Эффективная и слаженная
командная работа — залог успеха!

XXII РЕСПУБЛИКАНСКАЯ КОМАНДНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ ПО ПРОГРАММИРОВАНИЮ

Якутск
6 апреля 2025 г.

XXII Республиканская командная олимпиада школьников по программированию. — Якутск, 2025.

Сборник содержит условия задач XXII Республиканской командной олимпиады школьников по программированию и возможные варианты решений. Олимпиада проводилась 6 апреля 2025 г. на базе Института математики и информатики СВФУ им. М.К. Аммосова при участии Малой академии наук РС (Я) в г. Якутск. Участникам было предложено за пять часов решить двенадцать задач.

СПОНСОРЫ ОЛИМПИАДЫ

ГЕНЕРАЛЬНЫЙ СПОНСОР
ООО «Аксиома»
Директор
Наталья Леонтьевна Махонина

ООО «КопирТехСервис»
Генеральный директор
Александр Тарасович Никифоров

ООО «Саха Бизнес Автоматизация»
Генеральный директор
Дмитрий Иванович Авксентьев

ОРГКОМИТЕТ ОЛИМПИАДЫ

ГОЛИКОВ Алексей Иннокентьевич
проректор по учебно-методической работе
ФГАОУ ВО «Северо-Восточный федеральный университет
имени М. К. Аммосова» — председатель

ПАВЛОВ Василий Климович
ректор ГАНОУ РЦ РС (Я) «Малая академия наук РС (Я)» —
заместитель председателя

НИКОЛАЕВА Наталья Васильевна
зав. кафедрой «Информационные технологии» ИМИ СВФУ,
тьютор информатики ГАНОУ РЦ РС (Я) «Малая академия наук РС (Я)»

СЕМЕНОВА Галина Александровна
проректор по учебно-организационной и воспитательной работе
ГАНОУ РЦ РС (Я) «Малая академия наук РС (Я)»

КУЛИЧКИНА Инна Владимировна
диспетчер образовательного учреждения
ГАНОУ РЦ РС (Я) «Малая академия наук РС (Я)»

ЖЮРИ ОЛИМПИАДЫ

НИКОЛАЕВА Наталья Васильевна
к. ф.-м. н., зав. кафедрой «Информационные технологии» ИМИ СВФУ, тьютор информатики ГАНОУ РЦ РС (Я) «Малая академия наук РС (Я)» — председатель

ПАВЛОВ Никифор Никитич
к. ф.-м. н., доцент кафедры «Информационные технологии» ИМИ СВФУ

НИКИФОРОВ Дьулустан Васильевич
старший преподаватель кафедры «Информационные технологии» ИМИ СВФУ

ЛЕВЕРЬЕВ Владимир Семенович
старший преподаватель кафедры «Информационные технологии» ИМИ СВФУ

БУЛАТОВ Василий Алквиадович
старший разработчик Nebius

ДМИТРИЕВ Николай Николаевич
студент ИМИ СВФУ

КУЛИЧКИН Тимур Федорович
студент ИМИ СВФУ

ПАРНИКОВ Василий Васильевич
студент НИУ ВШЭ

КУЛИЧКИНА Инна Владимировна
студент ИМИ СВФУ, диспетчер образовательного учреждения ГАНОУ РЦ РС (Я) «Малая академия наук РС (Я)»— секретарь

СПИСОК УЧАСТНИКОВ

11 Тестировщики Java *(11 кл. ЯГЛ, руководитель Жиркова М. М.)*
Афанасьев Федор Алексеевич, Иванов Дмитрий Максимович,
Исаев Андрей Васильевич

BUG *(Бердигестяхская улусная гимназия им. В. В. Филиппова, рук. Сидорова Ф. Л.)*
Антонов Александр Дьулустанович *9 кл.*,
Михайлов Айсен Михайлович *11 кл.*, Старостин Василий Юрьевич *11 кл.*

Csmteam *(11 кл., Хатасская СОШ им. П. Н. и Н. Е. Самсоновых, рук. Петров П. П.)*
Абрамов Артем Александрович, Горохов Айсен Викторович,
Попов Максим Андреевич

Ctrl+Alt+Elite *(ИТЛ №24 г. Нерюнгри им. Е. А. Варшавского, рук. Богданов Р. А.)*
Гурьев Артур Альбертович *11 кл.*, Неграшев Денис Романович *9 кл.*,
Неживых Дмитрий Андреевич *11 кл.*

enaije’es *(11 кл. РЛИ, руководитель Уваровская М. И.)*
Федоров Альберт Викторович, Каратаев Святослав Семенович,
Попов Кирилл Николаевич

Eternity Gaming *(10 кл. Октемский НОЦ им. М. Е. Николаева, рук. Ковров Ф. Ф.)*
Афанасьев Леонид Владиславович, Никитин Егор Юрьевич,
Павлов Артём Дмитриевич

eng54 *(10 кл. РЛИ, руководитель Уваровская М. И.)*
Дормидонтов Ярослав Владимирович, Припузов Сулустаан Денисович,
Романов Альберт Сергеевич

Junior *(9 кл. ЯГЛ, руководитель Жиркова М. М.)*
Жирков Ярослав Константинович, Мункуев Эрдэм Амирзаяаевич,
Попов Эльдар Леонидович

KaaKii *(11 кл. РЛИ, руководитель Уваровская М. И.)*
Колодезников Андрей Аркадьевич, Коркин Иннокентий Иосифович,
Ксенофонтов Артём Александрович

