

The 25th International Olympiad **ТУУМААДА**
XXV Международная олимпиада **ТУЙМААДА**



PROBLEMS AND SOLUTIONS

Informatics

Информатика

ЗАДАЧИ И РЕШЕНИЯ

Yakutsk 2018
Якутск 2018

This booklet contains the problems and possible solutions of the 25th Tuymaada International Olympiad in informatics. The olympiad took place on 8–15 July 2018 in Yakutsk, Russia. The participants were invited to solve four problems on each of the two competition days.

XXV Международная олимпиада школьников по математике, физике, химии и информатике «Туймаада». ИНФОРМАТИКА. – Якутск, 2018.

Сборник содержит задачи XXV Международной олимпиады «Туймаада» по информатике, а также возможные варианты решений. Олимпиада проводилась 8–15 июля 2018 года в г. Якутске. Олимпиада прошла в два дня, в каждый из которых участникам было предложено решить четыре задачи.

АВТОРЫ | AUTHORS

<i>Grigory SPIROV</i> <i>North-Eastern Federal University</i>	A Григорий Александрович СПИРОВ Северо-Восточный федеральный университет
<i>Artem VASILYEV</i> <i>ITMO Univeristy</i>	Артем Тарасович ВАСИЛЬЕВ Университет ИТМО
<i>Alexander PAVLOV</i> <i>North-Eastern Federal University</i>	B Александр Викторович ПАВЛОВ СВФУ
<i>Kirill TCHIKHATCHEV</i> <i>Physics and Mathematics Lyceum No. 239, St Petersburg</i>	C Кирилл Борисович ЧИХАЧЁВ физико-математический лицей №239, Санкт-Петербург
<i>Aital DYAKONOV</i> <i>NEFU</i>	D Айтал Викторович ДЪЯКОНОВ СВФУ
<i>Vladimir EVERSTOV</i> <i>NEFU</i>	E Владимир Васильевич Эверстов СВФУ
<i>Yuri ANTONOV</i> <i>NEFU</i>	F Юрий Саввич АНТОНОВ СВФУ
<i>Artem VASILYEV</i>	G Артем Тарасович ВАСИЛЬЕВ
<i>Artem VASILYEV</i>	H Артем Тарасович ВАСИЛЬЕВ

Problems

A. Liberdance

Time limit: 2 seconds

Memory limit: 256 megabytes

Inhabitants of the planet Liberloun love to dance. The ballroom of the main Liberloun palace is a $h \times w$ rectangle, where h is the height and w is the width of the rectangle. Cells of this rectangle are enumerated from 1 to hw as shown in the picture.

1	2	...	w
$w + 1$	$w + 2$...	$2w$
...
$(h - 1)w + 1$	hw

The dance that is going to happen is really simple and only consists of one movement, that is repeated k times. In the beginning each cell is occupied by one dancer. Then, one movement consists of every dancer moving to an adjacent cell. The direction only depends on the cell itself regardless of the dancer standing there.

The staff of Liberloun's main palace are preparing for a closing ceremony of Liberloun International Science Games, and are asking for your help. Help the dancers figure out their positions after k dance movements.

Input

First line contains 3 integers h , w , and k ($1 \leq h, w \leq 100$, $0 \leq k \leq 10^9$, hw is even) – ballroom size and number of movements. Next h lines contain the movement description. Each line has w characters «L», «R», «D» or «U», corresponding to moving left, right, down and up. Direction from each cell leads to another cell of the rectangle. There is exactly one cell leading to any particular cell.

Output

Output the final dancers' position in the following format: each of h lines should contain w dancers numbers, standing in the corresponding cell.

Example

standard input	standard output
3 4 3	10 9 4 8
DLRD	6 5 3 7
DUUL	2 1 12 11
RURL	

Note

This problem contains two subtasks. Points for a subtask are awarded only if solution passes all the tests from this subtask and preceding subtasks.

Subtask 1 (points: 50)

$k \leq 100$.

Subtask 2 (points: 50)

No additional limitations.

B. Festival of the Stars

Time limit: 1 second
 Memory limit: 256 megabytes

Your spaceship is damaged and in need of repairs. The closest planet where you can repair your ship is called Y. Unfortunately, a purification season has been announced on the planet. During the purification season, the planet closes off its spaceports from the outside world. It is known that the spaceports will reopen on the day of the next Stars Festival.

The ancient culture of Y numbers the days "from the creation of the world". The Festivals of the Stars are held on such days that each digit k in the number of the day appears in it exactly k times. For example, on day 333212 there is a Festival of the Stars.

Determine when the spaceports of planet will open, knowing today's number in the calendar of Y.

Input

A single integer N ($0 < N < 10^{45}$), today's number in the calendar of Y.

Output

A single integer, the number of the day when the next Festival of the Stars will be held.

Решения 32

A. Либербал 32

B. Празднество Звезд на планете Y 32

C. Игра в слова 33

D. Вулканы 34

E. Пенальти 35

F. Еще одно необычное уравнение 36

G. Оценка кластеризации 36

H. Параллельные вычисления 37

Final Standings | Итоги олимпиады 39

Contents | Содержание

Problems	1
A. Liberdance	1
B. Festival of the Stars	2
C. A Game of Words	3
D. Volcanoes	5
E. Penalty	6
F. Yet another unusual equation	9
G. Clustering evaluation	10
H. Parallel computing	11
Solutions	13
A. Liberdance	13
B. Festival of the Stars	13
C. A Game of Words	14
D. Volcanoes	15
E. Penalty	15
F. Yet another unusual equation	16
G. Clustering evaluation	17
H. Parallel computing	17
Задачи	19
A. Либербал	19
B. Празднество Звезд на планете Ы	20
C. Игра в слова	21
D. Вулканы	23
E. Пенальти	25
F. Еще одно необычное уравнение	28
G. Оценка кластеризации	29
H. Параллельные вычисления	30

Scoring

This problem contains three subproblems. Points will be awarded for a subproblem only if all the tests in it pass. Subproblems are evaluated independently.

Subtask 1 (points: 30)

$N \leq 2^{31}$.

Subtask 2 (points: 70)

No additional limitations.

Examples

standard input	standard output
19	22
22	122

C. A Game of Words

Time limit: 2 seconds
Memory limit: 256 megabytes

In a popular Russian game of *Words*, several players take turns saying words from a certain set so that the first letter of the next word is the same as the last letter of the previous word. The first word is chosen randomly. Words cannot be repeated. Commonly used word sets include, for example, the names of cities, plants, or animals.

Vasya loves this game, however, he noticed that in certain cases, some words from the set cannot be called no matter how the players make their moves. Now Vasya is curious to find the minimal number of new words that must be added to the set so that for any choice of two words, let us call them *initial* and *target*, the players can make moves starting with the initial word to eventually call the target word.

Input

The first line contains two numbers N and M ($1 \leq N, M \leq 100\,000$), the number of letters in the alphabet and the number of words, respectively. The following M lines describe the words from the set, each i th line containing the number of the first and last letter of the word with number i . The letters of the alphabet are numbered from 1 to N .

Problem statements

Output

Output a single integer, the minimal number of words that must be added to the set.

Scoring

This problem contains three subproblems. Points will be awarded for a subproblem only if all the tests in it pass. Subproblems are evaluated independently.

Subtask 1 (points: 30)
 $N, M \leq 20$.

Subtask 2 (points: 30)
 No additional limitations on N and M . For every letter of the alphabet the set contains a word beginning and/or ending with that letter.

Subtask 3 (points: 30)
 No additional limitations.

