . Обозначения химических элементов состоят из одной или двух английских букв, первая из которых должна быть прописная, а вторая - строчная. Задана непустая последовательность символов, содержащая только цифры и строчные и прописные английские буквы. Длина последовательности от 1 до 1000 символов. Гарантируется, что в последовательности все однобуквенные и двухбуквенные подстроки, начинающиеся с прописной буквы - правильные обозначения химических элементов. Вы должны проверить, что заданная последовательность символов подходит под данное выше описание формулы химического вещества. Например, строки « H_2O » и « NaCl » - правильное описание вещества, а строки « cO_2 » или « COO » - неправильное.

32. Главный вождь племени Абба не умеет считать. В обмен на одну из его земель вождь другого племени предложил ему выбрать одну из трех куч с золотыми монетами. Но вождю племени Абба хочется получить наибольшее количество золотых монет. Помогите вождю сделать правильный выбор.

Входные данные - три натуральных числа через пробел. Каждое из чисел не превышает 10^{100} . Числа записаны без ведущих нулей.

Выходные данные - одно целое число — максимальное количество монет, которые может взять вождь.

33. Марсиане Миша и Маша решили вместе подобрать подарок на день рождения Кати. Когда они наконец нашли то, что хотели, и упаковали предмет в красивую коробку, надо было решить, как подписать подарок. Друзья подумали, что лучшим решением будет составить общую подпись так, чтобы в ней как подстроки содержались их имена. Учтите, что на Марсе принято подписываться полными именами, а они у марсиан могут быть достаточно длинными.

Входные данные - две строки, в которых записаны полные имена друзей. Имена, как ни странно, состоят из букв английского алфавита, из которых только первая – прописная. Длина имен от 1 до 1000 символов.

Выходные данные - вывести кратчайшую строку, в которой встречаются имена Миши и Маши одновременно. Буквы, с которых имена начинаются в этой строке, нужно сделать большими. Если существует несколько решений, вывести то, которое меньше в алфавитном порядке (следует считать, что латинская буква в верхнем регистре меньше, чем любая буква в нижнем регистре).

Примеры входных данных

Julya

Lyalya

Примеры выходных данных

Julyalya

34. Многие команды, участвующие в командных соревнованиях по программированию, используют так называемый «шаблон программы». Он набирается в самом начале соревнования и содержит общее для всех решений - например, открытие и закрытие входных и выходных файлов. Трехкратный чемпион мира по версии АМС команда Dream Team – не исключение. Во

многим их успехи связаны с тем, что они очень тщательно готовятся к соревнованиям, продумывая даже очень мелкие детали. Например, перед последним финалом они во время пробного тура рассчитали, сколько джоулей энергии потратится на набор шаблона. Организаторы финала использовали весьма странные клавиатуры — жесткость различных клавиш была различной. Таким образом, на нажатие разных клавиш требовалось различное количество энергии. Эксперименты, проведенные командой Dream Team во время пробного тура, показали следующее. На набор строчной буквы английского алфавита требуется количество энергии, равное сумме цифр ее порядкового номера в алфавите (буквы нумеруются с единицы). На нажатие клавиши «Shift» требуется 10 джоулей энергии (таким образом набор заглавной буквы английского алфавита требует на 10 джоулей больше, чем набор соответствующей ей строчной буквы), нажатие клавиши «Пробел» требует 4 джоуля энергии. Набор цифры x требует $(13 - x)$ джоулей энергии, набор точки — 5 джоулей, точки с запятой — 7 джоулей, запятой — 2 джоуля. Знак равенства, плюс, минус, одинарная и двойная кавычка требуют по 3 джоуля энергии. Закрывающая и открывающая круглые скобки требуют по 1 джоулю, а фигурные, квадратные и угловые (т.е. символы < и >) — по 8. При этом для всех упомянутых знаков препинания на клавиатуре, используемой на финале, существуют отдельные клавиши, и другой возможности набрать соответствующий символ нет. Нажатие клавиши «Enter» (перевод строки) оказалось настолько легким, что энергозатраты на него можно считать нулевыми. Ваша задача — написать программу, которая по тексту шаблона вычислит энергозатраты на его набор.