MVP *(11 кл. Октемский НОЦ им. М. Е. Николаева, руководитель Ковров Ф. Ф.)*
Карпов Максим Григорьевич, Кузьмин Байдам Андреевич,
Федоров Кирилл Владимирович

АТА-56 *(8 кл. РЛИ, руководитель Николаева Н. В.)*
Илларионов Анатолий Анатольевич, Саввинов Айаан Юрьевич,
Чупров Артём Сергеевич

Бо Сины *(СПЛ, руководители Саввинов И. С. и Фролова С. М.)*
Корякин Андрей Леонидович *9 кл.*, Лаптев Данияр Антонович *8 кл.*,
Припузов Дархан Денисович *8 кл.*

Дети Макарова *(Чурапчинская гимназия им. С. К. Макарова, рук. Захаров П. П.)*
Дьяконов Алексей Александрович *7 кл.*, Петров Виктор Кузьмич *8 кл.*,
Сивцев Аман Прокопьевич *7 кл.*

ЗФЯ *(СПЛ, руководители Саввинов И. С. и Фролова С. М.)*
Арьянов Янис Алексеевич *11 кл.*, Захаров Нюргун Семенович *10 кл.*,
Фёдоров Артём Афанасьевич *11 кл.*

КыыStar *(10 кл. РЛИ, руководитель Уваровская М. И.)*
Афанасьева Надежда Юрьевна, Винокурова Светлана Алексеевна,
Канаева Рената Артемовна

Название команды *(11 кл. РЛИ, руководитель Уваровская М. И.)*
Малышев Виктор Валерьевич, Неустроева Нарыйаана Иннокентьевна,
Решетников Ньургун Петрович

Саайса *(8 кл. Томпонская многопрофильная гимназия им. В. А. Штырова, руководитель Рогачева Е. Г.)*
Илларионов Саян Васильевич, Новгородов Айсен Петрович,
Роев Айтал Нюргунович

Сигмы *(9 кл. Айыы Кыһата, руководители Халабышева Е. В. и Готовцева Е. Д.)*
Семенова Майя Ньургуновна, Слободчиков Петр Павлович,
Тарский Иван Евгеньевич

СУНЦ-1 *(10 кл. СУНЦ СВФУ, руководитель Фролова С. М.)*
Евстифеев Игорь, Трофимова Диана, Халланова Айна

СУНЦ-2 *(10 кл. СУНЦ СВФУ, руководитель Фролова С. М.)*
Голиков Александр Алексеевич,
Абрамов Айдамир, *10 класс*
Козьмин Виталий, *10 класс*

Фломастеры *(8 кл. СПЛ, рук. Саввинов И. С. и Фролова С. М.)*
Егоров Иван Иванович, Козлов Ньургун Степанович,
Назаров Руслан Ильич

ФСБ *(8 кл. РЛИ, руководитель Николаева Н. В.)*
Большакова Амелия Егоровна, Слепцова Карина Гаврииловна,
Федорова Виктория Владиславовна

ФТЛ-7 *(7 кл. ФТЛ, руководитель Романов Ю. Н.)*
Местников Тимур, Григорьев Эрчим, Захаров Вячеслав

ФТЛ-8 *(8 кл. ФТЛ, рук. Куличкин Н. Н.)*
Васильева Ника Сарыаловна, Елизаров Василий Павлович,
Платонов Егор

Ханалас 1 *руководитель Егоров В. Д.*
Беца Максим Алексеевич *8 кл. Покровская СОШ №3-ОЦ с УИОП*,
Марков Артур Гаврильевич *9 кл. Покровская СОШ №1 им. И. М. Яковлева*,
Немечкин Артем Владимирович *9 кл. Покровская СОШ №3-ОЦ с УИОП*

Ханалас 2 *(Покровская СОШ №2 с УИОП, руководитель Мордовской Д. А.)*
Боярский Дмитрий Владимирович *8 кл.*,
Латышев Айгылаан Николаевич *7 кл.*,
Федоров Алексей Васильевич *9 кл.*

ЧИН chinA *(10 кл. РЛИ, руководитель Уваровская М. И.)*
Иевлев Арылхан Маркович, Николаев Тимофей Дмитриевич,
Чашкин Георгий Фомич

Чурапча
Александров Николай Васильевич *10 кл.*
(Чурапчинская СОШ им. С. А. Новгородова, руководитель Прокопьев Е. В.),
Куличкина Кристина Федоровна *10 кл.*, Павлов Артем Айаалович *9 кл.*
(Чурапчинская гимназия им. С. К. Макарова, руководитель Захаров П. П.)

Эрэл54 *(10 кл. РЛИ, руководитель Уваровская М. И.)*
Артемьев Георгий Иванович, Лыткин Марк Сергеевич,
Юмшанов Эрчим Петрович

Эрэл55 *(9 кл. РЛИ, руководитель Уваровская М. И.)*
Потапов Долан Васильевич, Сивцев Арсентий Андреевич,
Татаринов Тимур Иванович

Эрэл57-1 *(7 кл. РЛИ, руководитель Андреева Д. Д.)*
Колескин Артур Валерьевич, Уваровский Андрей Александрович,
Седалищев Михаил Петрович

Эрэл57-2 *(7 кл. РЛИ, руководитель Андреева Д. Д.)*
Васильев Бэргэн Егорович, Васильев Гансар Гаврильевич,
Васильев Дархан Валерьевич

ЯГЛ *(10 кл. ЯГЛ, руководитель Жиркова М. М.)*
Лиханов Владимир Павлович, Серкин Алексей Александрович,
Христофоров Александр Дмитриевич

ЯГНГ 10 *(10 кл. ЯГНГ, руководители Спиридонов Я. Я. и Жерготов П. П.)*
Данилов Сандал Фёдорович, Евсеева Алина Владимировна,
Егоров Алексей Иванович

ЯГНГ 11 *(11 кл. ЯГНГ, руководители Спиридонов Я. Я. и Жерготов П. П.)*
Комиссаров Марат Юрьевич, Ксенофонтов Гектор Тимофеевич
Петухов Тимур Иннокентьевич

УСЛОВИЯ ЗАДАЧ

A. Мы обожаем деревья

Мы обожаем деревья! Но больше всего их обожает Алёша. Недавно он придумал игру на дереве, которой он очень сильно хочет поделиться!