Example

	standard input	standard output
9	11	2
1	2	
2	3	
3	1	
4	5	
5	6	
6	4	
7	8	
8	9	
9	7	
1	4	
1	7	

23	FONG Hin Tung ФОНГ Хин-Тун	Hong Kong Гонконг	50	0	0	0	50	30	0	.	0	30	80	
24	MALTSEVA Yuliya МАЛЬЦЕВА Юлия	Россия Russia	50	.	.	.	50	.	30	.	.	30	80	
25	KHOVROV Dmitriy ХОВРОВ Дмитрий	Россия Russia	50	.	.	.	50	0	.	.	.	0	50	
26	WONG Wing Yan ВОНГ Вин-Ян	Hong Kong Гонконг	.	0	0	10	10	0	.	.	0	0	10	
27	ZAKHAROV Aytal ЗАХАРОВ Айтал	Россия Russia	0	.	.	0	0					0	0	
28	ANTONOV Anatoliy АНТОНОВ Анатолий	Россия Russia	.	0	0	.	0	0	.	0	.	0	0	

13.07.2018

НАГРАДЫ / AWARDS: 1DD – диплом 1 степени / 1st degree diploma,
 2DD – диплом 2 степени / 2nd degree diploma,
 3DD – диплом 3 степени / 3rd degree diploma,
 HML – почетная грамота / Honorable mention letter,
 SA – спецприз / special award

11	SEменов Nikita СЕМЕНОВ Никита	Россия Russia	100	.	0	30	130	100	30	70	.	200	330	3DD
12	РОРОВ Kirill ПОПОВ Кирилл	Россия Russia	100	0	.	.	100	100	68	.	0	168	268	3DD
13	EGOROV Michil ЕГОРОВ Мичил	Россия Russia	100	0	0	10	110	100	30	.	0	130	240	3DD
14	VOSOEVA Elvira БОСОЕВА Эльвира	Россия Russia	50	.	.	10	60	100	30	30	.	160	220	HML
15	АРМЕНАКУАН Karen АРМЕНАКЯН Карен	Россия Russia	100	0	.	0	100	30	30	30	0	90	190	HML
16	ULYANOV Alexey УЛЬЯНОВ Алексей	Россия Russia	50	0	.	0	50	100	.	30	0	130	180	HML
17	FILATOV Vaydam ФИЛАТОВ Вайдам	Россия Russia	50	.	.	30	80	100	0	.	0	100	180	HML
18	TSE Hoi Chun ЦЕ Хой-Чун	Hong Kong Гонконг	50	30	0	10	90	0	30	30	.	60	150	
19	IVANOV Danil ИВАНОВ Данил	Россия Russia	100	.	.	0	100	30	.	.	.	30	130	
20	VELYAEV Vladimir БЕЛЯЕВ Владимир	Россия Russia	50	.	0	20	70	.	0	30	0	30	100	
21	ZAKHAROV Alexey ЗАХАРОВ Алексей	Россия Russia	.	0	0	20	20	30	30	.	0	60	80	
22	BADANOV Shingis БАДАНОВ Чингис	Россия Russia	50	.	.	.	50	30	.	.	.	30	80	SA

D. Volcanoes

Time limit: 10 seconds
Memory limit: 256 megabytes

Scientists have long been interested in volcanic activity on a certain island. The island is an $M \times M$ square, composed of unit cells. Its rows and columns are numbered from 1 to M .

For each volcano its location and the day of the initial eruption are known. The volcanoes spew lava which on the first day only occupies the cell of the initial eruption, on the second day expands to floods a 3×3 square, on the third day floods a 5×5 square, and so on. All these squares are centered on the cell of the initial eruption. Volcanoes do not influence each other and the lava from each one flows independently from the lava of the others.

For further research, a station needs to be placed on the island. The station occupies one cell. Find the maximum number of days the station can operate. The station ceases operation when its cell is flooded by lava.

Input

The first line contains 2 integers N and M ($1 \leq N, M \leq 150\,000$), the number of volcanoes and the size of the island. The following N lines describe volcanoes, with each i th of the containing 3 integers x_i, y_i, t_i ($1 \leq x_i, y_i \leq M, 1 \leq t_i \leq 150\,000$) — the coordinates and the day of the initial eruption of the i th volcano. Eruption locations do not repeat. It is guaranteed that the station will be operational for at least one day.

Output

Output the maximum number of days that a station placed on the island can operate.

Scoring

This problem contains four subproblems. Points for a subtask are awarded only if solution passes all the tests from this subtask. Subproblems are evaluated independently.

Subtask 1 (points: 10)

$N = 1$.

Subtask 2 (points: 20)

$1 \leq N, M \leq 100$.

Subtask 3 (points: 30)

Problem statements

$1 \leq N, M \leq 2000$.

Subtask 4 (points: 40)

No additional limitations.

Example

standard input	standard output
5 10	5
5 1 1	
6 10 3	
1 10 1	
10 8 2	
10 4 2	

E. Penalty

Time limit: 2 seconds

Memory limit: 256 megabytes

The Russian national football team is making history – it has reached World Cup quarter-finals, where it will play Croatia. Stanislav Cherchesov, the Russian coach, understands that the only way to win is to take Croatia to penalty shootout. So the whole team is practicing shooting penalties.

In order to choose the best performers for penalty shootout, the coaches have analyzed the statistics for Russian players and the Croatian goalkeeper Danijel Subasic. Based on these statistics, they have made up a *penalty kick map* for all of them. To obtain the map, the area of the goal was divided into $N \times M$ equal rectangles, and for those rectangles they have calculated the probability to score penalty for Russian players and to save the penalty for the keeper.

The coaches now want to combine this data and select the players who have the greatest total area of those rectangles for which the *penalty probability* is not less than 0.65. If there are several players with equal areas, then the one whose name comes lexicographically first is selected.

The “penalty probability” mentioned above for a given rectangle is the product of the probability to score a penalty in it for the player and the probability to concede a penalty in this rectangle for the keeper.

ИТОГИ / FINAL STANDINGS

	Имя /Name	Страна /Country	Первый день / First day					Второй день / Second day					ИТОГО TOTAL	НАГРАДА AWARD
			A	B	C	D	Σ	E	F	G	H	Σ		
1	ROPOVICI Robert-Adrian ПОПОВИЧИ Роберт-Адриан	Romania Румыния	100	100	100	100	400	100	76	100	100	376	776	1DD
2	MUŞAT Tiberiu Ioan МУШАТ Тибериу Иоан	Romania Румыния	100	100	100	100	400	100	46	100	346	746	1DD	
3	DICU Adrian Emanuel ДИКУ Адриан Эмануэль	Romania Румыния	100	100	100	100	400	100	30	100	330	730	1DD	
4	FLORESCU Andreia-Cella ФЛОРЕСКУ Андреа-Селла	Romania Румыния	100	100	100	100	400	100	30	100	270	670	2DD, SA	
5	TINCA Matei ТИНКА Матей	Romania Румыния	100	100	100	100	400	30	30	100	260	660	2DD	
6	BAVBOLOV Temirjan БАЙБОЛОВ Темирлан	Kazakhstan Kazakhstan	100	100	0	60	260	100	64	100	304	564	2DD	
7	AUTKALI Akulbek АЙТКАЛИ Акылбек	Kazakhstan Kazakhstan	100	100	60	30	290	100	30	100	270	560	2DD	
8	TAN Luke ТАН Люк	Singapore Сингапур	100	30	100	30	260	100	30	100	260	520	2DD	
9	LEONG Song Zhu Owen ЛЕНГ Сонг-Чжу Оуэн	Singapore Сингапур	100	30	0	30	160	100	80	100	320	480	3DD	
10	MURATOV Adilkhan МУРАТОВ Адильхан	Kazakhstan Kazakhstan	100	100	0	100	300	100	30	100	170	470	3DD	

для всех j обновляем $dp[i + 1][j] = \max(dp[i][j], dp[i][j - w[i]] + v[i])$. Теперь зафиксируем L , добавляем по одному серверу и останавливаемся, когда $dp[i][C] \leq V$. Такое решение работает за $O(nC)$ для фиксированного L и $O(n^2C)$ в сумме для всех, что проходит первые две подгруппы.

