# A. Полиномиальный хеш

Алле очень понравился алгоритм вычисления полиномиального хеша. Помогите ей написать функцию, вычисляющую хеш строки s. В данной задаче необходимо использовать в качестве значений отдельных символов их коды в таблице ASCII.

Полиномиальный хеш считается по формуле:

https://contest.yandex.ru/testsys/tex/render/aChzKSA9IChzXzFhXntuLTF9ICsgc18yYV57bi0yfSArIFxkb3RzICsgc197bi0xfWEgKyBzX3tufSkgXCBtb2QgXCBt.png

## Формат ввода

В первой строке дано число *a* (*1 ≤ a ≤ 1000*) –— основание, по которому считается хеш.

Во второй строке дано число *m* (*1 ≤ m ≤ 109*) –— модуль.

В третьей строке дана строка *s* (*0 ≤ |s| ≤ 106*), состоящая из больших и маленьких латинских букв.

## Формат вывода

Выведите целое неотрицательное число –— хеш заданной строки.

### Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 123  100003  a | 97 |

### Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 123  100003  hash | 6080 |

### Пример 3

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 123  100003  HaSH | 56156 |

# B. Сломай меня

Гоша написал программу, которая сравнивает строки исключительно по их хешам. Если хеш равен, то и строки равны. Тимофей увидел это безобразие и поручил вам сломать программу Гоши, чтобы остальным неповадно было.

В этой задаче вам надо будет лишь найти две различные строки, которые для заданной хеш-функции будут давать одинаковое значение.

Гоша использует следующую хеш-функцию:

https://contest.yandex.ru/testsys/tex/render/aChzKSA9IChzXzFhXntuLTF9ICsgc18yYV57bi0yfSArIC4uLiArIHNfe24tMX1hICsgc197bn0gKSBcICBtb2QgXCBt.png

для *a = 1000* и *m = 123 987 123*.

В данной задаче необходимо использовать в качестве значений отдельных символов их коды в таблице ASCII.

## Формат ввода

В задаче единственный тест без ввода

## Формат вывода

Отправьте две строки, по одной в строке. Строки могут состоять только из маленьких латинских букв и не должны превышать в длину 1000 знаков каждая. Код отправлять не требуется. Строки из примера использовать нельзя.

**Пример вывода:**

ezhgeljkablzwnvuwqvp

gbpdcvkumyfxillgnqrv

# C. Префиксные хеши

Алла не остановилась на достигнутом –— теперь она хочет научиться быстро вычислять хеши произвольных подстрок данной строки. Помогите ей!

На вход поступают запросы на подсчёт хешей разных подстрок. Ответ на каждый запрос должен выполняться за O(1). Допустимо в начале работы программы сделать предподсчёт для дальнейшей работы со строкой.

Напомним, что полиномиальный хеш считается по формуле

https://contest.yandex.ru/testsys/tex/render/aChzKSA9IChzXzFhXntuLTF9ICsgc18yYV57bi0yfSArIFxkb3RzICsgc197bi0xfWEgKyBzX3tufSkgXCBtb2QgXCBt.png

В данной задаче необходимо использовать в качестве значений отдельных символов их коды в таблице ASCII.

## Формат ввода

В первой строке дано число *a* (*1 ≤ a ≤ 1000*) –— основание, по которому считается хеш. Во второй строке дано число *m* (*1 ≤ m ≤ 107*) –— модуль. В третьей строке дана строка *s* (*0 ≤ |s| ≤ 106*), состоящая из больших и маленьких латинских букв.

В четвертой строке дано число запросов *t* –— натуральное число от *1* до *105*. В каждой из следующих t строк записаны через пробел два числа *l* и *r* –— индексы начала и конца очередной подстроки. (*1 ≤ l ≤ r ≤ |s|*).

## Формат вывода

Для каждого запроса выведите на отдельной строке хеш заданной в запросе подстроки.

### Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 1000  1000009  abcdefgh  7  1 1  1 5  2 3  3 4  4 4  1 8  5 8 | 97  225076  98099  99100  100  436420  193195 |

### Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 100  10  a  1  1 1 | 7 |

D. Кружки

В компании, где работает Тимофей, заботятся о досуге сотрудников и устраивают различные кружки по интересам. Когда кто-то записывается на занятие, в лог вносится название кружка.