Дано дерево (связный граф без циклов) с n вершинами и корнем в вершине 1. У игрока имеется фишка, с помощью которой он делает ходы. Игру можно описать следующим образом:

1. Сначала игрок выбирает любую вершину в качестве начальной и ставит на неё фишку.
2. За один ход игрок может переместить фишку либо к родителю, либо к родителю родителя. При этом имеются две особенные вершины u и v: в вершине u имеется портал, который дополнительно может телепортировать фишку в вершину v. После использования портала он ломается, то есть портал можно использовать в игре максимум 1 раз.
3. Ход фишки в i-ю вершину изменяет очки игрока на величину a_i. Если вершина была посещена повторно, то a_i снова прибавляется к очкам. Начальная вершина тоже сразу же, как на неё поставлена фишка, приносит очки.
4. Игра заканчивается, когда игрок доходит до корня.

Алёша решил сделать m запросов, которые характеризуются вершинами u и v. Ответом на каждый запрос является максимальное количество очков, которое игрок может набрать, если портал стоит в вершине u и ведёт в вершину v. Требуется вывести ответ на каждый запрос.

Формат входных данных

Первая строка содержит два целых числа n и m ($3 \leqslant n \leqslant 10^5; 1 \leqslant m \leqslant 10^5$) — количество вершин и запросов соответственно.

Вторая строка содержит $n-1$ целых чисел $p_2, p_3, \ldots, p_n$ ($1 \leqslant p_i \leqslant n$) — каждое p_i является родителем вершины с номером i.

Третья строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($-10^4 \leqslant a_i \leqslant 10^4$) — значения очков i-й вершины.

Затем следуют m строк, каждая из которых содержит два целых числа u и v ($1 \leqslant u, v \leqslant n; u \neq v; u \neq 1$) — местоположение портала и куда он ведёт соответственно.

Формат выходных данных

Построчно для каждого запроса выведите ответ (в m строках).

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
6 5 1 2 2 4 2 -2 -3 -5 5 3 -6 2 4 5 4 2 3 4 1 5 6	8 6 6 6 6

Комментарии

Рассмотрим первый запрос. Траектория движения следующая: $5 \to 4 \to 2 \to$ (телепорт) $\to 4 \to 1$.

B. Обратная конкатенация

Тимуру даётся массив a длины n, состоящий из целых двузначных чисел. За одну операцию он может сделать следующее:

- Выбрать какие-то два числа с индексами i и j ($1 \leqslant i < j \leqslant n$);
- Удалить a_i из массива, а a_j заменить на число вида $a_j a_i$.

Иными словами, к правому числу дописывается левое, затем левое число удаляется. Например, если $a_i = 23$, а $a_j = 31$, то после операции a_i удаляется, а $a_j = 3123$. Очевидно, что после $n - 1$ операций останется лишь одно число.

Определите максимальное число, которое Тимур может получить в качестве оставшегося.

Формат входных данных

Первая строка содержит целое число n ($1 \leqslant n \leqslant 10^5$) — количество элементов в массиве.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($10 \leqslant a_i \leqslant 99$).

Формат выходных данных

Выведите максимальное число, которое можно получить в качестве последнего.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
1 53	53
2 78 42	4278
3 19 75 32	327519
10 11 11 11 11 11 11 11 11 11 11	11111111111111111111

C. Скобки

Айтал и Григорий — школьники и настоящие энтузиасты программирования. После уроков они частенько зависают в кабинете информатики, решая олимпиадные задачи и обсуждая тонкости алгоритмов.

Однажды, готовясь к региональной олимпиаде, они придумали игру: каждый из них написал несколько строк S_i, состоящих только из круглых скобок '(' и ')'. В итоге у них получилось N строк — не обязательно правильных, но каждая из них может быть частью большой правильной скобочной последовательности.

Правильная скобочная последовательность — это такая строка, в которой:

- каждая открывающая скобка '(' имеет соответствующую ей закрывающую скобку ')';
- открывающие скобки '(' должны быть записаны раньше своих закрывающих скобок ')';
- нет одиночных скобок, то есть таких, у которых нет пары.

Айтал предложил интересную задачу: можно ли упорядочить эти N строк в каком-либо порядке так, чтобы при их склеивании (то есть соединении в одну большую строку) получилась правильная скобочная последовательность?

Григорий задумался... А ты сможешь помочь им?

Формат входных данных

В первой строке записано единственное целое положительное число N ($1 \leqslant N \leqslant 4 \cdot 10^4$) — количество строк со скобочными последовательностями.

Далее идут N строк, в каждой из которых содержится скобочная последовательность S_i.

Гарантируется, что $\sum_{i=1}^{N} |S_i| \leqslant 10^5$, где $|S_i|$ — длина i-й скобочной последовательности.

