

## Задача A. Bartenders

Имя входного файла: `bartenders.in` или *стандартный ввод*  
Имя выходного файла: `bartenders.out` или *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Кто на Диком Западе не знает Неуловимого Джо? Неуловимого Джо не знает никто... как минимум, задав прямой вопрос, вы получите именно такой ответ. Но перед тем, как ответить, спрашиваемый обязательно оглянется по сторонам.

Ибо Неуловимый Джо является самым точным стрелком среди всех известных в округе ковбоев, при этом он знаменит ещё своим необузданным нравом и необычным для ковбоя кругом интересов — завсегда таи салуна рассказывали, что Джо как-то с интересом беседовал с невесть как оказавшимся в этих краях математиком. Правда беседа, как обычно, закончилась перестрелкой с другими посетителями салуна.

Так что с Джо предпочитали не связываться... Владельцы салунов даже установили для Джо специальную плату за вход (в дополнение к стоимости заказов).

Неуловимый Джо проснулся и понял, что ему чего-то не хватает... через несколько минут это «что-то» трансформировалось в количество бутылок виски. Джо собирается пройти по всем барам и купить нужное количество бутылок, потратив минимум денег. Естественно, что даже Неуловимый Джо не может купить в баре больше выпивки, чем там есть.

По заданным расценкам, размере входной платы для Джо и запасам спиртного в барах найдите минимальную сумму, которую потратит Джо.

### Формат входных данных

В первой строке входного файла задано одно целое число — количество тестовых примеров  $T$  ( $1 \leq T \leq 70$ ). В первой строке каждого тестового примера заданы два целых числа  $n$  и  $m$  — количество бутылок, которые хочет купить Джо, и количество баров в округе ( $1 \leq n \leq 10^4$ ,  $1 \leq m \leq 100$ ). Каждая из последующих  $m$  строк содержит по три целых числа  $b_i$  — количество бутылок виски в  $i$ -м баре,  $p_i$  — цена бутылки виски в  $i$ -м баре и  $j_i$  — плата, установленная за вход Джо в  $i$ -й бар ( $0 \leq b_i, p_i \leq 10^4$ ,  $0 \leq j_i \leq 10^6$ ). Гарантируется, что сумма всех  $b_i$  не менее  $n$ .

### Формат выходных данных

Для каждого тестового примера в отдельной строке выведите одно целое число — минимальную сумму, которая будет потрачена Джо на покупку выпивки.

### Пример

<code>bartenders.in</code> или стандартный ввод	<code>bartenders.out</code> или стандартный вывод
2	217
40 4	15
10 5 6	
20 4 12	
30 6 9	
42 7 0	
20 2	
10 0 5	
66 1 0	

## Задача В. Inside a Cube

Имя входного файла: `subbuilding.in` или *стандартный ввод*  
Имя выходного файла: `subbuilding.out` или *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Неуловимый Джо решил отправиться в поисках сокровищ в храм древней цивилизации майя. Храм имеет форму куба, составленного из одинаковых залов кубической формы. Каждому залу присвоены три номера  $x, y, z$  — порядковые номера по ширине, длине и высоте соответственно. Нумерация начинается с 1, 1, 1.

Из каждого зала можно перейти в любую из соседних, имеющих с ней общую стену.

В зале  $(x_B, y_B, z_B)$  Джо попался Страж Храма. Он сказал, что сокровища лежат в зале  $(x_M, y_M, z_M)$ , и что страж пропустит Джо только в случае, когда он скажет, чему равен остаток от деления количества способов, которыми можно попасть из зала со Стражем в комнату с сокровищами за наименьшее число переходов между залами, на  $10^9 + 7$ .

Помогите Джо в его расчётах.

### Формат входных данных

Входной файл содержит 6 целых чисел  $x_M, y_M, z_M, x_B, y_B, z_B$  ( $1 \leq x_M, y_M, z_M, x_B, y_B, z_B \leq 1000$ ).

### Формат выходных данных

Выходной файл должен содержать единственное целое число — остаток от деления количества оптимальных маршрутов на  $10^9 + 7$ .