Существует несколько способов улучшить это решение до $O(nC \log n)$ или $O(nC)$. Заметим для начала, что числа в выходном файле не убывают. Это наблюдение позволяет предположить, что в задаче применима техника двух указателей. Для реализации техники двух указателей нам понадобится структура данных, умеющая выполнять следующие операции:

- Добавить сервер с правого конца,
- Удалить сервер с левого конца,
- Узнать, существует ли для текущего множества серверов подмножество со стоимостью $\leq C$ и мощностью $\geq V$.

Такие операции напоминают структуру данных «очередь». Если мы научимся поддерживать такие операции для нашего динамического программирования, то, используя эту структуру данных и технику двух указателей, мы решим задачу.

Известно, что очередь можно проэмулировать с помощью двух стеков. На основе этой реализации, можно в очередь добавить операцию «минимум среди элементов в очереди». Покажем, как таким же образом можно сделать описанную выше структуру данных.

Храним в очереди два стека, каждый из которых хранит строки массива динамического программирования dp для всех элементов, лежащих в стеке под заданным. Удалить элемент из стека легко: просто выбросить вершину стека. Чтобы добавить элемент в стек, нужно сделать переход от одной строки массива dp к следующей. Для того, чтобы проверить, существует ли хорошее подмножество серверов, нам надо «объединить» вершины двух стеков. К счастью, нам интересен только последний элемент этого массива, а его можно найти за время $O(C)$: $\max_{i=0}^C (dp_1[i] + dp_2[C - i])$. Итого, мы научились реализовывать очередь с нужными операциями, каждая из которых работает за $O(C)$. Всего нам нужно сделать $O(n)$ таких операций, поэтому решение работает за $O(nC)$, что вписывается в ограничения по времени.

You are asked to help to Russian team and develop a program that will list the names of 5 best penalty shooters.

Input

The first line contains two integer numbers N, M ($0 < N, M \leq 100$). The next line contains an integer number K ($6 \leq K \leq 100$), the number of team members. The following lines contain “the penalty kick map” for Danijel Subasic. Then come the name (first and last names) and the “penalty kick map” for each player of the Russian team. Each “penalty kick map” is a matrix of size $N \times M$, whose elements a_{ij} ($0 \leq a_{ij} \leq 1$) are real numbers giving with two decimal places the possibility to score penalty for a player or a probability to save a penalty for the goalkeeper.

Output

Program should output names of five team members, that will take a penalty kick. Each name must be printed on a separate line.

Scoring

This problem contains two subproblems. Points will be awarded for a subproblem only if all the tests in it passed. Subproblems are evaluated independently.

Subtask 1 (points: 30)

$N \leq 2, M \leq 2, K \leq 10$.

Subtask 2 (points: 70)

No additional limitations on N and M .

Note

In the probability theory, the sum of probabilities of mutually exclusive events equals one:

$$P(A) + P(\bar{A}) = 1,$$

where A is some event (for example, player scored penalty), \bar{A} — negation of this event (goalkeeper saved penalty).

Example

standard input	standard output
3 3	Alan Dzagoev
6	Alexandr Golovin
0.05 0.90 0.05	Artem Dzyuba
0.95 1.00 0.95	Denis Cheryshev
0.75 1.00 0.75	Mario Fernandes
Alan Dzagoev	
0.85 0.90 0.85	
0.95 1.00 0.95	
0.85 1.00 0.85	
Sergey Ignashevich	
0.87 0.87 0.87	
0.85 1.00 0.85	
0.85 1.00 0.85	
Artem Dzyuba	
0.90 0.90 0.90	
0.90 1.00 0.90	
0.75 1.00 0.75	
Alexandr Golovin	
0.80 0.80 0.80	
0.85 1.00 0.85	
0.75 1.00 0.75	
Denis Cheryshev	
0.85 0.80 0.85	
0.85 1.00 0.85	
0.85 1.00 0.85	
Mario Fernandes	
0.75 0.90 0.75	
0.75 1.00 0.75	
0.55 1.00 0.55	

либо $a_i \neq a_j$ и $b_i \neq b_j$. Такое решение работает за $O(n^2)$.

Для второй подзадачи можно воспользоваться следующим подходом. Для каждой пары кластеров (i, j) посчитаем количество элементов, имеющих такую пару кластеров: $count[i][j]$ — количество таких k , что $a_k = i$ и $b_k = j$. Тогда количество пар, удовлетворяющих первому условию, равно $\sum_{i,j} \frac{count[i][j](count[i][j]-1)}{2}$. Количество пар, удовлетворяющих второму условию, можно посчитать, перебрав две пары кластеров (i_1, j_1) и (i_2, j_2) , где $i_1 \neq i_2$ и $j_1 \neq j_2$. Для всех таких пар к ответу следует добавить $count[i_1][j_1] \cdot count[i_2][j_2]$.

Полное решение выглядит похожим образом. Если интерпретировать пары (a_i, b_i) как точки на плоскости, то решение сводится к тому, чтобы посчитать количество пар точек, которые либо совпадают, либо не находятся на одной горизонтальной, либо на одной вертикальной прямой. Вначале посчитаем такой же массив $count[i][j]$, как было описано выше. Поскольку i и j могут быть порядка 10^5 , использовать массив нельзя, следует использовать аналог **map** в вашем языке программирования, либо сортировку пар (a_i, b_i) . Заодно для каждого кластера в первой и второй кластеризации посчитаем количество объектов, которые определены в этот кластер. Количество пар совпадающих точек считается так же, как и в предыдущей подзадаче, обозначим его за A .

Давайте посчитаем B — количество пар точек, которые лежат на одной горизонтальной прямой. Для этого нужно просуммировать $\frac{x(x-1)}{2}$, где x пробегает размеры кластеров в первой кластеризации. Аналогично C — количество пар различных точек, лежащих на одной вертикальной прямой: это сумма $\frac{y(y-1)}{2}$, где y пробегает размеры кластеров во втором разбиении. Тогда количество пар точек, не лежащих на одной вертикальной и горизонтальной прямой равно $D = \frac{n(n-1)}{2} - B - C + A$, (нужно добавить A , поскольку все такие пары точек мы вычли дважды). Вычисление всех этих формул можно реализовать за $O(n)$, ответ на задачу равен $A + D$.

Н. Параллельные вычисления

Для начала попробуем решить задачу для фиксированного L . Решаем задачу методом динамического программирования: $dp[i][c]$ — максимальная мощность подмножества первых i серверов, со стоимостью не больше c . Переход от $dp[i][*]$ к $dp[i + 1][*]$ может быть сделан за время $O(C)$:

```

for i in range(n):
    keeper.append(list(map(negation, input().split()))))

players = {}
for i in range(players_count):
    player_name = input()
    players[player_name] = 0
    for j in range(n):
        temp = list(map(float, input().split()))
        for l in range(m):
            if keeper[j][l] * temp[l] >= 0.65:
                players[player_name] += 1

players = sorted(players.items(),
                 key=lambda kv: (-kv[1], kv[0]))[:5]

for player in players:
    print(player[0])

```

F. Еще одно необычное уравнение

Рассматривая оценки $x < \sqrt{P}$ и $x > \sqrt{\frac{P}{2}}$, видим, что m либо равен количеству цифр числа \sqrt{P} , либо меньше на единицу. Из уравнения (1), применяя теорему Виета, получим уравнение:

$$X^2 + (m - 1)X - m - P = 0.$$

Приведем это уравнение к виду

$$2x = -m + 1 + \sqrt{4p + (m + 1)^2}.$$

Отсюда видно, что для того, чтобы x было целым, нужно, чтобы было целым $\sqrt{4p + (m + 1)^2}$. Подбираем значение m и находим x .

G. Оценка кластеризации

Для решения первой подзадачи достаточно перебрать все пары $1 \leq i < j \leq n$ и проверить описанное условие: либо $a_i = a_j$ и $b_i = b_j$,

F. Yet another unusual equation

Time limit: 2 seconds
Memory limit: 256 megabytes

Consider the equation

$$X^2 + mX + b - P = 0, \quad (1)$$

where m is the number of decimal digits in the positive root of the equation (1), and b is the negative root of the equation (1).

