

Технокубок 2019 - Отборочный Раунд 3

А. Столовые приборы

1 секунда, 256 мегабайт

На ужин по случаю дня рождения короля пришло k гостей. Ужин удался на славу: каждый из гостей отведал некоторое (одинаковое для всех) количество блюд, а к каждому блюду каждому гостю подавали новый набор столовых приборов.

Все виды столовых приборов в стране короля пронумерованы от 1 до 100. Известно, что все наборы приборов одинаковые и состоят из некоторого количества приборов различного вида, прибор одного вида может входить в набор не более одного раза. Например, набор может состоять из одной вилки, одной ложки и одного ножа.

После ужина королю стало интересно, какое минимальное количество его столовых приборов было украдено гостями. К сожалению, король забыл, сколько подавалось блюд, зато он знает список столовых приборов, которые остались после ужина. Помогите ему определить минимальное количество украденных приборов.

Входные данные

Первая строка содержит два целых числа n и k ($1 \leq n \leq 100, 1 \leq k \leq 100$) — количество столовых приборов, оставшихся после ужина, и количество гостей, которые пришли к нему на ужин.

Следующая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 100$) — виды столовых приборов в списке. Одинаковыми числами обозначены одинаковые столовые приборы, а разными — разные.

Выходные данные

Выведите единственное число — минимальное количество столовых приборов, которые были украдены гостями короля.

Входные данные

```
5 2
1 2 2 1 3
```

Выходные данные

```
1
```

Входные данные

```
10 3
1 3 3 1 3 5 5 5 5 100
```

Выходные данные

```
14
```

В первом примере ясно, что хотя бы один прибор типа 3 украден, так как всего два гостя, а остался лишь один такой прибор. В то же время возможно такое, что подавали лишь одно блюдо и всего приборов было шесть: по набору (1, 2, 3) каждому из гостей. Значит, ответ равно 1.

Во втором примере можно показать, что должно было быть подано хотя бы 2 блюда, а значит приборов всего должно быть хотя бы 24: по 4 в каждом наборе, по два набора каждому из 3 людей. Значит, всего украли не меньше 14 предметов. Обратите внимание, приборы некоторых типов (например, типа 2 и 4 в этом примере) могли вообще не подаваться на стол.

В. Именной кубок

1 секунда, 256 мегабайт

На многих соревнованиях, которые содержат в своем названии слово «кубок», победителю торжественно вручают собственно кубок. В этот раз организаторы одного необычного соревнования по программированию решили еще больше порадовать победителя и добавить на кубок табличку с хэндлом победителя.

Табличка будет прямоугольная, а текст на ней будет напечатан в виде таблицы из нескольких строк и столбцов. Измерив кубок, организаторы выяснили, что число строк a не может превосходить 5, а число столбцов b не может быть больше 20. В любой клетке таблицы организаторы хотят написать либо звездочку («*»), либо букву из хэндла победителя.

Кроме того, организаторы хотят, чтобы строки таблицы были равномерными, то есть разница между количеством звездочек в разных строках была не более, чем один (то есть, например, нельзя в первой строке написать две звездочки, а во второй — ни одной). И самое главное, если прочитать табличку в порядке сверху вниз, в каждой строке слева направо, пропуская звездочки, должен получаться в точности хэндл победителя.

Организаторы хотят, чтобы табличка содержала как можно меньше строк, а из всех возможных таблиц с минимальным числом строк они хотят сделать такую, которая содержит как можно меньше столбцов.

Победитель еще не определен, поэтому ваша задача — написать программу, которая по заданному хэндлу победителя сгенерирует необходимую табличку.

Входные данные

Единственная строка содержит одну строку s ($1 \leq |s| \leq 100$), состоящую только из заглавных и строчных букв английского алфавита, — хэндл победителя.

Выходные данные

В первой строке выведите минимальное количество строк в табличке a , и минимальное число столбцов в ней, при таком количестве строк, b .