Входные данные - шаблон программы, энергетические затраты на набор которого необходимо вычислить. Он содержит только цифры, пробелы, строчные и заглавные буквы английского алфавита, точки, запятые, знаки равенства, плюсы, точки с запятыми, двойные кавычки ("), одинарные кавычки ('), закрывающие и открывающие круглые, фигурные и квадратные скобки. Его размер не превышает 20000 байт.

Пример входных данных

abCD '+=1;2,3."()[]{}

Пример выходных данных

127

35. Одно из известных развлечений со словами – составление палиндромов. Палиндромом называется предложение, которое, после удаления из него всех пробелов и знаков препинания, читается одинаково справа налево и слева направо. Создатели одного известного текстового редактора пишут новую версию модуля для проверки орфографии. Они хотят реализовать возможность вывода подсказки для пользователя на тот случай, если он допустил опечатку при наборе какого-нибудь палиндрома. Конечно же, они решили обратиться именно к вам. Более точно, по заданной строке нужно определить, может ли она быть результатом замены, удаления или добавления не более чем одного символа в некотором палиндроме. При этом строчные и прописные английские буквы не различаются, а все остальные символы должны игнорироваться.

Входные данные – строка (гарантируется, что она содержит хотя бы одну букву), длина которой не превосходит 10^5 .

Выходные данные

В первой строке выведите YES, если строка может быть получена каким-нибудь из описанных выше преобразований из некоторого палиндрома, и NO в противном случае. В случае положительного ответа во второй строке выведите какой-нибудь из палиндромов, в которых мог допустить опечатку пользователь.

Задача 2

1. Дан текст и строка. Подсчитать количество вхождений строки в текст.
2. Дана строка из символов латинского алфавита. Замените все вхождения строки "two" на символ '2'.

3. Дан текст, слова в котором разделены одинарным символом пробела. Для каждого слова текста вывести сколько в нем букв, совпадающих с его первой буквой.

4. Дан текст. Зашифровать его буквы методом Цезаря по ключу 2. Каждая буква, являющаяся i -й по алфавиту, заменяется на $(i + 2)$ -ю букву, предпоследняя буква по алфавиту заменяется первой и т.д.

5. Дан текст. Заменить в нем стоящие подряд одинаковые символы одним символом и подсчитать количество таких замен.

6. Дан текст. Выполнить форматирование текста следующим образом: каждое предложение текста должно иметь свой порядковый номер и начинаться с красной строки.

7. Дан текст. Посчитать в нем сумму всех встречающихся в нем цифр.

8. Дан текст, слова в котором разделены одинарным символом пробела. Посчитать в нем количество однобуквенных слов.

9. Дан текст. Определить количество слогов текста.

10. Дан текст, слова в котором разделены одинарным символом пробела. Определить наиболее часто встречающийся символ в самом длинном слове этого текста.

11. Дан текст, слова в котором разделены одинарным символом пробела. Определить число символов и число слов текста (пробелы входят в число символов)

12. Дан текст, слова в котором разделены одинарным символом пробела, и строка, состоящая из букв. Вывести все слова текста, которые содержат хотя бы одну букву из заданной строки.

13. Дан текст, слова в котором разделены одинарным символом пробела. Определить сколько раз заданная буква (буква вводится пользователем) встречается в каждом слове текста.

14. Дан текст, слова в котором разделены одинарным символом пробела. Заменить самое длинное слово (слова) теста на введенное пользователем слово.

15. Дан текст, слова в котором разделены одинарным символом пробела. Заменить в нем первые буквы всех слов на заглавные (если слово начинается с заглавной буквы, оставляет его без изменения).