Формат выходных данных

В первой строке нужно вывести «Yes», если существует хотя бы один вариант составления правильной скобочной последовательности, использующей все N строк, либо «No» в противном случае.

Если ответ существует, то во второй строке вывести последовательность номеров строк, в соответствии с которой нужно расставить скобочные последовательности. Строки нумеруются с 1 в порядке появления во входных данных. Если ответов несколько, выведите любой.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод

`3`
`)()(`
`)`
`(`
	`Yes`
	`3 1 2`

D. Каверна

«Хитрые зайцы» попали в опасную каверну и должны как можно быстрее выбраться из неё. Каверна представляет собой прямоугольное клеточное поле размером $N \times M$, где каждая клетка может:

- быть пустой;
- содержать препятствие, через которое нельзя пройти;
- содержать опасных эфириалов, с которыми необходимо сразиться при входе в клетку.

Помогите «хитрым зайцам» выбраться из каверны с минимальным числом сражений, если они могут передвигаться только по соседним по стороне клеткам. Они начинают в клетке $(1, 1)$ и должны добраться до клетки (N, M).

Формат входных данных

Первая строка содержит два целых числа N и M ($1 \leqslant N \cdot M \leqslant 10^5$) — размеры каверны.

Далее следуют N строк длины M, описывающие каверну:

- символ «`.`» обозначает пустую строку;

- символ «#» обозначает препятствие;
- символ «X» обозначает клетку с эфириалом.

Гарантируется, что клетки $(1, 1)$ и (N, M) всегда пустые.

Формат выходных данных

Выведите единственное целое число — минимальное количество сражений с эфириалами, чтобы дойти от клетки $(1, 1)$ до клетки (N, M). Если пути не существует, выведите -1.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод

`3 3`
`...`
`#X.`
`...`
	`0`

`5 5`
`..X..`
`..#X#`
`.X...`
`.#...`
`XX.#.`
	`1`

`3 4`
`...#`
`##.#`
`XX#.`
	`-1`

E. Колесо в графе

Коля по прозвищу «Легенда» придумал новый тип графов — *колесо*. Колесом он называет граф, содержащий не менее 4 вершин, который можно изобразить на бумаге в виде «колеса» — все вершины, кроме одной (которая является «центром колеса») образуют цикл, и от каждой из них идёт по ребру к центру колеса. Других рёбер в графе нет.

Теперь Легенда ищет колеса во всех графах, которые он видит. Сегодня он смог раздобыть весьма интересный граф — неориентированный связный граф без петель и кратных ребер из n вершин и m ребер. Легенда хочет найти колесо в этом графе. Он считает, что в графе имеется колесо, если из графа можно убрать некоторые вершины и ребра таким образом, чтобы оставшийся граф стал этим самым колесом.

Разумеется, Коля (Легенда) моментально нашел колесо (или понял, что нет никакого колеса), но он хочет проверить, сможете ли вы найти колесо?

Формат входных данных

В первой строке заданы два целых числа n и m ($0 < n \leqslant 5000$, $0 \leqslant m \leqslant n \leqslant n(n-1)/2$) — количество вершин и количество ребер в графе.

В следующих m строках задаются ребра графа: в i-ой строке записаны два числа u и v ($0 < u, v \leqslant n$, $u \neq v$), означающие, что между вершинами u и v есть ребро.

Формат выходных данных

Если колес в графе нет, то выведите -1.

Иначе в первой строке выведите два числа: k ($k \geqslant 4$) — количество вершин в найденном колесе и номер центра колеса, во второй строке — остальные $k-1$ вершин колеса в порядке обхода по циклу.

Если есть несколько колес, то выведите любое из них.

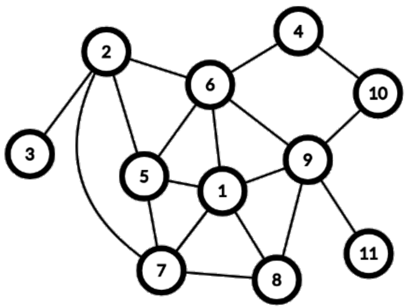
Примеры входного и выходного файлов

<table>
<tr><th>стандартный ввод</th><th>стандартный вывод</th></tr>
<tr><td><pre>11 17
1 6
9 1
5 1
11 9
9 10
2 3
2 5
10 4
4 6
6 9
5 6
7 8
8 1
1 7
2 6
7 5
2 7</pre></td><td><pre>5 5
7 2 6 1</pre></td></tr>
</table>

стандартный ввод	стандартный вывод
6 10 1 2 4 5 6 1 1 4 4 3 1 3 5 1 4 6 2 5 4 2	4 1 4 5 2
5 6 1 3 3 4 3 2 2 5 4 5 4 1	-1

Комментарии

Иллюстрация к примеру 1:



В данном графе имеется два колеса: $\{1, 5, 6, 9, 8, 7\}$ и $\{5, 1, 7, 2, 6\}$.

F. Время гринда

Коля продолжает играть в игры от компании TimurKul Games. Недавно он скачал игру «Время приключений 2: в погоне за деньгами», которая, к сожалению, отличается своей реиграбельностью.

Игру можно представить как n последовательных локаций, каждая из которых характеризуется значением a_i — энергией, требуемой для прохождения локации. Изначально у Коли имеется k единиц энергии. Игра проходит следующим образом:

1. Вначале игрок находится в локации $i = 1$.
2. Игрок может пройти i-ю локацию, если у него имеется хотя бы a_i энергии. В противном случае игра заканчивается.
3. После прохождения i-й локации от энергии игрока отнимается a_i, а игрок перемещается в локацию $i + 1$. Если после прохождения последней локации, у игрока все еще имеется энергия, игрок перемещается к первой локации $i = 1$, и на свое усмотрение усложняет одну из локаций, то есть к выбранному игроком значению a_i прибавляется величина c.