В следующих a строках выведите по b символов — табличку, соответствующую условию задачи.

ВХОДНЫЕ ДАННЫЕ
tourist
ВЫХОДНЫЕ ДАННЫЕ
1 7 tourist

ВХОДНЫЕ ДАННЫЕ
MyNameIsLifeIAmForeverByYourSideMyNameIsLife
ВЫХОДНЫЕ ДАННЫЕ
3 15 MyNameIsLifeIAm ForeverByYourSi deMyNameIsL*ife

С. Игра на фортепиано

1 секунда, 256 мегабайт

Маленький Пол хочет научиться играть на фортепиано. Он уже нашёл мелодию, которую будет играть. Для простоты он выписал последовательность a_1, a_2, \dots, a_n целых чисел, которые означают номер клавиш: чем больше номер, тем правее клавиша на клавиатуре.

Пол очень умный и понимает, что самое важное — правильная аппликатура, то есть, правильно выбрать, каким пальцем какую ноту играть. Если выбрать неудобные пальцы, то потом можно потратить уйму времени на попытки научиться играть мелодию и всё равно в итоге не преуспеть.

Обозначим пальцы на руке числами от 1 до 5. Назовём *аппликатурой* любую последовательность b_1, \dots, b_n номеров пальцев. Назовём аппликатуру *удобной*, если для любого $1 \leq i \leq n - 1$ выполнено следующее:

- если $a_i < a_{i+1}$, то $b_i < b_{i+1}$, потому что иначе придётся оторвать руку от клавиатуры, чтобы сыграть $(i + 1)$ -ю ноту;
- если $a_i > a_{i+1}$, то $b_i > b_{i+1}$ по той же причине;
- если $a_i = a_{i+1}$, то $b_i \neq b_{i+1}$, потому что довольно нелепо использовать один и тот же палец два раза подряд. **Обратите внимание, что между b_i и b_{i+1} стоит знак \neq , а не $=$.**

Найдите любую удобную аппликатуру или скажите, что таких нет.

Входные данные

Первая строка содержит одно целое число n ($1 \leq n \leq 10^5$), обозначающее число нот в мелодии.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 2 \cdot 10^5$) — номера этих нот.

Выходные данные

Если удобных аппликатур к этой мелодии не существует, выведите -1 . В противном случае выведите n целых чисел b_1, b_2, \dots, b_n от 1 до 5, задающих удобную аппликатуру.

входные данные
5 1 1 4 2 2
выходные данные
1 4 5 4 5

входные данные
7 1 5 7 8 10 3 1
выходные данные
1 2 3 4 5 4 3

входные данные
19 3 3 7 9 8 8 8 8 7 7 7 7 5 3 3 3 3 8 8

выходные данные
1 3 4 5 4 5 4 5 4 5 4 5 4 3 5 4 3 5 4

Третий тест из условия — что-то вроде песни "Нон стоп" группы Рефлекс.

D. Барселонское расстояние

1 секунда, 256 мегабайт

В этой задаче мы рассматриваем упрощённую модель города Барселона.

Барселона может быть представлена как плоскость, а её улицы — как прямые вида $x = c$ и $y = c$ для любого целого числа c (то есть, улицы представляют собой прямоугольную сетку). Однако есть кое-что, что отличает Барселону от Манхэттена, а именно, проспект, который называется Avinguda Diagonal (Диагональный проспект) и который может быть представлен на плоскости как множество точек (x, y) , удовлетворяющих $ax + by + c = 0$.

Перемещаться по городу можно только по улицам, в том числе и по Диагональному проспекту. Вам даны точки A и B в Барселоне. Найдите минимальное расстояние, которое необходимо пройти, чтобы попасть в точку B из точки A .

Входные данные

Первая строка содержит три целых числа a, b и c ($-10^9 \leq a, b, c \leq 10^9$, хотя бы одно из чисел a и b не равно нулю), задающих Диагональный проспект.