По записям в логе составьте список всех кружков, в которые ходит хотя бы один человек.

Формат ввода

В первой строке даётся натуральное число *n*, не превосходящее *10 000* –— количество записей в логе.

В следующих *n* строках —– названия кружков.

Формат вывода

Выведите уникальные названия кружков по одному на строке, в порядке появления во входных данных.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 8  вышивание крестиком  рисование мелками на парте  настольный керлинг  настольный керлинг  кухня африканского племени ужасмай  тяжелая атлетика  таракановедение  таракановедение E. Подстроки На вход подается строка. Нужно определить длину наибольшей подстроки, которая не содержит повторяющиеся символы. Формат ввода Одна строка, состоящая из строчных латинских букв. Длина строки не превосходит *10 000*. Формат вывода Выведите натуральное число —– ответ на задачу. Пример 1  | **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод | | --- | --- | | abcabcbb | 3 |  Пример 2  | **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод | | --- | --- | | bbbbb | 1 | | вышивание крестиком  рисование мелками на парте  настольный керлинг  кухня африканского племени ужасмай  тяжелая атлетика  таракановедение |

# F. Анаграммная группировка

Вася решил избавиться от проблем с произношением и стать певцом. Он обратился за помощью к логопеду. Тот посоветовал Васе выполнять упражнение, которое называется анаграммная группировка. В качестве подготовительного этапа нужно выбрать из множества строк анаграммы.

Анаграммы –— это строки, которые получаются друг из друга перестановкой символов. Например, строки «SILENT» и «LISTEN» являются анаграммами.

Помогите Васе найти анаграммы.

## Формат ввода

В первой строке записано число *n* —– количество строк.

Далее в строку через пробел записаны *n* строк.

*n* не превосходит *6000*. Длина каждой строки не более *100* символов.

## Формат вывода

Нужно вывести в отсортированном порядке индексы строк, которые являются анаграммами.

Каждая группа индексов должна быть выведена в отдельной строке. Индексы внутри одной группы должны быть отсортированы по возрастанию. Группы между собой должны быть отсортированы по возрастанию первого индекса.

Обратите внимание, что группа анаграмм может состоять и из одной строки. Например, если в исходном наборе нет анаграмм, то надо вывести *n* групп, каждая из которых состоит из одного индекса.

## Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 6  tan eat tea ate nat bat | 0 4  1 2 3  5 |

# G. Соревнование

Жители Алгосов любят устраивать турниры по спортивному программированию. Все участники разбиваются на пары и соревнуются друг с другом. А потом два самых сильных программиста встречаются в финальной схватке, которая состоит из нескольких раундов. Если в очередном раунде выигрывает первый участник, в таблицу с результатами записывается *0*, если второй, то *1*. Ничьей в раунде быть не может.

Нужно определить наибольший по длине непрерывный отрезок раундов, по результатам которого суммарно получается ничья. Например, если дана последовательность *0 0 1 0 1 1 1 0 0 0*, то раунды с 2-го по 9-й (нумерация начинается с единицы) дают ничью.

## Формат ввода

В первой строке задаётся *n* (*0 ≤ n ≤ 105*) –— количество раундов. Во второй строке через пробел записано *n* чисел –— результаты раундов. Каждое число равно либо *0*, либо *1*.

## Формат вывода

Выведите длину найденного отрезка.

### Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 2  0 1 | 2 |

### Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3  0 1 0 | 2 |

# H. Странное сравнение

Жители Алгосского архипелага придумали новый способ сравнения строк. Две строки считаются равными, если символы одной из них можно заменить на символы другой так, что первая строка станет точной копией второй строки. При этом необходимо соблюдение двух условий:

* Порядок вхождения символов должен быть сохранён.
* Одинаковым символам первой строки должны соответствовать одинаковые символы второй строки. Разным символам —– разные.