Проблема в том, что значение c выбирается самим игроком. Коля хочет, чтобы выполнялось неравенство $0 \leqslant c \leqslant 10^9$ и чтобы максимально возможное количество локаций, которое возможно пройти, было не меньше m. Если подходящих значений c несколько, то он предпочитает самое максимальное. Помогите Коле узнать, какое значение c ему выбрать.

Формат входных данных

Первая строка содержит три целых числа n, m и k $(1 \leqslant n \leqslant 10^5; 0 \leqslant m \leqslant 10^9; 0 \leqslant k \leqslant 10^9)$ — количество локаций, требуемое количество пройденных локаций и начальное количество энергии соответственно. Гарантируется, что у заданного m существует хотя бы одно c, которое удовлетворяет вышеописанным условиям.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n (1 \leqslant a_i \leqslant 10^4)$ — значения энергии, необходимые для прохождения локаций.

Формат выходных данных

Выведите максимальное из подходящих значений c.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
2 5 14 2 3	2
3 0 10 2 4 6	1000000000
1 500000 1000000000 1	0

Комментарии

В первом примере сначала $a = [2, 3]$, и Коля находится в локации 1. Он может пройти обе локации, потратив на это 5 единиц энергии. После прохождения локации 2 Коля должен прибавить к какой-либо локации $c = 2$. Пусть он выберет a_2. После всего этого Коля находится в локации 1, имеет энергию $k = 9$ и $a = [2, 5]$. Коля может снова пройти обе эти локации, потратив на это 7 единиц энергии и прибавив к a_2 2. Наконец, Коля снова находится в локации 1, имеет энергию $k = 2$ и $a = [2, 7]$. Коля может пройти первую локацию, но застрянет на второй. В итоге он прошёл 5 локаций, что не меньше $m = 5$. Можно показать, что c не может быть больше 2.

Во втором примере Коля, ничего не делая, уже выполняет свой минимум по пройденным локациям, из-за чего значение c можно выбрать любое. Поскольку Коля предпочитает максимальное из возможных c, ответом является 10^9.

G. Соревнование

В зимних соревнованиях по рыбной ловле участники ловили рыбу в n лунках, расположенных в один ряд. По окончании контрольного времени жюри подсчитало количество рыб, пойманных в каждой лунке.

Оказалось, что в любых двух соседних лунках количество рыб отличается ровно на единицу, а в одной из лунок поймали m рыб. Общее количество всех пойманных рыб может быть достаточно большим и принимать различные значения. Например, если лунок 4 и в одной из них поймали 3 рыбы, то суммарное количество пойманных рыб может принимать значения: 6, 8, 10, 12, 14, 16, 18.

Помогите жюри определить, сколько различных значений может принимать общее количество пойманных рыб?

Формат входных данных

В единственной строке записаны два целых неотрицательных числа n и m, разделенных пробелом ($0 \leqslant m \leqslant 10^6$, $1 \leqslant n \leqslant 10^6$).

Формат выходных данных

Программа должна напечатать одно число — количество различных значений общего количества рыб.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
4 3	7
5 0	8

H. Радикал

В самом сердце одной страны лежит магический камень, на котором выгравировано натуральное число n. С каждым новым указом синей партии число на камне увеличивается на *радикал* этого числа. Формально, после каждого указа значение n изменяется по правилу: $n \to n + \mathrm{rad}(n)$.

Радикалом числа называют произведение всех его различных простых делителей:

- $\mathrm{rad}(1) = 1$;
- Если $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k}$, то $\mathrm{rad}(n) = p_1 \cdot p_2 \cdot \ldots \cdot p_k$.

Ваша задача — определить, какое число будет написано на камне после k указов синей партии. Поскольку результат может быть очень большим, выведите его по модулю $10^9 + 7$.

Формат входных данных

В единственной строке содержатся два целых числа n и k ($1 \leqslant n \leqslant 10^6$, $0 \leqslant k \leqslant 10^3$).

Формат выходных данных

Выведите единственное целое число — ответ на задачу по модулю $10^9 + 7$.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
12 4	60
34 76	669278610

I. Фишки

В каждой клетке доски $3 \times n$ стоит фишка одного из трёх цветов, причём всего фишек каждого цвета на доске поровну. Внутри каждой из трёх строк разрешается переставлять фишки в любом порядке. Расставьте их так, чтобы в каждом столбце были фишки всех трёх цветов.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 10^5$) — количество столбцов на доске.

Далее следуют три строки, каждая из которых содержит n чисел от 1 до 3, обозначающих цвета фишек. Гарантируется, что фишек каждого цвета поровну.

Формат выходных данных

Выведите три строки по n чисел — итоговую расстановку фишек. Если такой расстановки не существует, выведите -1.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
4 1 3 3 2 3 2 1 1 2 1 3 2	1 3 2 3 3 2 1 1 2 1 3 2

J. Игра на вычитание

Айаал и Яша играют в следующую игру на доске. Изначально на доске записаны два различных натуральных числа: A и B. Игроки ходят по очереди, начиная с Айаала.

За один ход игрок может записать на доску новое натуральное число, равное разности любых двух уже имеющихся чисел, если этого числа ещё нет на доске.

Проигрывает тот, кто не может сделать ход.

Определите, кто победит при оптимальной игре — Айаал или Яша?