Следующая строчка содержит четыре целых числа x_1, y_1, x_2 и y_2 ($-10^9 \leq x_1, y_1, x_2, y_2 \leq 10^9$), задающих точки $A = (x_1, y_1)$ и $B = (x_2, y_2)$.

Выходные данные

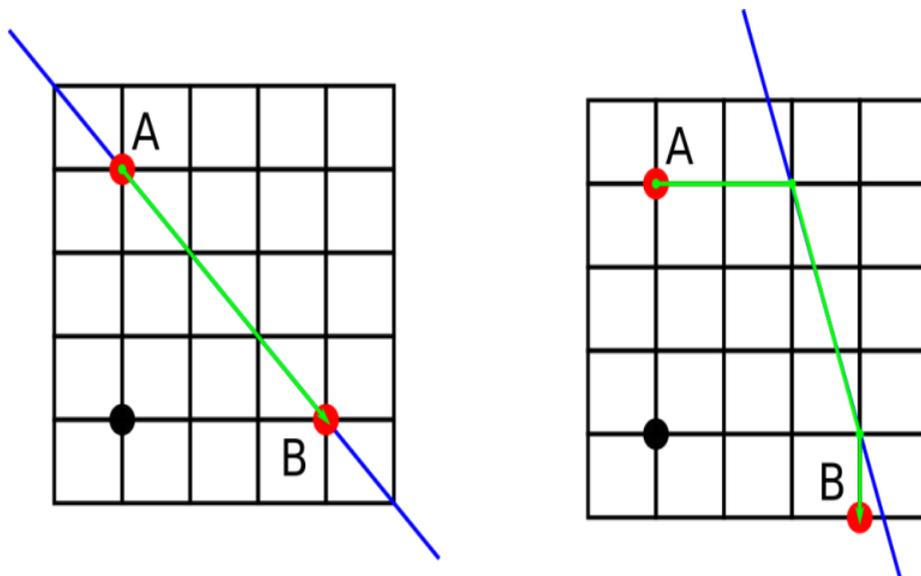
Найдите минимальное расстояние, которое необходимо пройти от A до B . Ваш ответ будет считаться правильным, если его абсолютная или относительная ошибка не превосходит 10^{-6} .

Формально, пусть ваш ответ равен a , а ответ жюри равен b . Ваш ответ будет зачтен, если и только если $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$.

Входные данные
1 1 -3 0 3 3 0
Выходные данные
4.2426406871

Входные данные
3 1 -9 0 3 3 -1
Выходные данные
6.1622776602

Первый пример показан на рисунке ниже слева, второй — справа. Проспект показан синим, начало координат отмечено черной точкой.



Е. Невыносимая лёгкость гирек

1 секунда, 256 мегабайт

У вас есть набор из n гирек. Вам известно, что они весят a_1, a_2, \dots, a_n граммов, но вы не знаете, какая гиричка сколько весит: для вас они неотличимы друг от друга.

Ваш друг знает точный вес каждой гири, и вы можете попросить его дать вам набор из k гирек с суммарным весом m (параметры k и m выбираете вы сами), и ваш друг укажет вам на какой-нибудь подходящий набор гирек, если такой вообще есть.

Вы можете спросить вашего друга лишь один раз. У какого наибольшего числа гирек вы можете узнать точный вес после этого запроса?

Входные данные

Первая строка содержит одно целое число n ($1 \leq n \leq 100$) — количество гирек.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 100$) — веса гирек.

Выходные данные

Выведите единственное число — максимальное количество гирек, вес которых вы можете узнать, один раз спросив друга.

Входные данные
4 1 4 2 2
Выходные данные
2
Входные данные
6 1 2 4 4 4 9

ВЫХОДНЫЕ ДАННЫЕ

2

В первом примере мы можем попросить набор из двух гирек с суммарным весом 4, и единственное, что мы можем получить — $\{2, 2\}$.