16. Дан текст, слова в котором разделены одинарным символом пробела. Сформировать новый текст на основе данного, переставляя слова в обратном порядке (т.е. первое слово будет последним, второе – предпоследним и т.д.).

17. Дано предложение (из знаков препинания встречаются только точки и запятые). Перевернуть каждое четное слово предложения, содержащее нечетное число букв.

18. Дан текст, слова в котором разделены одинарным символом пробела. Определить слово (слова) текста с минимальным числом повторяющихся букв.

19. Дан текст, слова в котором разделены одинарным символом пробела. Определить буквы, которые встречаются в каждом из слов текста по одному разу или вывести сообщение, что таких букв нет.

20. Дан текст, слова в котором разделены одинарным символом пробела. Определить слова, в которых каждая буква встречается не менее двух раз.

21. Дан текст на английском языке, слова в котором разделены одинарным символом пробела. Для каждой буквы алфавита вывести, в скольких словах этого текста она встречается с учетом регистра и сами слова (например, «a – 1: programming», «b – 0» и т. д.).

22. Дан текст на английском языке, слова в котором разделены одинарным символом пробела. Определить слово (в случае нескольких слов, удовлетворяющих условию – найти все слова) текста, в котором доля гласных (a, e, i, o, u, y) максимальна.

23. Дан текст, слова в котором разделены одинарным символом пробела. Удалить в нем все слова заданной длины, вывести оставшиеся слова текста в алфавитном порядке.

24. Дан текст. Определить максимальное по длине слово текста, являющееся палиндромом.

25. Дан текст на английском языке, слова в котором разделены одинарным символом пробела. Вывести слова заданного текста в соответствии с ростом доли гласных в этих словах (а, е, i, о, u, y) - отношения количества гласных букв к общему количеству букв в слове.

26. Дан текст на английском языке, слова в котором разделены одинарным символом пробела. Определить слово (слова) с максимальным числом повторяющихся букв.

27. Дан текст на английском языке, слова в котором разделены одинарным символом пробела. Отсортировать слова текста по убыванию в них доли согласных букв (отношение количества букв к общему количеству букв в слове).

28. Дан текст, слова в котором разделены одинарным символом пробела. Найти в нем самое короткое слово-палиндром.

29. Дан текст, слова в котором разделены одинарным символом пробела. Найти в нем все пятибуквенные слова-палиндромы.

30. Дан текст, слова в котором разделены одинарным символом пробела и строка. Найти в нем все слова-палиндромы, заканчивающиеся на заданную строку.

31. Палиндромом называют строку, читающуюся одинаково с обеих сторон. Задана строка s . Найдите ее наибольшую по длине подстроку, не являющуюся палиндромом.

Входные данные строка, состоящая только из строчных букв английского алфавита, не пуста, а ее длина не превышает 10^5 символов.

Выходные данные - ответ на задачу. Если все подстроки s являются палиндромами, выведите NO SOLUTION.

32. Дана непустая строка S длиной N символов. Будем считать, что элементы строки нумеруются от 1 до N . Для каждой i -й позиции строки S определим подстроку, заканчивающуюся в этой позиции, которая совпадает с некоторым началом всей строки S и имеет длину, меньшую, чем i (т.е. не равна i -му префиксу исходной строки). Значением префикс-функции $P(i)$ будем считать длину этой подстроки. Требуется для всех i от 1 до N вычислить значение $P(i)$.

Входные данные - строка, состоящая из символов с кодами ASCII от 33 до 127. Длина строки не превышает 10^6 .

Выходные данные - все значения префикс-функции.

33. Дана непустая строка S длиной N символов. Будем считать, что элементы строки нумеруются от 1 до N . Для каждой i -й позиции строки S определим Z -блок как наибольшую подстроку, которая начинается в этой позиции и совпадает с некоторым началом всей строки S . Значением Z -функции $Z(i)$ будем считать длину этого Z -блока. При $i = 1$ будем считать, что $Z(1) = 0$, несмотря на то, что в начале строки строка совпадает сама с собой. Требуется для всех i от 1 до N вычислить значение $Z(i)$.