Формат входных данных

В единственной строке содержатся два натуральных числа A и B ($1 \leqslant A, B \leqslant 10^6$) — начальные числа, записанные на доске. Гарантируется, что $A \neq B$.

Формат выходных данных

Выведите «`Ayal`», если выиграет Айаал, иначе — «`Yasha`».

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
2 1	Yasha
25 37	Ayal

K. Чиселки

Назовём натуральное число *чиселкой*, если оно является наименьшим числом среди всех чисел с той же суммой цифр. Ваша задача — найти N-ую чиселку среди всех натуральных чисел.

Формат входных данных

Единственная строка содержит единственное натуральное число N ($1 \leqslant N \leqslant 1000$).

Формат выходных данных

Выведите единственное натуральное число — ответ на задачу.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
7	7
19	199

L. Тимур любит куличи

Совсем скоро будет Пасха, и Тимур очень ждет этот праздник. Особенно он любит пасхальные куличи.

Куличи бывают разных размеров, для простоты будем считать, что каждый кулич можно представить в виде прямоугольника на плоскости с целочисленными координатами вершин и сторонами, параллельными осям координат.

В стране, где он живет, принято разрезать куличи по диагонали на две равные половинки. Тогда половинки представляют собой прямоугольные треугольники с целочисленными координатами вершин и катетами, параллельными осям координат.

Тимуру подарили две такие половинки от куличей. Он задался вопросом: можно ли эти половинки совместить в целый кулич с помощью параллельного переноса? Вращать половинки при этом **нельзя**.

Он дал вам координаты вершин двух половинок и просит помочь найти ответ на вопрос.

Формат входных данных

В первой строке заданы 6 целых чисел x_1, y_1, x_2, y_2, x_3, y_3 — соответственно, координаты первой, второй и третьей вершин треугольника, представляющего первую половинку кулича Тимура. **Гарантируется**, что вершина с координатами x_1, y_1 лежит напротив гипотенузы.

Во второй строке заданы 6 целых чисел $x_4, y_4, x_5, y_5, x_6, y_6$ — соответственно, координаты первой, второй и третьей вершин треугольника, представляющего вторую половинку кулича Тимура. **Гарантируется**, что вершина с координатами x_4, y_4 лежит напротив гипотенузы.

Все координаты не превышают 10^9 по модулю.

Формат выходных данных

Выведите «YES», если две данные половинки могут образовать один кулич, в противном случае выведите «NO».

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
5 4 5 2 -1 4 8 -4 14 -4 8 -2	YES
5 -1 5 -4 8 -1 1 3 1 6 -2 3	YES

РЕШЕНИЯ

A. Мы обожаем деревья

Давайте сначала решим задачу для дерева без портала. Заметим, что ответом для вершины i является максимум из 0, ответов потомков и ответов потомков потомков, к которому прибавлен a_i. Данные ответы будем хранить в массиве `dp_down[n]`.

Далее, заметим, что при введении портала, ответом для вершины 1 является либо `dp_down[1]`, либо счёт, который получается, если мы оптимально дойдём до портала, воспользуемся им и оптимально дойдём до вершины 1. Пусть портал находится в вершине u и ведёт в вершину v. Перед использованием портала максимальный счёт равен `dp_down[u]`. Пусть максимальным счётом от вершины v до вершины 1 является `dp_up[v]`. Теперь надо понять, как заполнить этот массив.

Давайте перефразируем задачу для заполнения `dp_up[n]`. Мы начинаем в вершине 1 и можем переместиться либо в потомка, либо в потомка потомка. Базой `dp_up[n]` является `dp_up[1]` = a_1. Далее, ответом для `dp_up[i]` является максимум из ответов предков и ответов предков предков, к которому прибавлен a_i. Заметьте, что `dp_up[i]` может быть меньше a_i, так как мы уже не можем пропускать вершину от вершины 1 до вершины i.

Наконец, ответом для запроса u, v является `max(dp_down[1], dp_down[u] + dp_up[v])`.

B. Обратная конкатенация

Сначала сделаем наблюдение, что последнее число по-любому будет стоять первым в конечном числе. Далее, заметим, что порядок расположения всех остальных чисел после последнего числа может быть любым. Это значит, что наша задача - расположить числа $a_1, a_2, \ldots, a_{n-1}$ в таком порядке, чтобы конечное число было максимальным.

Сравним два возможных конечных числа b и c. Представим их в виде $b_1 b_2 \ldots b_{n-1}$ и $c_1 c_2 \ldots c_{n-1}$. Пусть $b_i = c_i$ для $1 \leqslant i < x$ ($x \leqslant n-1$) и $b_x < c_x$. В таком случае, очевидно, что $b < c$. Случай при $b_x > c_x$ рассматривается аналогично. Это означает, что мы можем жадно ставить в качестве очередного числа максимальный из неиспользованных чисел.

Пусть из $a_1, a_2, \ldots, a_{n-1}$ мы можем получить максимальное число $m_1 m_2 \ldots m_{n-1}$. Ответом будет $a_n m_1 m_2 \ldots m_{n-1}$.

C. Скобки

В данной задаче необходимо заметить следующее:

- Для того чтобы можно было составить правильную скобочную последовательность, количество открывающихся скобок должно быть равно количеству закрывающихся;
- Если существует ответ на эту задачу, то он всегда будет начинаться и заканчиваться открывающейся скобкой, и заканчиваться закрывающейся.