For a given integer P , find the positive integer root of the equation (1).

Input

A single integer number P ($0 < P < 10^{21000}$).

Output

Output the positive integer root of the equation (1). If the equation has no such root, output **-1**.

Scoring

This problem contains four subtasks. Points for the first three subtasks are awarded only if solution passes all the tests from this subtask. Points for each test of the last subtask are awarded independently. The subtasks are evaluated independently.

Subtask 1 (points: 10)

$P < 10^5$.

Subtask 2 (points: 20)

$P < 10^{10}$.

Subtask 3 (points: 30)

$P < 10^{90}$.

Subtask 4 (points: 40)

No additional limitations.

Example

standard input	standard output
208	14

G. Clustering evaluation

Time limit: 1 second
Memory limit: 256 megabytes

Clusterization is one of the most important aspects of now popular machine learning. Clusterization is the task of grouping the set of n objects into m sets (called *clusters*), so that objects in the same groups are similar to one another (for example, they are close in sense of some metric, or any other similarity measures).

Marina came up with a new clustering algorithm. To figure out whether her algorithm is working well, Marina run it on some data, for which she knows the reference clustering. Help Marina to calculate the number of pairs of objects that are classified correctly by her algorithm. We consider the pair of objects to be classified correct in one of two cases: either both of these objects are in the same cluster in both clusterings, or these two objects are in different clusters in both clusterings.

Input

First line contains two integers n and m ($1 \leq m \leq n \leq 10^5$) — the number of objects and the number of clusters. Second line contains the description of reference clustering: n integers a_i ($1 \leq a_i \leq m$) — number of set of i -th object in reference clustering. Third line contains the description of clustering produced by Marina's algorithm: n integers b_i ($1 \leq b_i \leq m$) — number of set of i -th object in Marina's clustering.

Output

Output one integer — the number of pairs of correctly classified objects.

Scoring

This problem contains three subtasks. Points for a subtask are awarded only if solution passes all the tests from this subtask and preceding subtasks.

Subtask 1 (points: 30)

$n \leq 1000$.

Subtask 2 (points: 40)

$n \leq 10^5, m \leq 10$.

Subtask 3 (points: 30)

No additional limitations.

сканирующей (заметающей) прямой (sweep line), расположенной вертикально и двигающейся вправо. Нас будут интересовать события (events, event points) двух типов: начала и концы квадратных областей. Сканирующая прямая будет представлять из себя прямоугольник $1 \times M$. Возьмем массив из M элементов, элементы которого будут соответствовать клеткам прямой. В массиве для каждой клетки будем хранить количество вулканов, которые покрывают эту клетку на данный момент.

Если случается событие «начало квадрата», то надо увеличить на единицу значение элементов массива на отрезке от верхней границы квадрата до нижней (или наоборот, в зависимости от реализации). В случае «конец квадрата» нужно отнять по единице. Для каждой позиции с событиями находим минимум в массиве. Если минимум будет равен 0, то это значит, что существует не затопленная клетка. Если все минимумы будут строго больше 0, значит, остров полностью затоплен.

Если реализовать это наивно, решение будет работать за $O(nm \log m)$, и это позволит решить третью подзадачу. Наконец, если для заполнения массива и поиска в нём минимума использовать дерево отрезков с обновлением на отрезке (segment tree), то решение будет работать за $O(n \log n \log m)$,

Е. Пенальти

Решение задачи предполагает сортировку игроков по количеству областей, в которых вероятность забить пенальти не менее 0.65, а затем по имени. Вычислять общую площадь нет необходимости, так как прямоугольники одинаковые для всех. Для каждой области вероятность забить пенальти вычисляется как произведение вероятности забить пенальти в этой области для игрока и вероятности пропустить пенальти в этой области для вратаря (то есть $1 -$ вероятность отбить пенальти). Один из вариантов решения задачи приведен ниже (на языке Питон):

```
def negation(p):
    return 1 - float(p)

n, m = map(int, input().split())
players_count = int(input())
keeper= []
```

ко известный алгоритм, основанный на двух поисках в глубину: по прямым ребрам и по обратным ребрам, первый из которых присваивает вершинам «время выхода», а второй запускается в порядке уменьшения этих времен и при каждом запуске выделяет очередную компоненту сильной связности.

В завершение вычисляем количество истоков и стоков в графе конденсации, т.е. количество компонент сильной связности, в вершины которых не ведет ни одного ребра из вершин других компонент и количество компонент, из вершин которых не ведет ни одного ребра в вершины других компонент. Ответом на задачу будет максимум из этих двух количеств. Данный факт легко доказывается, например, по индукции.

Также следует отдельно рассмотреть два частных случая, не покрываемых описанным алгоритмом:

- Если количество компонент сильной связности равно 1, т.е. конденсация состоит из единственной вершины, то ответ на задачу равен нулю.
- Если исходный набор содержит единственное слово, то вне зависимости от того, совпадает ли его первая буква с последней ответ на задачу будет равен нулю. Однако, если исходный набор состоит из двух или более слов, даже если они задают два кратных ребра, то этот случай следует рассматривать в общем порядке.

D. Вулканы

Для решения первой подзадачи достаточно рассмотреть четыре угла острова. Один из них будет затоплен одним из последних. Вторая подзадача решается перебором. Нужно рассмотреть все точки на острове. Для каждой точки вычисляется последний день перед затоплением. Из вычисленных значений выбирается максимальное. Это и будет ответом. Но такой подход не проходит по времени для остальных подзадач.

Для решения оставшихся подзадач можно использовать двоичный (бинарный) поиск (binary search). Ищем такой день d , в котором еще существует не затопленная клетка, а в день $d + 1$ весь остров затоплен.

Теперь остается научиться быстро определять, существуют ли не затопленные места в заданный день. Для этого воспользуемся методом

Examples

standard input	standard output
5 3 1 2 3 1 3 2 1 1 2 1	8
4 3 1 2 3 2 3 1 2 1	6
4 4 1 2 3 4 1 1 1 1	0

Note

In the first sample there are two incorrectly classified pairs: (2, 3) and (2, 5). All other $\frac{5 \cdot 4}{2} - 2 = 8$ pairs are classified correctly.

H. Parallel computing

Time limit: 1 second
Memory limit: 256 megabytes

In addition to participating in «Tuymaada», your team also decided to take part in a hackathon. To train a neural network, your team will need a lot of computational power, so you decided to contact the nearest data center.

The data center has n servers, ready to be rented. Servers are arranged in a row and are enumerated from 1 to n from left to right. It costs c_i rubles to rent the i -th server, and its computational power is equal to p_i . You have C rubles and you must rent servers having at least P units of computational power. Due to the data center rules, you have to provide the cables yourself, so you would like to make the distance between the first rented server and the last rented server as small as possible.

You still haven't decided which servers you are going to rent, so for each possible first server you would like to find the minimum possible last server to minimize the length of cables you have to bring. Formally, for each L from 1 to n you need to find the minimum R such that you can select some servers from segments $[L, R]$ so that their total cost is no more than C rubles and they have at least P units of computational power in total.

Input

The first line contains integers n, C and P ($1 \leq n, C, P; n \leq 10^5, nC \leq 10^6, P \leq 10^{18}$) – the number of servers, your maximum budget in rubles and required total computational power. Next line contains n integers c_i ($1 \leq c_i \leq C$) – costs of server rent. Next line contains n integers p_i ($1 \leq p_i \leq P$) – computational powers of servers.

Output

Output n integers, the minimum possible values of R for $L = 1, 2, \dots, n$. If there is no suitable R , output -1 .

Scoring

This problem contains three subtasks. Points for a subtask are awarded only if solution passes all the tests from this subtask and preceding subtasks.

Subtask 1 (points: 30)

$n, C, P \leq 20$.