Входные данные - строка, состоящая из символов с кодами ASCII от 33 до 127. Длина строки не превышает 10^6 .

Выходные данные - все значения Z -функции.

34. Для игры в «Поле чудес» используется круглый барабан, разделенный на секторы, и стрелка. В каждом секторе записано некоторое число. В различных секторах может быть записано одно и то же число. Однажды ведущий игры решил изменить правила. Он сам стал вращать барабан и называть игроку (который барабана не видел) все числа подряд в том порядке, в котором на них указывала стрелка в процессе вращения барабана. Получилось так, что барабан сделал целое число оборотов, то есть последний сектор совпал с первым. После этого, ведущий задал участнику вопрос: какое наименьшее число секторов может быть на барабане? Требуется написать программу, отвечающую на этот вопрос ведущего.

Входные данные - в первой строке записано число N – количество чисел, которое назвал ведущий ($2 \leq N \leq 30000$). Во второй строке записано N чисел, на которые указывала стрелка в процессе вращения барабана. Первое число всегда совпадает с последним (в конце стрелка указывает на тот же сектор, что и в начале). Числа, записанные в секторах барабана – натуральные, не превышающие 32000.

Выходные данные – минимальное число секторов, которое может быть на барабане.

35. Мальчик Кирилл написал однажды на листе бумаги строчку, состоящую из больших и маленьких английских букв, а после этого ушел играть в футбол. Когда он вернулся, то обнаружил, что его друг Дима написал под его строкой еще одну строчку такой же длины. Дима утверждает, что свою строчку он получил циклическим сдвигом строки Кирилла направо на несколько шагов (циклический сдвиг строки abcde на 2 позиции направо даст строку deabc). Однако Дима известен тем, что может случайно ошибиться в большом количестве вычислений, поэтому Кирилл в растерянности - верить ли Диме? Помогите ему! По данным строкам выведите минимально возможный размер сдвига вправо или -1, если Дима ошибся.

Входные данные - строки Кирилла и Димы соответственно. Строки состоят только из английских символов. Длины строк одинаковы, не превышают 10000 и не равны 0.

Выходные данные - ответ на поставленную задачу.

Задача 3

1. Дан текст. Заменить в нем все вхождения буквосочетания «ай» на «ой».
2. Дано слово. Поменять местами первую из букв «о» и последнюю из букв «а» (предусмотреть вариант, что таких букв в слове может не быть).

3. Дан текст. Удалить из него все символы, расположенные между k -м и t -м индексами (k и t - целые).
4. Дан текст. После каждого знака препинания вставить пробел (в тексте из знаков препинания встречаются только точки и запятые).
5. Дан текст. Вставить пробел перед каждой заглавной буквой текста.
6. Дан текст. Удалить из него все знаки препинания.
7. Дано слово. Если его длина четная, удалить из него первую и последнюю буквы, в противном случае – удалить среднюю букву.
8. Дан текст, слова в котором разделены одинарным символом пробела, и некоторая подстрока. Вывести слова текста, оканчивающиеся заданной подстрокой.
9. Дан текст. После каждой удвоенной буквы вставить символ '&'.
10. Дан текст. Найти наибольшее количество идущих подряд одинаковых символов в тексте.
11. Дан текст, слова в котором разделены одинарным символом пробела (из знаков препинания встречаются только точки и запятые). Удалить из текста все слова, содержащие удвоенные символы.
12. Дан текст. Заменить наиболее часто встречающийся символ на наименее часто встречающийся.
13. Дан текст, слова в котором разделены одинарным символом пробела (символ « \cdot » в предложении отсутствует). В каждом третьем слове заменить первую букву на последнюю.
14. Даны два слова. Для каждой буквы первого слова определить, входит ли она во второе слово (повторяющиеся буквы первого слова не учитывать). Например, если заданы слова «факультет» и «университет», ответ должен быть «нет», «нет», «нет», «да», «нет», «нет», «да», «да».
15. Даны два слова. Вывести буквы, которые есть только в одном из двух слов. Например, если заданы слова «факультет» и «университет», ответ должен быть «ф», «а», «к», «л», «ь», «н», «и», «в», «р», «с», «и».