Теперь рассмотрим варианты скобочных последовательностей, которые могут передаваться во входных данных:

1. Только открывающиеся скобки

$$\underbrace{((\cdots(((}_{n\text{ скобок}}$$

   В таком случае легко заметить, что их можно расположить в начале результирующей последовательности, при этом порядок расположения не важен;
2. Только закрывающиеся скобки

$$\underbrace{))\cdots)))}_{m\text{ скобок}}$$

   Аналогично первому варианту, их можно разместить в конце итоговой последовательности в любом порядке;
3. Сначала закрывающиеся, потом открывающиеся скобки

$$\underbrace{))\cdots)))}_{m\text{ скобок}}\ \underbrace{((\cdots(((}_{n\text{ скобок}}$$

   Перед расположением данного варианта последовательности необходимо, чтобы перед ней было как минимум m открывающихся скобок без пары;
4. Произвольный набор скобок
   Легко заметить, что если удалить все правильные скобочные подпоследовательности, то результатом будет последовательность типа 1, 2 или 3.
   В решении задачи можно упростить последовательность, удалив все подпоследовательности () пока это возможно.

После упрощения всех последовательностей во входных данных, можно вывести следующий алгоритм, который позволит найти ответ:

1. Разместить в начале строки все последовательности типа 1;
2. Разделить последовательности типа 3 на две группы:
    (a) количество открывающихся скобок больше или равно количеству закрывающихся,
    (b) количество закрывающихся скобок больше, чем открывающихся;
3. Каждую группу отсортировать по возрастанию количества закрывающихся скобок;
4. Разместить первую группу, затем вторую группу в получившемся порядке;
5. Разместить в конце строки все последовательности типа 2;
6. Убедиться, что получившаяся последовательность является правильной.

Если получившаяся последовательность не является правильной, значит ответа нет. Иначе эта последовательность является ответом на задачу. Необходимо учитывать, что в ответе нужно вернуть номера строк входных данных, поэтому при сортировке и манипуляциях с последовательностью необходимо сохранять эти данные.

D. Каверна

Преобразуем карту каверны в ориентированный взвешенный граф: ячейки станут вершинами, а рёбра будут иметь вес 1, если ведут в клетку `<<X>>`, или 0, если ведут в клетку `<<.>>`. Поскольку получившийся граф ориентированный и его рёбра имеют веса 0 и 1, на нём можно запустить алгоритм `0-1 BFS`.

E. Колесо в графе

Будем перебирать центр колеса: для очередной вершины v сделаем его центром возможного колеса и будем искать цикл среди всех вершин, смежных с v. Поиск цикла делается стандартным обходом в глубину (игнорируя все вершины, не смежные с центром колеса).

F. Время гринда

Есть 2 способа решить данную задачу: с помощью бинарного поиска, с помощью формулы.

Первый способ заключается в том, что если при каком-то c количество локаций, которое может пройти Коля, не меньше m, то при $c-1$ данное условие тоже выполняется. Также замечаем, что начиная с какого-то c количество локаций, которое пройдёт Коля, будет меньше m. Всё это значит, что c можно перебрать с помощью бинарного поиска.

Далее, мы должны быстро вычислить количество локаций, которое Коля сможет пройти при каком-то c. Сначала сделаем вывод, что Коле выгоднее всего прибавить c к последней локации. Если мы будем обрабатывать локацию за локацией, пока энергия не кончится, то наше решение не пройдёт по времени (например, если массив a представляет из себя 100000 единичек). Есть очень лёгкий способ оптимизировать процесс вычисления пройденных локаций: если мы находимся в локации 1 и мы можем пройти все локации, то давайте не будем поштучно обрабатывать все локации, а просто добавим к счётчику n и вычтем из энергии сумму a_i всех локаций и «индекс» текущего прохождения. Это будет работать быстро по той причине, что каждое новое прохождение характеризуется увеличением суммы всех a_i хотя бы на 1. Это значит, что примерно через 45000 прохождений всех локаций даже если $a = [1]$, то энергия Коли всё равно закончится.

Учтите, что $c = 0$ не надо проверять, поскольку условие гарантирует, что при $c = 0$ Коля всегда сможет пройти m локаций.

Для решения задачи с помощью второго способа, найдём количество энергии, требуемое для того, чтобы пройти m локаций при $c = 0$. Данное значение равно (индексация с нуля):

$$(a_0 + a_1 + \ldots + a_{n-1}) \cdot \left\lfloor \frac{m}{n} \right\rfloor + a_0 + a_1 + \ldots + a_{m \bmod n} \tag{1}$$

Далее, если мы будем применять операцию только для последней локации, то количество дополнительной энергии можно выразить с помощью следующей формулы:

$$c \cdot \frac{\left(\left\lfloor \frac{m}{n} \right\rfloor - 1\right) \cdot \left\lfloor \frac{m}{n} \right\rfloor}{2} \tag{2}$$

С помощью (1) и (2), выразим условие, выполнение которого означает, что c подходит:

$$(a_0 + a_1 + \ldots + a_{n-1}) \cdot \left\lfloor \frac{m}{n} \right\rfloor + a_0 + a_1 + \ldots + a_{m \bmod n} + c \cdot \frac{\left(\left\lfloor \frac{m}{n} \right\rfloor - 1\right) \cdot \left\lfloor \frac{m}{n} \right\rfloor}{2} \leqslant k \tag{3}$$

Пусть $k' = k - (1)$, а $b = \dfrac{\left(\left\lfloor \frac{m}{n} \right\rfloor - 1\right) \cdot \left\lfloor \frac{m}{n} \right\rfloor}{2}$. Вычтем из обеих частей (1):

$$c \cdot b \leqslant k' \tag{4}$$

Пусть Коля может пройти все локации хотя бы 2 раза. Тогда (4) равносильно:

$$c \leqslant \frac{k'}{b} \tag{5}$$

Наконец, найдём ответ. Если Коля не сможет пройти все локации хотя бы 2 раза, то ответом является 1 000 000 000, поскольку от количество требуемой энергии не зависит от c. В противном случае ответом является:

$$c = \left\lfloor \frac{k'}{b} \right\rfloor \tag{6}$$

G. Соревнование

Обозначим

- SumMax(m,n) — максимальная сумма пойманных рыб на n лунках, если в одной из лунок поймали m рыб;
- SumMin(m,n) — минимальная сумма пойманных рыб на n лунках, если в одной из лунок поймали m рыб.