Subtask 2 (points: 10)

$n, C \leq 500$.

Subtask 3 (points: 60)

No additional limitations.

Example

standard input	standard output
7 12 20	1 4 5 7 -1 -1 -1
10 3 6 1 5 10 5	
20 10 8 2 10 3 8	

Например для $N = 32241$ число знаков равно 5, и возможны следующие его разложения в сумму: $1 + 4 = 2 + 3 = 5$.

Повторяя каждую цифру d в каждом множестве по d раз, получим наборы цифр, из которых можно составить нужной длины записи чисел. В нашем примере это будут $(1, 4, 4, 4, 4)$, $(2, 2, 3, 3, 3)$ и $(5, 5, 5, 5, 5)$.

Для поиска минимально возможного числа, превосходящего номер сегодняшнего дня N , заметим следующее. Число $K = \overline{b_0 b_1 \dots b_n}$ больше числа $N = \overline{a_0 a_1 \dots a_n}$, если найдется $i \geq 0$ такое, что начальные i цифр в обоих числах одинаковые, а $b_i > a_i$. При этом чем правее эта позиция, то есть чем больше i , тем меньше число K . Оставшиеся на позициях справа от i цифры нужно расположить в возрастающем порядке.

Для наборов рассмотренного примера получаем значения $i = 2, 0, 0$ и числа $K = 32322, 41444, 55555$.

Таким образом, для решения задачи достаточно перебрать все допустимые наборы цифр длины n , определяя для каждого набора максимальное подходящее i , если оно существует, и запоминая минимум из построенных таким образом чисел K . Если для n ни одного подходящего набора не нашлось, то достаточно ограничиться первым в лексикографическом порядке набором из $n + 1$ цифры. Например, для $N = 55556$ ответ будет $K = 122333$.

С. Игра в слова

Для начала заметим, что условие о запрете повторения слов является избыточным, поскольку достижимость одного слова от другого от этого условия не зависит.

Построим граф, вершинами которого будут буквы алфавита, а ребрами – слова исходного набора: для слова, начинающегося с буквы a и заканчивающегося на букву b , проведем ребро из вершины a в вершину b . При этом возможны кратные ребра и вершины с петлями, которые следует отличать от изолированных вершин без петель. Последние, т.е. буквы, на которые не начинаются и не заканчиваются никакие слова исходного набора, имеет смысл сразу исключить из рассмотрения, а оставшиеся вершины перенумеровать.

Далее выделяем в получившемся графе компоненты сильной связности и строим конденсацию графа. Для этого можно применить широ-

Решения

А. Либербал

В условии описывалась перестановка, которую следовало к двумерному массиву размера $n \times m$ применить k раз. Первая подзадача на 50 баллов решалась следующим простым алгоритмом: k раз выполним один шаг танца. Для этого создадим два массива, первый содержит номера танцоров на позициях перед шагом, а во второй запишем номера танцоров после сделанного движения танца. Обозначим эти массивы old и new . Затем, переберем все клетки (i, j) в прямоугольнике (i от 1 до n , j от 1 до m), и посмотрим, куда переходит танцор в клетке (i, j) (обозначим эту клетку (p, q)) и переставим танцора во второй массив new : $new[p][q] = old[i][j]$. Сделаем так для всех клеток (i, j) . Этот шаг займет порядка $O(nm)$ времени. После этого забудем про массив new и сделаем следующий шаг танца аналогично. Повторив этот алгоритм k раз, выводим конечный массив. Такое решение работает за $O(nmk)$, где $nmk \leq 10^6$.

Для того, чтобы получить 100 баллов, следовало воспользоваться тем, что наше движение задает перестановку. Как известно, любую перестановку можно разбить на набор циклов. Внутри каждого цикла один шаг танца выглядит просто как циклический сдвиг номеров танцоров на 1. Соответственно, k шагов танца это циклический сдвиг массива номеров танцоров в этом списке на $k \bmod l$, где l — длина цикла. Осталась реализационная часть: в перестановке, заданной во входных данных, следует выделить циклы, записать номера танцоров на этом цикле в массив, сдвинуть этот массив циклически на $k \bmod l$, затем записать номера танцоров обратно в двумерный массив. Эффективно выделить циклы в перестановке можно за $O(nm)$, после этого каждого танцора рассмотрим в решении ровно один раз, поэтому финальное время работы этого решения есть $O(nm)$.

В. Празднество Звезд на планете Y

Сначала определим множества цифр, которые могут входить в номер искомого дня: либо простым перебором (число всевозможных множеств из положительных цифр 2^9), либо раскладывая требуемое число знаков в сумму возрастающих натуральных слагаемых, не превосходящих 9.

Solutions

A. Liberdance

Problem statement described a permutation which was applied to a two-dimensional array of size $n \times m$ k times. First subtask for 50 points can be solved with a simple simulation: just simulate one dance movement k times. To do that, create two array, the first one containing the dancers' positions before the movement, and the second array containing the numbers after the movement. Let's call these arrays old and new . Next, iterate over all cells (i, j) in the rectangle (i ranges from 1 to n , j ranges from 1 to m), look where does the dancer go from (i, j) (let's call this cell (p, q)) and move the dancer into the second array new : $new[p][q] = old[i][j]$. Do this assignment for all cells (i, j) . This step takes $O(nm)$ time. After that we forget about the old array and repeat the same step $k - 1$ times. This solutions works in $O(nmk)$ time, where $nmk \leq 10^6$, so this is fast enough.

To get 100 points, you should use the fact that the movement is a permutation. It is known that every permutation can be represented as a set of cycles. In each cycle one dance movement is just a cyclic shift by 1. In the same manner, k movements is a cyclic shift by $k \bmod l$, where l is the cycle length. Now, the solution outline looks like this: take the input permutations, decompose it into cycles, write down the dancers' numbers into the array, shift this array cyclically by $k \bmod l$, then write the dancers' numbers back into the two-dimensional array. It is possible to identify all cycles efficiently in $O(nm)$ time. Every dancer will be moved into its place exactly once, so the time complexity of this solution is $O(nm)$.

B. Festival of the Stars

Let N have n digits. Consider all the multisets K of digits of given length n where each digits d repeats d times. To find the minimum number greater than N composed from the digits in K , one should try to make the first position where the numbers differ as far to the right as possible. If a number greater than N cannot be made from n digits, take lexicographically first number with $(n + 1)$ digits.

C. A Game of Words

Notice that the condition banning word repetition is unnecessary, because it doesn't affect word reachability.

To solve this problem, build a graph with vertices marked by the letters and edges marked by the words read from the input data. For every word starting at the letter a and ending at the letter b create an edge connecting the vertices marked by letters a and b . Notice that this graph may have multiple edges and loops.

The isolated vertices without loops that are not connected by any edge should be excluded from this graph.

Next, we need to find strongly connected components, i.e. maximal subgraphs, such that any pair of its vertices are reachable from each other in both directions. There is a well known algorithm, based on two depth-first searches: by forward and backward directed edges. First of them marks visited vertices by the "Time of exit". Let's run the second DFS for every marked vertex, sorted by the "Time of exit" in descending order. The latter DFS finds a strongly connected component at each run.

In the final part of the solution we contract each strongly connected component to a single vertex and obtain a condensation graph, which is an acyclic directed graph. Let's find the number of sources and sinks in this condensation graph, i.e. the number of strongly connected components that

- don't have edges directed to it from other components (sources);
- don't have edges directed from it to other components (sinks).

The maximum of this two numbers is an answer of this problem. This statement may be proved using induction.

Finally two special cases are not covered above:

- If there is only one strongly connected component, i.e. the condensation graph consist of a single vertex, then the answer to the problem is zero.
- If input data consist of only one word, then the answer of problem is zero. It does not matter if first letter of this word equals to last letter. In case if input data consist of two or more words we should use the general algorithm even if they represent multiple edges.