16. Дано слово. Определить сколько и какие в нем различные буквы. Например, в слове «университет» 8 различных букв – «у», «н», «и», «в», «е», «р», «с», «т».

17. Дан текст. Найти в строке самый часто встречающийся символ. Вывести на экран сам символ и число его повторений.

18. Даны два слова. Определить, можно ли из букв первого слова получить второе. Учитывать, что каждая буква второго слова должна входить в первое слово столько же раз, сколько и во второе.

19. Дан текст, слова в котором разделены одинарным символом пробела. Вывести самое короткое слово в предложении (если таких слов несколько, то вывести первое из них).

20. Дан текст, слова в котором разделены одинарным символом пробела (символ «—» в предложении отсутствует). В каждом слове с четным числом букв заменить буквосочетания «ла» на «не».

21. Дан текст, слова в котором разделены одинарным символом пробела, и целое n . Вывести на экран все слова длины n .

22. Дан текст. Проверить правильность расстановки в нем круглых скобок: каждой открытой скобке должна соответствовать корректно закрытая скобка.

23. Дан текст, слова в котором разделены одинарным символом пробела (из знаков препинания встречаются только точки и запятые). В каждом слове (с длиной не менее 3-х букв), в котором первая буква совпадает с последней поменять местами вторую и предпоследнюю буквы.

24. Дан текст, слова в котором разделены одинарным символом пробела. Для каждого слова вывести количество букв, совпадающих с его первой буквой.

25. Дано предложение. Из каждого слова, содержащего букву «т» удалить все повторяющиеся буквы. Например, для слова «автовладелец» результат будет следующий: «тодц».

26. Дан текст, слова которого разделены пробелами (из знаков препинания встречаются только точки и запятые). Преобразовать каждое слово в строке, заменив в нем все предыдущие вхождения его последней буквы на символ «+» (плюс). Например, слово «минимум» надо преобразовать в «+ини+ум». Количество пробелов между словами не изменять.

27. Дан текст. Преобразовать его так, чтобы сначала следовали символы, представляющие собой цифры, а затем все остальные. Порядок следования символов между собой не изменять.

28. Дан текст, слова которого разделены пробелами (из знаков препинания встречаются только точки и запятые). Сформировать строку, содержащую упорядоченные по алфавиту слова текста (регистр букв не менять).

29. Дан текст. Определить, какой процент слов текста содержит удвоенную согласную.

30. Дан текст вида $N_0O_1N_1O_2 \dots O_kN_k$, представляющий собой некое арифметическое выражение, в котором N_i – целое число, O_i – один из двух знаков простейших арифметических действий: сложение (+) и вычитание (-). Вычислить заданное выражение.

31. В задачах гофска часто используются так называемые хеш-функции. Одним из важнейших классов хеш-функций являются так называемые полиномиальные хеш-функции. Пусть дана строка $S = s_1s_2 \dots s_k$, состоящая из цифр от 0 до 9. Тогда значение полиномиальной хеш-функции $p(S, x, m)$ вычисляется следующим образом:

$$p(S, x, m) = \left(\sum_{i=1}^k s_i x^{i-1} \right) \bmod m$$

($a \bmod b$ обозначает остаток от деления числа a на число b). Например, пусть $S = 0123$, тогда $p(S, 2, 5) = (0 \cdot 1 + 1 \cdot 2 + 2 \cdot 4 + 3 \cdot 8) \bmod 5 = 4$.