Например, если строка s = «abacaba», то ей будет равна строка t = «xhxixhx», так как все вхождения «a» заменены на «x», «b» –— на «h», а «c» –— на «i». Если же первая строка s=«abc», а вторая t=«aaa», то строки уже не будут равны, так как разные буквы первой строки соответствуют одинаковым буквам второй.

## Формат ввода

В первой строке записана строка *s*, во второй –— строка *t*. Длины обеих строк не превосходят *106*. Обе строки содержат хотя бы по одному символу и состоят только из маленьких латинских букв.

Строки могут быть разной длины.

## Формат вывода

Выведите «YES», если строки равны (согласно вышеописанным правилам), и «NO» в ином случае.

### Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| mxyskaoghi  qodfrgmslc | YES |

### Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| agg  xdd | YES |

### Пример 3

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| agg  xda | NO |

# I. Общий подмассив

Гоша увлёкся хоккеем и часто смотрит трансляции матчей. Чтобы более-менее разумно оценивать силы команд, он сравнивает очки, набранные во всех матчах каждой командой.

Гоша попросил вас написать программу, которая по результатам игр двух выбранных команд найдёт наибольший по длине отрезок матчей, когда эти команды зарабатывали одинаковые очки.

**Рассмотрим первый пример:**

Результаты первой команды: *[1 2 3 2 1]*.

Результаты второй команды: *[3 2 1 5 6]*.

Наиболее продолжительный общий отрезок этих массивов имеет длину *3* –— это *[3 2 1]*.

## Формат ввода

В первой строке находится число *n* (*1 ≤ n ≤ 10000*) –— количество матчей, которые были сыграны первой командой.

Во второй строке записано *n* целых чисел –— очки в этих играх.

В третьей строке дано число *m* (*1 ≤ m ≤ 10000*) —– количество матчей, которые сыграла вторая команда.

В четвертой строке заданы *m* целых чисел —– результаты второй команды.

Число очков, заработанных в одной игре, лежит в диапазоне от *0* до *255*.

## Формат вывода

Выведите целое неотрицательное число —– максимальное количество матчей подряд, в которых команды зарабатывали одинаковые очки.

### Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5  1 2 3 2 1  5  3 2 1 5 6 | 3 |

### Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5  1 2 3 4 5  3  4 5 9 | 2 |

# J. Сумма четвёрок

У Гоши есть любимое число *S*. Помогите ему найти все уникальные четвёрки чисел в массиве, которые в сумме дают заданное число *S*.

## Формат ввода

В первой строке дано общее количество элементов массива *n* (*0 ≤ n ≤ 1000*).

Во второй строке дано целое число *S* https://contest.yandex.ru/testsys/tex/render/fFN8wqBcbGXCoDEwXjk=.png .

В третьей строке задан сам массив. Каждое число является целым и не превосходит по модулю *109*.

## Формат вывода

В первой строке выведите количество найденных четвёрок чисел.

В последующих строках выведите найденные четвёрки. Числа внутри одной четверки должны быть упорядочены по возрастанию. Между собой четвёрки упорядочены лексикографически.

### Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 8  10  2 3 2 4 1 10 3 0 | 3  0 3 3 4  1 2 3 4  2 2 3 3 |

### Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 6  0  1 0 -1 0 2 -2 | 3  -2 -1 1 2  -2 0 0 2  -1 0 0 1 |

### Пример 3

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5  4  1 1 1 1 1 | 1  1 1 1 1 |

# K. Ближайшая остановка

Гоша едет в гости к друзьям. Ему придётся сначала ехать на метро, а потом пересаживаться на автобус. Гоша не любит долго ждать, поэтому хочет выбрать такую станцию метро, рядом с которой расположено как можно больше остановок автобуса. Гоша считает, что остановка находится рядом с метро, если расстояние между ними не превосходит *20* метров.

Гоше известны все координаты автобусных остановок и координаты выходов из метро. Помогите ему найти выход из метро, рядом с которым расположено больше всего остановок.

Напомним, что расстояние между двумя точками с координатами *x1*, *y1* и *x2*, *y2* вычисляется по формуле https://contest.yandex.ru/testsys/tex/render/XHNxcnR7KHhfMSAtIHhfMileMiArICh5XzEgLSB5XzIpXjJ9.png.