Очевидно, что максимальная сумма будет равна

`SumMax(m,n) = m + (m+1) + (m+2) + …+ (m+n-1) = (2m+n-1)*n/2.`

Вычисление минимальной суммы требует рассмотрения двух вариантов:

- $m < n$: `SumMin(m, n) = m +(m-1) + (m-2) + …+ 1 + 0 +1 + 0 + 1 + …= SumMax(0, m+1) + (n -- m) / 2;`
- $m \geqslant n$: `SumMin(m,n) = m +(m-1) + (m-2) + …+ m -- n + 1)= SumMax(m -- n + 1, n)`

Всевозможные суммы для четного числа n всегда будут четными или нечетными, т.к. суммы всех пар нечетны, а не пересекающихся пар либо четно, либо нечетно (см. пример). Для нечетного числа n суммы могут быть как четными, так и нечетными. Эти суммы можно разбить на два множества, с четными и нечетными элементами. Первая максимальная сумма вычислена.

Вычислим вторую максимальную сумму с другой четностью элементов:

`MaxSum = (m+1) + m + (m+1) + …+ (m+n-2) = m + 1 + SumMax(m,n-1)`

минимальная сумма для m > 0:

`MinSum = (m-1) + m + (m-1) + …+ (m+n-3) = m - 1 + SumMin(m, n-1)`

для m = 0:

`MinSum = 1 + 0 + 1 + …+ 0 + 1 = 1 + SumMin(0, n-1)`

Для четного n:

`ans = (SumMax(m,n) - SumMin(m,n))/2 + 1;`

для нечетног n:

`ans = ans + (MaxSum - MinSum)/2 + 1;`

H. Радикал

Обозначим для числа N как его «остаток» $r = \dfrac{N}{\mathrm{rad}(N)}$. Заметим, что при каждом изменении исходного числа:

1. Увеличивается на 1. Например:

$$N = 12, \quad \mathrm{rad}(N) = 6, \quad r = 2$$

$$\rightarrow \qquad N = 18, \quad \mathrm{rad}(N) = 6, \quad r = 2 + 1 = 3.$$

2. Потом делится на все свои простые делители, которых не было у N. Например:

$$N = 24, \quad \mathrm{rad}(N) = 6, \quad r = 4$$

$$\rightarrow \qquad N = 30, \quad \mathrm{rad}(N) = 30, \quad r = (4 + 1) \div 5 = 1.$$

В языках программирования с длинной арифметикой достаточно просимулировать алгоритм, вычисляя радикал за разумное время $O(\min(\sqrt{N}, \log N))$. Для длинных чисел радикал можно вычислить за логарифм, так как самым большим простым делителем в разложении может добавиться только k-ое простое число, как было показано выше.

Для языков без длинной арифметики можно хранить множество простых делителей, остаток r и проверять с каждым новым изменением, не появляются ли у r новые делители.

I. Фишки

Есть множество способов построения искомой расстановки. Мы покажем, почему ответ всегда существует. Доказательство проведём с помощью математической индукции:

- **База:** при $n = 1$ очевидно, что любая такая доска 3×1 с одной фишкой каждого цвета удовлетворяет искомой расстановке.
- **Предположение:** предположим, что любую расстановку фишек на доске $3 \times n$ можно привести к искомой.
- **Переход:** в доске $3 \times (n + 1)$ сформируем первый столбец, состоящий из всех трёх цветов, а оставшиеся n столбцов переставим, следуя предположению (ведь у нас останется по n штук фишек каждого цвета).

J. Игра на вычитание

Прежде всего заметим, что по алгоритму Евклида на доске в конечном итоге окажется число, равное наибольшему общему делителю A и B. С другой стороны, с помощью последовательных вычитаний невозможно получить число, которое не делится на этот наибольший общий делитель. Таким образом, на доске будут записаны все числа:

$$\text{НОД}(A, B), 2 \cdot \text{НОД}(A, B), 3 \cdot \text{НОД}(A, B), \dots, \max(A, B).$$

Анализируя чётность количества этих чисел, можно легко определить, кто победит в любой игре.

K. Чиселки

Самым маленьким числом среди всех чисел с заданной суммой цифр является то, которое имеет в первую очередь минимальную длину. Это достигается размещением наибольших возможных цифр (девяток) в младших разрядах. Следовательно, чиселка с суммой цифр S — это наименьшее число, сумма цифр которого равна S, при этом оно должно состоять из максимального количества цифр «9» в младших разрядах, а самый старший разряд — это остаток от деления S на 9.

L. Тимур любит куличи

Диагональ прямоугольника делит его на два равных треугольника, значит нам нужно проверить что треугольники равны. Для этого достаточно проверить равенство длин катетов. Также нужно проверить что соответствующие катеты лежат в разные стороны от точки лежащей напротив гипотенузы.

Содержание