Вам даны числа m и x и множество из n строк $(S^{(1)}, S^{(2)}, \dots, S^{(n)})$, каждая из которых состоит только из цифр от 0 до 9. Необходимо найти количество таких пар (i, j) , где $1 \leq i, j \leq n$, $i < j$, что $p(S^{(i)}, x, m) = p(S^{(j)}, x, m)$.

Входные данные - первая строка содержит три целых числа: n ($1 \leq n \leq 30000$), m ($1 \leq m \leq 2000$), x ($1 \leq x \leq 100$). Далее идут n строк, каждая из которых содержит по одной строке из данного множества: 2-ая строка входного файла содержит $S^{(1)}$, 3-я - $S^{(2)}$, ..., $(i + 1)$ -ая - $S^{(i)}$, ..., $(n + 1)$ -ая - $S^{(n)}$. Длины $S^{(i)}$ не превосходят 100, $S^{(i)}$ не пусты и состоят только из цифр от 0 до 9.

Пример входных данных:

8 3 8

1234

239

366

261

32890

43823490

382390

3043840

Пример выходных данных:

11

32. В некоторых задачах защиты информации используются так называемые хеш-функции. Одним из важнейших классов хеш-функций являются так называемые полиномиальные хеш-функции. Пусть дана строка $S = s_1 s_2 \dots s_k$, состоящая из цифр от 0 до 9. Тогда значение полиномиальной хеш-функции $p(S, x, m)$ вычисляется следующим образом:

$$p(S, x, m) = \left(\sum_{i=1}^k s_i x^{i-1} \right) \bmod m$$

($a \bmod b$ обозначает остаток от деления числа a на число b). Например, пусть $S = 0123$, тогда $p(S, 2, 5) = (0 \cdot 1 + 1 \cdot 2 + 2 \cdot 4 + 3 \cdot 8) \bmod 5 = 4$.

Одним из способов применения хеш-функций является хранение паролей. Часто бывает так, что пароли приходится хранить в незащищенной таблице базы данных, поэтому вместо них хранят хеш-функции от них. При проверке пароля вычисляется хеш-функция от введенной строки и

сравнивается со значением, хранящимся в таблице. Ваша задача состоит в том, чтобы по заданным числам x , m , L и v найти строку S из цифр от 0 до 9 длины L , значение полиномиальной хеш-функции $p(S, x, m)$ равно v .

Входные данные - четыре целых числа: x (x – простое число, $5 \leq x \leq 100$), m (m является степенью двойки, $1 \leq m \leq 256$), L ($10 \leq L \leq 100$) и v ($0 \leq v \leq m-1$).

Выходные данные - выведите искомую строку или *NO SOLUTION*, если такой строки не существует. Если решений несколько, выведите любое.

Пример входных данных:

5 16 10 9

Пример выходных данных:

0422207956

33. Сторонники языков *Java* и *C++* часто спорят о том, какой язык лучше для решения олимпиадных задач. Одни говорят, что в *Java* есть масса полезных библиотек для работы со строками, хорошо реализованы механизмы чтения и вывода данных, а так же радует встроенные возможности для реализации длинной арифметики. С другой стороны, *C++* является классическим языком, скорость выполнения программ благодаря существующим компиляторам (например, *Intel Compiler 10.0*) гораздо выше, чем у *Java*. Но сейчас нас интересует лишь небольшие отличия, а именно соглашения, которыми пользуются программисты при описании имен переменных в *Java* и *C++*. Известно, что для понимания значений переменных часто используют английские слова или даже целые предложения, описывающие суть переменных, содержащих те или иные значения. Приведем ниже правила описания переменных, которыми руководствуются программисты, реализующие программы на *Java* и *C++*.

В языке *Java* принято первое слово, входящее в название переменной записывать с маленькой английской буквы, следующее слово идет с большой буквы (только первая буква слова большая), слова не имеют разделителей и состоят только из английских букв. Например, правильные записи

переменных в *Java* могут выглядеть следующим образом: *javaIdentifier*, *longAndMnemonicIdentifier*, *name*, *nEERC*.