**Пояснение к примеру 1:**

Рядом с 1-м выходом *(-1, 0)* находится только остановка с координатами *(10, 0)*.

Рядом со 2-м выходом *(1, 0)* находятся уже две остановки —– *(10, 0)* и *(20, 0)*.

Рядом с 3-м выходом *(2, 5)* расположены все три остановки –— *(22, 5)*, *(20, 0)* и *(10, 0)*

**Пояснение к примеру 2:**

Третий выход теперь находится в точке *(0, 5)*. Он рядом только с двумя остановками — *(20, 5)* и *(10, 0)*.

Рядом с 2-м и 3-м выходом одинаковое число остановок, поэтому в ответ идет 2-й выход, так как он раньше во входных данных.

## Формат ввода

В первой строке дано количество выходов из метро –— натуральное число *n* (*1 ≤ n ≤ 104*). В следующих *n* строках даны координаты выходов из метро. Каждый выход описывается двумя координатами *x* и *y*, записанными через пробел.

В следующей строке дано количество автобусных остановок —– натуральное число *m* (*1 ≤ m ≤ 106*). В следующих *m* строках заданы координаты остановок. Каждая остановка описывается двумя числами —– своими *x* и *y* координатами, записанными через пробел.

Все координаты —– целые числа, не превосходящие *109* по модулю. Единицы измерения —– метры.

## Формат вывода

Выведите единственное число –— номер того выхода из метро, рядом с которым расположено больше всего остановок. Номер выхода –— его порядковый номер во входных данных (нумерация с единицы).

Если подходящих выходов из метро несколько, выведите тот, который встречается раньше во входных данных.

### Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3  -1 0  1 0  2 5  3  10 0  20 0  22 5 | 3 |

### Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3  -1 0  1 0  0 5  3  10 0  20 0  20 5 | 2 |

# L. МногоГоша

Дана длинная строка, состоящая из маленьких латинских букв. Нужно найти все её подстроки длины *n*, которые встречаются хотя бы *k* раз.

## Формат ввода

В первой строчке через пробел записаны два натуральных числа *n* и *k*.

Во второй строчке записана строка, состоящая из маленьких латинских букв. Длина строки *1 ≤ L ≤ 106*.

*n ≤ L*, *k ≤ L*.

## Формат вывода

Для каждой найденной подстроки выведите индекс начала её первого вхождения (нумерация в строке начинается с нуля).

Выводите индексы в любом порядке, в одну строку, через пробел.

### Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 10 2  gggggooooogggggoooooogggggssshaa | 0 5 |

### Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3 4  allallallallalla | 0 1 2 |

# A. Поисковая система

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Язык | Ограничение времени | Ограничение памяти | Ввод | Вывод |
| Все языки | 4 секунды | 128Mb | стандартный ввод или input.txt | стандартный вывод или output.txt |
| Golang 1.14.4 + network | 6 секунд | 128Mb |
| Node.js 14.15.5 | 6.5 секунд | 400Mb |
| Python 3.7.3 | 6.5 секунд | 128Mb |
| gc go | 6 секунд | 128Mb |
| Mono C# 5.2.0 | 4 секунды | 400Mb |
| Oracle Java 8 | 4 секунды | 400Mb |
| OpenJDK Java 11 | 4 секунды | 400Mb |
| Node JS 8.16 | 6.5 секунд | 400Mb |

Тимофей пишет свою поисковую систему.

Имеется n документов, каждый из которых представляет собой текст из слов. По этим документам требуется построить поисковый индекс. На вход системе будут подаваться запросы. Запрос —– некоторый набор слов. По запросу надо вывести 5 самых релевантных документов.

Релевантность документа оценивается следующим образом: для каждого уникального слова из запроса берётся число его вхождений в документ, полученные числа для всех слов из запроса суммируются. Итоговая сумма и является релевантностью документа. Чем больше сумма, тем больше документ подходит под запрос.

Сортировка документов на выдаче производится по убыванию релевантности. Если релевантности документов совпадают —– то по возрастанию их порядковых номеров в базе (то есть во входных данных).

