

А. Граф подстрок

Ограничение времени	6 секунд
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Антон стажировается в группе обработки комментариев и отзывов. Для проверки гипотезы об автоматической генерации текстов Антон должен построить граф подстрок существующих текстов.

Антон берет все имеющиеся у него слова и действует следующим образом:

- слово $S = S_1S_2 \dots S_{n-1}S_n$ образует $n-2$ слова длины 3: $w_1 = S_1S_2S_3, w_2 = S_2S_3S_4, w_3 = S_3S_4S_5 \dots w_{n-2} = S_{n-2}S_{n-1}S_n$;
- если для какого-то из слов w_i еще нет вершины в графе, то она создается;
- для каждой пары слов (w_i, w_{i+1}) добавляется ориентированное ребро веса 1, или увеличивается вес существующего ребра на 1.

Таким образом получается граф G с v вершинами и e ориентированными ребрами. Между некоторыми вершинами может быть несколько дуг (от a к b и от b к a).

По заданному набору слов помогите Антону найти количество вершин в графе и вывести ориентированные ребра между вершинами.

Формат ввода

В первой строке записано одно целое число T ($1 \leq T \leq 40000$) — количество слов, имеющих у Антона.

В каждой из T следующих строк записано одно слово S_i ($4 \leq |S_i| \leq 30$). Все слова состоят из строчных букв английского алфавита.

Формат вывода

В первой строке выведите количество вершин v в графе G .

Во второй строке выведите количество пар вершин e , между которыми есть ориентированные ребра.

В каждой из следующих e строк выведите слово w_s , соответствующее началу ребра, затем слово w_f , соответствующее концу ребра, и количество ориентированных ребер из вершины w_s в w_f .

Ребра вы можете перечислить в произвольном порядке.

Пример 1

Ввод

2

aaaaaaaaaaaaa

aaabbbbaabbba

Вывод

6

7

aaa aaa 10

aaa aab 2

aab abb 2

abb bbb 2

bbb bba 2

bba baa 1

baa aaa 1

Пример 2

Ввод

2

abab

baba

Вывод

2

2

aba bab 1

bab aba 1

Пример 3

Ввод

1

qwertyqwertyqwertyqwerty

Вывод

6

6

qwe wer 5

wer ert 5

ert rty 5

rty tyq 4

tyq yqw 4

yqw qwe 4

В. Телефонные номера

Ограничение времени	3 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Есть база данных телефонных номеров. Необходимо для каждого номера определить страну, оператора, а также привести номер в определённый формат.

Существует список шаблонов, которым может удовлетворять номер. Он имеет вид: `NUMBER - COUNTRY OPERATOR`

- `NUMBER` – шаблон номера, формат ниже
- `COUNTRY` – названия страны, последовательности символов латинского алфавита
- `OPERATOR` – названия оператора, последовательности символов латинского алфавита и цифр

Номер в шаблоне задаётся в следующем виде: `+COUNTRY_CODE (OPERATOR_CODE) PERSONAL_NUMBER`

- `COUNTRY_CODE` – код страны, от одной до трёх цифр, первая цифра не может быть нулём
- `OPERATOR_CODE` – код оператора, от двух до четырёх цифр
- `PERSONAL_NUMBER` – шаблон номера абонента внутри оператора, строка длиной от пяти до девяти символов. Символом может быть цифра от 0 до 9 или символ `X`.

Символ `X` означает, что на данной позиции может быть любая цифра от 0 до 9. Справа от символа `X` не может стоять цифра

Все номера телефонов в текущей базе данных содержатся в полном формате (все цифры присутствуют), но, в отличие от формата выше:

- могут отсутствовать знак `+` и скобки
- могут отсутствовать или быть в любом месте номера пробелы и знак-разделитель дефис
- не содержат больше никакой информации (имени абонента, оператора и т.п.)

Гарантируется, что для каждого номера существует ровно один шаблон, которому он удовлетворяет, и все шаблоны не пересекаются.

Формат ввода

Первая строка содержит число N ($1 \leq N \leq 1000$) – количество номеров в базе данных.

Далее следует N строк – номера телефонов по одному номеру в строке. Длина строки не превосходит 100 символов.

Следующая строка содержит число M ($1 \leq M \leq 1000$) – количество шаблонов.

Далее M строк – шаблоны в формате, описанном выше. Длина шаблона не превосходит 100 символов.

Формат вывода

Выведите N строк, в каждой номер в новом формате в том порядке, в котором они указаны во входе.

Пример 1

Ввод

```
4
28-49-5-123-45-67
87544456789
+28 (495) 123 45 56
875-(29)-123456
3
+875 (29) 1XXXXXX - Atlantis
MythCell
+875 (44) 4XXXXXX - Atlantis
MobTelecom
+28 (495) XXXXXXXX - ElDorado
GoldLine
```

Вывод

```
+28 (495) 1234567 - ElDorado
GoldLine
+875 (44) 456789 - Atlantis
MobTelecom
+28 (495) 1234556 - ElDorado
GoldLine
+875 (29) 123456 - Atlantis
MythCell
```

Пример 2

Ввод

```
2
12345678
123456789
2
+1 (234) XXXXXX - USA SomeCell
+1 (23) XXXXXX - Canada AnotherCell
```

Вывод

```
+1 (23) 45678 - Canada AnotherCell
+1 (234) 56789 - USA SomeCell
```

Примечания

Некоторые страны могут иметь одинаковый COUNTRY_CODE.