Вы еще не определились, какие серверы вы собираетесь арендовать, поэтому для каждого первого арендованного сервера вам стоит найти минимальный последний, чтобы минимизировать длину кабелей, их соединяющих. Более формально, для всех L от 1 до n вам нужно найти минимальное R , такое что на отрезке серверов $[L, R]$ вы можете арендовать некоторые серверы на сумму не более C и суммарной вычислительной мощностью не менее P .

Формат входных данных

В первой строке содержатся числа n, C и P ($1 \leq n, C, P; n \leq 10^5, nC \leq 10^6, P \leq 10^{18}$) — количество серверов, максимальный бюджет в рублях и минимальная вычислительная мощность. В следующей строке содержатся n чисел c_i ($1 \leq c_i \leq C$) — стоимости аренды серверов. В следующей строке содержатся n чисел p_i ($1 \leq p_i \leq P$) — вычислительные мощности серверов.

Формат выходных данных

Выведите n целых чисел, минимально возможные R для $L = 1, 2, \dots, n$. Если подходящего R не существует, выведите -1 .

Система оценивания

Данная задача содержит три подзадачи. Баллы за подзадачу начисляются только если все тесты этой подзадачи и всех предыдущих подзадач пройдены.

Подзадача 1 (баллов: 30)

$n, C, P \leq 20$.

Подзадача 2 (баллов: 10)

$n, C \leq 500$.

Подзадача 3 (баллов: 60)

Нет дополнительных ограничений.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
7 12 20 10 3 6 1 5 10 5 20 10 8 2 10 3 8	1 4 5 7 -1 -1 -1

$n \leq 10^5, m \leq 10$.

Подзадача 3 (баллов: 30)

Без дополнительных ограничений.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
5 3 1 2 3 1 3 2 1 1 2 1	8
4 3 1 2 3 2 3 1 2 1	6
4 4 1 2 3 4 1 1 1 1	0

Замечание

В первом примере существует две пары некорректно определенных объектов: (2, 3) и (2, 5). Все остальные $\frac{5 \cdot 4}{2} - 2 = 8$ пар определены корректно.

N. Параллельные вычисления

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Помимо участия в олимпиаде «Туймаада», ваша команда также решила поучаствовать в хакатоне. Для тренировки нейронной сети вам непременно понадобится много вычислительных мощностей, поэтому вы решили обратиться к представителям ближайшего дата-центра.

В дата-центре находится n серверов, которые можно арендовать. Серверы расположены в ряд и пронумерованы от 1 до n слева направо. Стоимость аренды i -го из них равна c_i рублей, а его вычислительная мощность равна p_i . У вас есть S рублей, и вам необходимо арендовать суммарно как минимум P единиц вычислительной мощности. По правилам датацентра, кабели вы предоставляете сами, поэтому вы бы хотели минимизировать расстояние между первым и последним арендованным сервером.

D. Volcanoes

We can find the last day before flooding for each cell, and then select the maximum. This suffices for the first two subproblems (enough to consider only the four corner cells for the first subproblem), but is too slow for the remaining subproblems.

A binary search for the day d such that there are non-flooded cells on day d but not on day $d + 1$ may be used.

We now need to quickly determine whether there are non-flooded cells on a given day. Use a sweeping line method. A vertical line scans from left to right and is represented by an $1 \times M$ array. Each element of this array is the number of volcanoes that cover the corresponding cell at that moment. We are interested in events of two types, the start of a square area and an end of a square area.

For a *square start* event we need to increase by one each array element from the top edge of the square to the bottom edge (or vice versa, depending on the implementation). For a *square end* area we need to subtract 1 from each. For each position with events, find the minimum in the array. A minimum of 0 means that there is a non-flooded cell. If all minimums are positive, it means that the island is completely flooded.

A naive implementation will run in $O(nm \log m)$ which allows to solve the third subproblem. Finally, using a segment tree to update and find minimums in the array will allow an $O(n \log n \log m)$ solution to solve the fourth subproblem.

E. Penalty

To solve this problem one doesn't need to calculate area. Since all small rectangles have same area, you just need to count the number of rectangles having "penalty probability" not less than 0.65. Next you need to sort the players by number of such rectangles and their names.

For each region the "penalty probability" is calculated as the product of the probability to score penalty by the player and the probability concede the penalty for the keeper (which is simply $1 - \text{probability to save penalty}$).

One of the possible solutions (in Python) follows:

```
def negation(p):
```

```

    return 1 - float(p)

n, m = map(int, input().split())
players_count = int(input())
keeper = []
for i in range(n):
    keeper.append(list(map(negation, input().split())))

players = {}
for i in range(players_count):
    player_name = input()
    players[player_name] = 0
    for j in range(n):
        temp = list(map(float, input().split()))
        for l in range(m):
            if keeper[j][l] * temp[l] >= 0.65:
                players[player_name] += 1

players = sorted(players.items(),
                 key=lambda kv: (-kv[1], kv[0]))[:5]

for player in players:
    print(player[0])

```

F. Yet another unusual equation

Taking into account the bounds $x < \sqrt{P}$ and $x > \sqrt{\frac{P}{2}}$, we see that m is either equal to the number of digits of \sqrt{P} or one less.

From the initial equation (1), using the Vieta theorem, we obtain:

$$x^2 + (m - 1)x - m - P = 0.$$

Rewriting, we get

$$2x = -m + 1 + \sqrt{4p + (m + 1)^2}.$$

We see that for x to be integral, we need the number $\sqrt{4p + (m + 1)^2}$ to be integral. Finding a suitable m allows us to find x .

G. Оценка кластеризации

Ограничение по времени: 1 секунда
 Ограничение по памяти: 256 мегабайт

Кластеризация является важнейшей областью ныне популярного машинного обучения. Задача кластеризации формулируется следующим образом: дано n объектов, следует разбить их на m множеств (которые называют *кластерами*) по определенному признаку (например, по расстоянию или каким-либо общим чертам).

Марина придумала новый метод кластеризации. Для того, чтобы понять, насколько хорошо он работает, Марина запустила его на данных, для которых ей известно эталонное разбиение на кластеры. Помогите Марине посчитать количество пар объектов, которые ее алгоритм определил корректно. Пара объектов является корректно определенной в двух случаях: либо оба объекта находятся в одном кластере в обоих разбиениях, либо в обоих разбиениях эти объекты находятся в разных кластерах.

Формат входных данных

В первой строке содержатся числа n и m ($1 \leq m \leq n \leq 10^5$) — количество объектов и кластеров, соответственно. Вторая строка содержит описание эталонной кластеризации: n чисел a_i ($1 \leq a_i \leq m$) — номер, присвоенный i -му объекту в эталонной кластеризации. Третья строка содержит описание кластеризации, выданной алгоритмом Марины: n чисел b_i ($1 \leq b_i \leq m$) — номер, присвоенный i -му объекту в результате исполнения метода Марины.

Формат выходных данных

Выведите одно число — количество пар корректно определенных объектов.

Система оценивания

Данная задача содержит три подзадачи. Баллы за подзадачу начисляются только если все тесты этой подзадачи пройдены. Подзадачи оцениваются независимо.

Подзадача 1 (баллов: 30)

$n \leq 1000$.

Подзадача 2 (баллов: 40)

F. Еще одно необычное уравнение

Ограничение по времени: 2 секунды
 Ограничение по памяти: 256 мегабайт

Рассмотрим уравнение

$$X^2 + mX + b - P = 0, \quad (1)$$

где m совпадает с количеством десятичных цифр положительного корня уравнения (1), а число b — отрицательный корень уравнения (1). Требуется для заданного целого P найти целый положительный корень уравнения (1).

Формат входных данных

Одно целое число P ($0 < P < 10^{21000}$).

Формат выходных данных

Выходной файл должен содержать положительный целый корень уравнения (1). Если целого положительного корня у уравнения (1) нет, то печатается -1 .