В языке *C++* для описания переменных используются только маленькие английские символы и символ «_», который отделяет непустые слова друг от друга. Примеры: *java_identifier*, *long_and_mnemonic_identifier*, *name*, *n_e_e_r_c*.

Вам требуется написать программу, которая преобразует переменную, записанную на одном языке в формат другого языка.

Входные данные - наименование переменной длиной от 1 до 100 символов, в котором могут встречаться только заглавные и строчные буквы английского алфавита, а также символ подчёркивания.

Выходные данные - вывести аналог имени переменной в другом языке. Т.е. если переменная представлена в формате *Java*, то следует перевести в формат *C++* и наоборот. В том случае, когда имя переменной не соответствует ни одному из вышеописанных языков, следует вывести «*Error!*»

Пример входных данных:

long_and_mnemonic_identifier

Пример выходных данных:

longAndMnemonicIdentifier

34. На далекой планете Тау Кита есть непонятные нам обычаи. Например, таукитяне очень необычно для землян выбирают имена своим детям. Родители так выбирают имя ребенку, чтобы оно могло быть получено как удалением некоторого набора букв из имени отца, так и удалением некоторого набора букв из имени матери. Например, если отца зовут «*abaca*», а мать — «*bbssaa*», то их ребенок может носить имена «*a*», «*bba*», «*bca*», но не может носить имена «*aaa*», «*ab*» или «*bbc*». Возможно, что имя ребенка совпадает с именем отца и/или матери, если оно может быть получено из имени другого родителя удалением нескольких (возможно, ни одной) букв. Пусть отец по имени *X* и мать по имени *Y* выбирают имя своему новорожденному ребенку. Так как в таукитянских школах учеников часто

вызывают к доске в лексикографическом порядке имен учеников, то есть в порядке следования имен в словаре, то они хотят выбрать своему ребенку такое имя, чтобы оно лексикографически следовало как можно позже. Формально, строка S лексикографически больше строки T , если выполняется одно из двух условий:

- строка T получается из S удалением одной или более букв с конца строки S ;
- первые $(i - 1)$ символов строк T и S не различаются, а буква в i -й позиции строки T следует в алфавите раньше буквы в i -й позиции строки S .

Требуется написать программу, которая по именам отца и матери находит лексикографически наибольшее имя для их ребенка.

Входные данные - первая строка содержит имя отца X . Вторая строка входного файла содержит имя матери Y . Каждое имя состоит из строчных букв английского алфавита, включает хотя бы одну букву и имеет длину не более 10^5 букв.

Выходные данные - выведите искомое лексикографически наибольшее из возможных имен ребенка. В случае, если подходящего имени для ребенка не существует, ничего не выводите.

Пример входных данных:

ссба

ассббаа

Пример выходных данных:

ссба

35. Вам необходимо проверить домашнюю работу Васи Пупкина, в которой он написал равенство. Например, запись вида « $2+3=5$ » является правильной, а « $23*7=421$ » неверная, но корректная. Корректной записью выражения будем называть последовательность: число, операция («+», «-», «*», «/»), число, знак равенства, число. Числом будем считать последовательность из одной или более десятичных цифр, перед которой может стоять один знак минус. В корректной записи выражения нет пробелов.

Если запись не соответствует описанному правилу, то она считается некорректной. Например, записи « $2* = 3$ », «173» и « $2+2 = a$ » некорректны.

Входные данные - запись арифметического выражения, все числа в записи не превышают по абсолютной величине 30000. Длина арифметического выражения может быть от 0 до 100 символов.

Выходные данные - выведите «YES», если указанная запись правильна (т.е. равенство представляет собой тождество), «NO» - если корректная, но неверная и «ERROR», если запись некорректная.