C. RLE-сжатие

Ограничение времени	1 секунда
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

RLE-сжатие – один из самых простых методов сжатия строки, основанный на сокращении подстрок, состоящих из одинаковых символов. Сжатие осуществляется следующим образом:

- Строка разбивается на минимальное количество подстрок, состоящих из одинаковых символов. Например, `abbbcaaa` превращается в строки `a`, `bb`, `c`, `aaa`.
- Каждая из полученных строк превращается в строку, состоящую из числа и буквы. Числом является количество повторений символа в этой строке, буква берётся из первого символа обрабатываемой строки. Число не добавляется, если количество символов в строке равно единице. Из предыдущего массива строк мы получаем `a`, `2b`, `c`, `3a`.
- Затем полученные строки конкатенируются в исходном порядке. В рассмотренном примере в итоге получим `a2bc3a`.

Вам дана строка `s`, уже сжатая в RLE-формате. Назовём строку, из которой была получена `s`, строкой `t`. Вам даны `q` запросов, каждый из них представлен целыми числами `l` и `r`. В каждом запросе вам необходимо найти длину **сжатой** подстроки `t[l...r]`.

Формат ввода

В первой строке входного файла записана строка `s`, состоящая из строчных букв латинского алфавита и цифр ($1 \leq |s| \leq 1000000$). Гарантируется, что существует такая непустая строка `t`, из которой RLE-сжатием получается строка `s`. Также гарантируется, что в строке `t` не было больше 1000000000 одинаковых подряд идущих символов.

В следующей строке дано количество запросов `q` ($1 \leq q \leq 100000$). Каждая из следующих `q` строк содержит два числа `li` и `ri` ($1 \leq l_i \leq r_i \leq |t|$) — параметры запросов.

Формат вывода

Выведите `q` чисел, каждое в отдельной строке — ответы на запросы в том порядке, в котором запросы были заданы во входных данных.

Пример 1

Ввод

`a2bc3a`

5

Вывод

6

2

Ввод

1 7

5 7

1 2

3 5

4 4

Вывод

2

3

1

Пример 2

Ввод

x1000000000yz

3

2 1000000001

2 1000000002

5938493 15938493

Вывод

11

12

9

D. Любимые числа

Ограничение времени	1 секунда
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

У Саши есть три любимых числа: 5, 6, 10.

Кроме них, у Саши есть число N , не содержащее нулей. Он хочет сделать так, чтобы оно делилось хотя бы на одно из его любимых чисел. Для этого он K раз выполняет следующее: выбирает две цифры, стоящие на разных позициях, и меняет их местами.

Помогите Саше найти вероятность того, что итоговое число будет делиться хотя бы на одно из любимых чисел.

Формат ввода

В первой строке содержится единственное целое число ($10 \leq N \leq 10^{18}$), не содержащее нулей. Во второй строке содержится единственное целое число ($0 \leq K \leq 100$) – количество операций над числом N .

Формат вывода

В единственной строке выведите искомую вероятность. Абсолютная или относительная погрешность не должна превышать 10^{-9} .

Пример 1

Ввод

21

1

Вывод

1.0000000000000000

Пример 2

Ввод

145

2

Вывод

0.3333333333333333

Е. Произведение

Ограничение времени	1 секунда
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

У Васи есть массив A длины N из неотрицательных целых чисел и число M . Необходимо выбрать ровно K элементов массива A , чтобы их произведение было в точности равно M .

Формат ввода

Первая строка входного файла содержит три числа N, M, K ($1 \leq K \leq N \leq 5000, 0 \leq M \leq 10^9$) – размер массива A , произведение, которое нужно построить, и количество выбираемых элементов соответственно.

Вторая строка входного файла содержит N неотрицательных целых чисел $A_i (0 \leq A_i \leq 10^9)$ – элементы массива A .

Гарантируется, что ответ всегда существует.

Формат вывода

Выведите K различных натуральных чисел – индексы выбранных элементов массива A . Если решений несколько, выведите любое. Индексы можно выводить в произвольном порядке.

Пример 1

Ввод

7 27 2
9 1 1 27 3 27 3

Вывод

4 2

Пример 2

Ввод

7 60 4
30 1 1 3 10 6 4

Вывод

5 6 3 2

Г. Квадрат и окружность

Ограничение времени	1 секунда
Ограничение памяти	512Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

У Васи есть квадрат, вершины которого расположены в точках с координатами $(0,0)$, $(1,0)$, $(0,1)$, $(1,1)$. В этом квадрате расположены N фишек, i -я фишка имеет координаты (x_i, y_i) . Фишки имеют пренебрежительно малые размеры, будем считать их материальными точками.

Вася играет в игру. Он выбирает случайную точку с равномерным распределением в своем квадрате. Потом Вася подсчитывает количество фишек, расстояние от которых до выбранной точки не превышает R , и получает соответствующее количество очков.

Помогите Васе узнать математическое ожидание количества очков, которые он наберет в описанной выше игре.

Формат ввода

В первой строке входных данных содержатся два числа N и R ($1 \leq N \leq 1000$, $0.001 \leq R \leq 2$). Каждая из следующих N строк содержит разделенные пробелом координаты очередной фишки x_i и y_i ($0.001 \leq x_i, y_i \leq 0.999$).

Все числа во входных данных содержат не более 3 знаков после десятичной точки.

Формат вывода

Выведите единственное число – матожидание количества очков.

Ответ будет считаться верным, если его относительная или абсолютная погрешность не превосходит 10^{-9} .

Пример 1

Ввод

```
1 0.5
0.5 0.5
```

Вывод

```
0.7853981634
```

Пример 2

Ввод

```
2 2
0.001 0.001
0.999 0.999
```

Вывод

```
2.0000000000
```