Система оценивания

Данная задача содержит четыре подзадачи. Баллы за первые три подзадачи начисляются только если все тесты данной подзадачи пройдены. Баллы за тесты четвертой подзадачи начисляются независимо. Подзадачи оцениваются независимо друг от друга.

Подзадача 1 (баллов: 10)

$P < 10^5$.

Подзадача 2 (баллов: 20)

$P < 10^{10}$.

Подзадача 3 (баллов: 30)

$P < 10^{90}$.

Подзадача 4 (баллов: 40)

Нет дополнительных ограничений.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
208	14

G. Clustering evaluation

The first subtask can be solved by simply iterating over all pairs $1 \leq i < j \leq n$ and checking the condition in the problem statement: either $a_i = a_j$ and $b_i = b_j$, or $a_i \neq a_j$ and $b_i \neq b_j$. This solution works in $O(n^2)$.

For the second subtask you can use the following approach. For each pair of clusters (i, j) calculate the number of elements having this pair: $count[i][j]$ — number of k that have $a_k = i$ and $b_k = j$. Then the number of pairs, satisfying the first condition is equal to $\sum_{i,j} \frac{count[i][j](count[i][j]-1)}{2}$. Number of pairs satisfying the second condition can be calculated, iterating over two pairs of clusters (i_1, j_1) and (i_2, j_2) , where $i_1 \neq i_2$ и $j_1 \neq j_2$. For all such pairs, add $count[i_1][j_1] \cdot count[i_2][j_2]$ to the answer.

Full solution for this problem goes as follows. If we view pairs (a_i, b_i) as points on a 2D plane, then the answer to this problem is the number of pairs of points that are either equal, or not lie on the same vertical or horizontal line. First, let's calculate the array $count[i][j]$, the same as above. Because i and j can be up to 10^5 , you can't use arrays, so use either the analogue of **map** in your programming language, or sort pairs (a_i, b_i) . Also, for each cluster in the first and second clusterization calculate the number of objects in that cluster. Number of equal points can be calculated the same as in previous subtask. Let's call this number A .

Let's calculate B — number of pairs of points that lie on the same vertical line. It is equal to sum of $\frac{x(x-1)}{2}$, where x goes over all sizes of clusters in the first clusterization. Likewise, define C as the number of pairs of points on the same vertical line: it is equal to sum of $\frac{y(y-1)}{2}$, where y goes over all sizes of clusters in the second clusterization. Then number of points not on the same vertical or horizontal line is equal to $D = \frac{n(n-1)}{2} - B - C + A$, (we add A back because we subtracted pairs equal points twice). All these calculations can be done in $O(n)$, output $A + D$ after that.

H. Parallel computing

First, let's try solving the problem for a fixed L . We will solve this using dynamic programming: $dp[i][c]$ is the maximum total power, if we consider some subset of first i servers, having cost $\leq c$. Transition from $dp[i][*]$ to $dp[i+1][*]$ can be done in $O(C)$ time: for all j relax the DP state

$dp[i + 1][j] = \max(dp[i][j], dp[i][j] - w[i] + v[i])$. Now, fix L , add servers one by one and stop when we have $dp[i][C] \leq V$. This solution works in $O(nC)$ time for a fixed L and in $O(n^2C)$ time in total, which should pass first two subtasks.

There are multiple ways to optimize this solution to $O(nC \log n)$ or $O(nC)$ time. Notice that answers in the output are non-decreasing. This allows us to guess that the two pointers technique can be applied to this problem. For the two pointers technique we would need a data structure with following operations:

- Add the server to the right,
- Remove the leftmost server,
- Check whether there is a subset of servers with cost $\leq C$ and power $\geq V$.

These operations resemble the queue data structure. If we manage to support these operations and calculate DP values at the same time, then we should be able to solve this problem.

It is known that a queue can be simulated using two stacks. Based on this implementation, for example, we can add a «find a minimum element in the queue» operation. Let's show how we implement the data structure described above in the same manner.

The queue contains two stack, each one of them storing rows of DP values dp for all servers stored under the current element in that stack. It's easy to remove the top element: just discard the top of the stack and we are done. To add an element on top of the stack, we have to perform the DP transition similarly as above. To check whether there is a good subset of servers, we have to «unite» DP rows on top of two stacks. Luckily, we are only interested in the last element of this array, and it can be found in $O(C)$ time: $\max_{i=0}^C (dp_1[i] + dp_2[C - i])$. Finally, we are able to implement a queue with required operations, each operation works in $O(C)$ time. The two pointers method required $O(n)$ such operations, so the whole solution works in $O(nC)$ time, which fits the time limit.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
3 3	Alan Dzagoev
6	Alexandr Golovin
0.05 0.90 0.05	Artem Dzyuba
0.95 1.00 0.95	Denis Cheryshev
0.75 1.00 0.75	Mario Fernandes
Alan Dzagoev	
0.85 0.90 0.85	
0.95 1.00 0.95	
0.85 1.00 0.85	
Sergey Ignashevich	
0.87 0.87 0.87	
0.85 1.00 0.85	
0.85 1.00 0.85	
Artem Dzyuba	
0.90 0.90 0.90	
0.90 1.00 0.90	
0.75 1.00 0.75	
Alexandr Golovin	
0.80 0.80 0.80	
0.85 1.00 0.85	
0.75 1.00 0.75	
Denis Cheryshev	
0.85 0.80 0.85	
0.85 1.00 0.85	
0.85 1.00 0.85	
Mario Fernandes	
0.75 0.90 0.75	
0.75 1.00 0.75	
0.55 1.00 0.55	

Формат выходных данных

Программа должна вывести имена пятерых игроков сборной России, которые будут бить пенальти. Каждое имя должно начинаться с новой строки.

Система оценивания

Данная задача содержит две подзадачи. Баллы за подзадачу начисляются только если все тесты этой подзадачи пройдены. Подзадачи оцениваются независимо.

Подзадача 1 (баллов: 30)
$$N \leq 2, M \leq 2, K \leq 10.$$
Подзадача 2 (баллов: 70)

Без дополнительных ограничений.

Замечание

В теории вероятностей сумма вероятностей взаимоисключаемых событий равна 1.

$$P(A) + P(\bar{A}) = 1,$$

где A — некоторое событие (например, игрок забил пенальти), \bar{A} — отрицание этого события (вратарь отбил пенальти)

Задачи**А. Либербал**

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

На планете Либерлун очень любят танцевать. Танцевальный зал главного дворца Либерлуна представляет собой прямоугольник $h \times w$, где h — высота, w — ширина прямоугольника. Клетки прямоугольника пронумерованы числами от 1 до hw как показано на рисунке.

1	2	...	w
$w + 1$	$w + 2$...	$2w$
...
$(h - 1)w + 1$	hw

Танец очень прост и состоит только из одного действия, которое повторяется k шагов: вначале все танцоры стоят в каждой клетке прямоугольника. Затем в результате шага каждый из них переходит в соседнюю клетку согласно направлению, написанному на самой клетке и не зависящему от танцора, который в этой клетке находится.

В главном дворце Либерлуна готовятся к церемонии закрытия Либерлунских интеллектуальных игр. Вас попросили помочь в этом непростом деле. Покажите танцорам, где каждый из них будет стоять после k шагов танца.

Формат входных данных

В первой строке содержится 3 целых числа: h , w и k ($1 \leq h$, $w \leq 100$, $0 \leq k \leq 10^9$, hw чётно) — размеры танцевального зала и число повторов соответственно. Следующие h строк содержат описание движений танца. Каждая строка состоит из w символов «L», «R», «D» или «U», соответствующих направлениям влево, вправо, вниз и вверх. Направления из каждой клетки ведут в другую клетку зала, а также в каждую клетку можно прийти из ровно одной другой.

Формат выходных данных

Выведите конечные позиции танцоров в следующем формате: в каждой из h строк должны содержаться w номеров танцоров, стоящих в соответствующих местах.