Получить такой же результат можно, попросив набор из двух гирек с суммарным весом 5, тогда нам дадут $\{1, 4\}$. Отсюда мы можем заключить, что каждая из двух оставшихся гирь имеет вес 2.

Во втором примере мы можем попросить набор из двух гирек с суммарным весом 8 и получить $\{4, 4\}$. Можно доказать, что нельзя за один запрос узнать веса трех гирек, но мы не будем приводить доказательство здесь.

Г. Вася и максимальное паросочетание

3 секунды, 256 мегабайт

У Васи есть дерево из n вершин. Он хочет знать количество способов удалить некоторое количество ребер (возможно, ни одного) из дерева так, чтоб в оставшемся графе максимальное паросочетание было единственно.

Паросочетанием в графе называется подмножество ребер такое, что все концы этих ребер различны. Максимальным паросочетанием называется такое паросочетание, в котором количество ребер максимально возможное.

Так как количество способов может быть велико, выведите его по модулю 998244353.

Входные данные

Первая строка содержит целое число n ($1 \leq n \leq 3 \cdot 10^5$) — количество вершин дерева.

Следующие $n - 1$ строк содержат по два числа u и v ($1 \leq u, v \leq n, u \neq v$), представляющие ребро между вершинами u и v . Гарантируется, что данные ребра составляют дерево.

Выходные данные

В единственной строке выведите число — количество способов удалить ребра в дереве так, чтобы в оставшемся графе максимальное паросочетание было единственно. Выведите ответ по модулю 998244353.

ВХОДНЫЕ ДАННЫЕ

4
1 2
1 3
1 4

ВЫХОДНЫЕ ДАННЫЕ

4

ВХОДНЫЕ ДАННЫЕ

4
1 2
2 3
3 4

ВЫХОДНЫЕ ДАННЫЕ

6

ВХОДНЫЕ ДАННЫЕ

1

ВЫХОДНЫЕ ДАННЫЕ

1

В первом тестовом примере максимальное паросочетание будет единственно в четырех случаях:

- если удалить ребра $(1, 2)$ и $(1, 3)$.
- если удалить ребра $(1, 2)$ и $(1, 4)$.
- если удалить ребра $(1, 3)$ и $(1, 4)$.
- если удалить все ребра.

Во втором тестовом примере максимальное паросочетание будет единственно в шести случаях:

- если не удалять ни одного ребра.
- если удалить ребра (1, 2) и (2, 3).
- если удалить ребра (1, 2) и (3, 4).
- если удалить ребра (2, 3) и (3, 4).
- если удалить ребро (2, 3).
- если удалить все ребра.

Г. Болтовня

2 секунды, 256 мегабайт

n попугаев стоят по кругу. У каждого попугая есть некоторый уровень уважения среди остальных попугаев, обозначаемый r_i . Если попугай с уровнем уважения x начинает болтовню, то через одну секунду его x соседей слева и справа начинают повторять за ним. Естественно, слыша болтовню, их соседи также включаются в процесс, и так далее, пока все попугаи не начнут болтать.

Вам даны уровни уважения всех попугаев. Для каждого попугая независимо ответьте на вопрос: если данный попугай начнёт болтать, через сколько секунд все остальные попугаи начнут повторять за ним?

Входные данные

В первой строке задано целое число n — количество попугаев ($1 \leq n \leq 10^5$).

В следующей строке задано n целых чисел r_1, \dots, r_n — уровни уважения попугаев в том порядке, в котором они стоят в кругу ($1 \leq r_i \leq n$).

Выходные данные

Выведите n чисел. i -е из них должно равняться количеству секунд, которое пройдёт между тем, как i -й попугай начнёт болтать, и тем, когда заболтают все попугаи в кругу.

входные данные
4 1 1 4 1
выходные данные
2 2 1 2

входные данные
8 1 2 2 1 5 1 3 1
выходные данные
3 3 2 2 1 2 2 3