## Формат ввода

В первой строке дано натуральное число *n* —– количество документов в базе (*1 ≤ n ≤ 104*).

Далее в *n* строках даны документы по одному в строке. Каждый документ состоит из нескольких слов, слова отделяются друг от друга одним пробелом и состоят из маленьких латинских букв. Длина одного текста не превосходит *1000* символов. Текст не бывает пустым.

В следующей строке дано число запросов —– натуральное число *m* (*1 ≤ m ≤ 104*). В следующих *m* строках даны запросы по одному в строке. Каждый запрос состоит из одного или нескольких слов. Запрос не бывает пустым. Слова отделяются друг от друга одним пробелом и состоят из маленьких латинских букв. Число символов в запросе не превосходит *100*.

## Формат вывода

Для каждого запроса выведите на одной строке номера пяти самых релевантных документов. Если нашлось менее пяти документов, то выведите столько, сколько нашлось. Документы с релевантностью 0 выдавать не нужно.

### Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3  i love coffee  coffee with milk and sugar  free tea for everyone  3  i like black coffee without milk  everyone loves new year  mary likes black coffee without milk | 1 2  3  2 1 |

### Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 6  buy flat in moscow  rent flat in moscow  sell flat in moscow  want flat in moscow like crazy  clean flat in moscow on weekends  renovate flat in moscow  1  flat in moscow for crazy weekends | 4 5 1 2 |

# B. Хеш-таблица

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Язык | Ограничение времени | Ограничение памяти | Ввод | Вывод |
| Все языки | 5 секунд | 64Mb | стандартный ввод или input.txt | стандартный вывод или output.txt |
| Node.js 14.15.5 | 5 секунд | 256Mb |
| Python 3.7.3 | 15 секунд | 64Mb |
| Mono C# 5.2.0 | 5 секунд | 400Mb |
| Oracle Java 8 | 5 секунд | 400Mb |
| OpenJDK Java 11 | 5 секунд | 400Mb |
| Node JS 8.16 | 5 секунд | 256Mb |

Тимофей, как хороший руководитель, хранит информацию о зарплатах своих сотрудников в базе данных и постоянно её обновляет. Он поручил вам написать реализацию хеш-таблицы, чтобы хранить в ней базу данных с зарплатами сотрудников.

Хеш-таблица должна поддерживать следующие операции:

* put key value —– добавление пары ключ-значение. Если заданный ключ уже есть в таблице, то соответствующее ему значение обновляется.
* get key –— получение значения по ключу. Если ключа нет в таблице, то вывести «None». Иначе вывести найденное значение.
* delete key –— удаление ключа из таблицы. Если такого ключа нет, то вывести «None», иначе вывести хранимое по данному ключу значение и удалить ключ.

В таблице хранятся уникальные ключи.

Требования к реализации:

* Нельзя использовать имеющиеся в языках программирования реализации хеш-таблиц (std::unordered\_map в С++, dict в Python, HashMap в Java, и т. д.)
* Число хранимых в таблице ключей не превосходит *105*.
* Разрешать коллизии следует с помощью метода цепочек или с помощью открытой адресации.
* Все операции должны выполняться за *O(1)* в среднем.
* Поддерживать рехеширование и масштабирование хеш-таблицы не требуется.
* Ключи и значения, id сотрудников и их зарплата, —– целые числа. Поддерживать произвольные хешируемые типы не требуется.

## Формат ввода

В первой строке задано общее число запросов к таблице *n* (*1≤ n≤ 106*).

В следующих *n* строках записаны запросы, которые бывают трех видов –— get, put, delete —– как описано в условии.

Все ключи и значения –— целые неотрицательные числа, не превосходящие *109*.

## Формат вывода

На каждый запрос вида get и delete выведите ответ на него в отдельной строке.

### Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 10  get 1  put 1 10  put 2 4  get 1  get 2  delete 2  get 2  put 1 5  get 1  delete 2 | None  10  4  4  None  5  None |

### Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 8  get 9  delete 9  put 9 1  get 9  put 9 2  get 9  put 9 3  get 9 | None  None  1  2  3 |