Система оценивания

Данная задача содержит две подзадачи. Баллы за подзадачу начисляются только если все тесты этой подзадачи и всех предыдущих подзадач пройдены.

Подзадача 1 (баллов: 50)

$k \leq 100$.

Подзадача 2 (баллов: 50)

Без дополнительных ограничений.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
3 4 3	10 9 4 8
DLRD	6 5 3 7
DUUL	2 1 12 11
RURL	

В. Празднество Звезд на планете Ы

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Ваш космический корабль получил повреждения и требует ремонта. Ближайшая планета, на которой можно провести ремонт, называется Ы. Однако в данный момент на планете объявлен сезон очищения, во время которого Ыанцы не принимают корабли из внешнего мира. Известно, что космопорты планеты вновь откроются только в день ближайшего следующего празднества Звезд.

Древняя культура планеты Ы нумерует дни подряд «от сотворения мира». Празднества Звезд устраиваются в такие дни, что каждая входящая в номер дня цифра k повторяется в нем ровно k раз. Например в 333212-й день состоится празднество Звезд.

Определите, когда откроются космопорты планеты Ы, зная номер сегодняшнего дня по Ыанскому календарю.

Формат входных данных

Одно целое число N ($0 < N < 10^{45}$) — номер сегодняшнего дня по Ыанскому календарю.

Е. Пенальти

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На наших глазах вершится футбольная история. Сборная Российской Федерации по футболу впервые преодолела групповой этап и 1/8 чемпионата мира по футболу. Но в 1/4 финала соперником футболистов России является сборная Хорватии. Тренер сборной Станислав Черчесов понимает, что единственный шанс победить — это серия послематчевых пенальти. Поэтому команда тренируется бить пенальти.

Для того, чтобы выбрать наиболее подходящих игроков для серии пенальти было решено составить «карту пенальти» для всех игроков сборной и голкипера сборной Хорватии Даниела Субашича. Для этого ворота, представляющие собой прямоугольник, разделили на $N \times M$ равных прямоугольников. Для каждой из этих областей по данным статистики матчей вычислили вероятности попасть в эту область для игроков и отбить мяч для вратаря.

Тренерский штаб решил совместить эту информацию и выбрать тех игроков, у которых наибольшая площадь областей, где вероятность забить пенальти не менее 0.65. Если у нескольких игроков эта площадь одинаковая, то должны быть выбраны те, имена которых стоят раньше в лексикографическом списке.

Для области вероятность забить пенальти — это произведение вероятностей забить гол игроком и пропустить мяч вратарем.

Разработайте программу, которая поможет тренерам подобрать пятерых игроков, которые будут бить пенальти в игре с командой Хорватии.

Формат входных данных

На вход программе даются целые числа N и M ($0 < N, M \leq 100$). В следующей строке дается целое число K ($6 \leq K \leq 100$) — число игроков в сборной. Далее приводится описание «карты пенальти» голкипера сборной Хорватии Даниела Субашича, а затем перечисляются имена игроков сборной (имя и фамилия) и их «карты пенальти».

«Карта пенальти» представляет собой матрицу $N \times M$, элементы матрицы a_{ij} ($0 \leq a_{ij} \leq 1$) — десятичные числа с двумя знаками после запятой — вероятности либо отбить пенальти для вратаря, либо попасть в эту область для игрока.

Формат входных данных

В первой строке находятся 2 целых числа: N и M ($1 \leq N, M \leq 150\,000$) — количество вулканов и размер острова. Следующие N строк содержат параметры вулканов. Каждая строка содержит 3 целых числа: x_i, y_i, t_i ($1 \leq x_i, y_i \leq M, 1 \leq t_i \leq 150\,000$) — координаты места и день начала извержения очередного вулкана. Место начала извержения у всех вулканов разное. Гарантируется, что можно разместить станцию так, что она проработает хотя бы один день.

Формат выходных данных

Выведите одно целое число — максимальное количество дней, которое может проработать станция.

Система оценивания

Данная задача содержит четыре подзадачи. Баллы за подзадачу начисляются только если все тесты этой подзадачи пройдены. Подзадачи оцениваются независимо.

Подзадача 1 (баллов: 10)

$N = 1$.

Подзадача 2 (баллов: 20)

$1 \leq N, M \leq 100$.

Подзадача 3 (баллов: 30)

$1 \leq N, M \leq 2\,000$.

Подзадача 4 (баллов: 40)

Без дополнительных ограничений.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
5 10	5
5 1 1	
6 10 3	
1 10 1	
10 8 2	
10 4 2	

Формат выходных данных

Выведите единственное целое число — номер дня, в который состоится ближайшее следующее празднество Звезд.

Система оценивания

Данная задача содержит две подзадачи. Баллы за подзадачу начисляются только в том случае, если все тесты этой подзадачи пройдены. Подзадачи оцениваются независимо.

Подзадача 1 (баллов: 30)

Во всех тестах данной подзадачи N не превосходит 2^{31} .

Подзадача 2 (баллов: 70)

Дополнительных ограничений на N нет.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
19	22
22	122

С. Игра в слова

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Многим известна игра в «слова», в которой несколько игроков по очереди говорят слова из некоторого набора таким образом, чтобы первая буква очередного слова совпадала с последней буквой предыдущего слова, причем слова повторяться не могут. Первое слово может быть любым. В качестве наборов слов часто используются, например, названия городов, животных или растений.

Вася очень любит эту игру, однако заметил, что иногда, начав игру на одном из слов, некоторые другие слова из набора никогда не могут быть названы в течение игры вне зависимости от действий игроков. Теперь Васю интересует, какое минимальное количество новых слов необходимо добавить к исходному набору для того, чтобы при любом выборе двух слов (начального и целевого), игроки могли бы говорить слова в таком порядке, чтобы целевое слово встретилось в игре.

Формат входных данных

В первой строке находится два целых числа: N — количество букв в алфавите и M ($1 \leq N, M \leq 100\,000$) — количество слов в исходном наборе. В следующих M строках описываются слова, входящие в исходный набор. В i -й строке даны два числа: номер первой и последней буквы i -го слова. Буквы в алфавите нумеруются с единицы.

Формат выходных данных

Выведите единственное число — минимальное количество новых слов, которое необходимо добавить в набор.

Система оценивания

Данная задача содержит три подзадачи. Баллы за подзадачу начисляются только в том случае, если все тесты этой подзадачи пройдены. Подзадачи оцениваются независимо.

Подзадача 1 (баллов: 30)

Во всех тестах данной подзадачи N и M не превосходят 20.

Подзадача 2 (баллов: 30)

Дополнительных ограничений на N и M нет. Гарантируется, что для каждой буквы алфавита существует слово, которое начинается и/или заканчивается на эту букву.

Подзадача 3 (баллов: 40)

Дополнительных ограничений нет.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
9 11	2
1 2	
2 3	
3 1	
4 5	
5 6	
6 4	
7 8	
8 9	
9 7	
1 4	
1 7	

D. Вулканы

Ограничение по времени: 10 секунд
Ограничение по памяти: 256 мегабайт

Ученые долгие годы проводили наблюдения над вулканами на одном острове. Остров представляет из себя квадрат $M \times M$ клеток, строки и столбцы которого нумеруются от 1 до M .

У каждого вулкана есть два важных параметра: место и день начала извержения. Вулканы извергают лаву, которая в день начала извержения заполняет только клетку в месте начала извержения. На следующий день лава разливается и заполняет квадрат 3×3 , на третий день лава заполняет квадрат 5×5 , и так далее. Центры всех этих квадратов совпадают с местом начала извержения. Вулканы друг на друга не влияют, и распространение лавы каждого вулкана происходит независимо.

Ученым удалось вычислить точное количество вулканов и их параметры. Для дальнейших исследований нужно на острове поставить станцию, которая занимает одну клетку. Ученым интересно, какое максимальное количество дней может проработать станция. Станция становится неработоспособной в тот день, когда ее клетка будет затоплена лавой.