### Пример

<code>subbuilding.in</code> или стандартный ввод	<code>subbuilding.out</code> или стандартный вывод
1 1 1 2 3 4	60
2 3 4 5 6 7	1680

## Задача C. DFS

Имя входного файла: `dfs.in` или *стандартный ввод*  
Имя выходного файла: `dfs.out` или *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Когда Неуловимый Джо «доходит до кондиции» в салуне старого Бена, он начинает рассуждать о странных для ковбоя вещах... например, о теории графов. Недавно Джо, основательно набравшись, озадачил всех следующим вопросом.

Рассмотрим алгоритм поиска в глубину для неориентированного графа  $G(V, E)$  без петель и кратных рёбер:

```
function dfs(x)
    print x;
    color[x] = true;
    for each (x, i) in E
        if (not color[i])
            dfs(i);

for each i in V
    if (not color[i])
        dfs(i);
```

Первоначально все элементы массива `color` равны `false`.

После работы алгоритма будет получена последовательность чисел — некий обход графа в глубину.

Даны граф и последовательность целых чисел  $a$ . Необходимо определить, существует ли такой порядок перебора вершин, чтобы приведенный алгоритм выдал последовательность  $a$ ?

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n \leq 10\,000$ ,  $0 \leq m \leq 100\,000$ ) — количество вершин и рёбер в графе. Далее следуют  $m$  пар целых чисел  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ ) обозначающих ребро  $(u_i, v_i)$  в графе. Далее следует строка, содержащая  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq n$ ,  $a_i \neq a_j$  при  $i \neq j$ ) — последовательность  $a$ .

### Формат выходных данных

Выходной файл должен содержать «Yes», если последовательность  $a$  может быть обходом в глубину этого графа, иначе — «No».

### Примеры

dfs.in или стандартный ввод	dfs.out или стандартный вывод
4 3 1 2 2 3 3 4 3 4 2 1	Yes
4 3 1 2 2 3 3 4 3 4 1 2	No

## Задача D. Drunk Walking

Имя входного файла: *или стандартный ввод*  
Имя выходного файла: `drunk.out` *или стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Будучи в нетрезвом состоянии, Неуловимый Джо начинает случайным образом перемещаться по городу. Дома в городе расположены на  $N$  горизонтальных и  $M$  вертикальных улицах так, что каждый дом находится на пересечении двух улиц. Таким образом, всего в городе  $N \times M$  домов. Каждую секунду Джо с равной вероятностью случайно выбирает один из соседних с текущим перекрёстков и переходит на него; при этом он, даже будучи мертвецки пьяным, никогда не выйдет за пределы города (тем самым в случае его нахождения в углу он выбирает любое из двух направлений с вероятностью  $1/2$ , в случае нахождения на стороне — любое из трёх направлений с вероятностью  $1/3$ , в случае нахождения около «внутреннего» дома — любое из четырёх направлений с вероятностью  $1/4$ ). Как только Джо попадает на перекрёсток, на котором находится его дом, он заходит внутрь и засыпает. Если же Джо за  $P$  или менее переходов не добрался домой, он устаёт и засыпает прямо на улице.

Вам известны размеры города, координаты салуна, из которого только что вышел Джо, а также координаты его дома. С какой вероятностью Джо в эту ночь доберётся домой?

### Формат входных данных

В первой строке входного файла задано одно целое число  $T$  ( $1 \leq T \leq 100$ ) — количество тестовых примеров. Далее следуют  $C$  строк, каждая из которых содержит один тестовый пример — 7 целых чисел  $N, M, X_j, Y_j, X_h, Y_h$  и  $P$  ( $2 \leq N, M \leq 10, 0 \leq X_j, X_h < N, 0 \leq Y_j, Y_h < M, 1 \leq P \leq 5000$ ) — соответственно размеры города, изначальные координаты Джо, координаты дома Джо и количество переходов, после которого Джо засыпает. Считать, что дом с координатами  $(0, 0)$  расположен в левом нижнем углу карты, а дом с координатами  $(N - 1, M - 1)$  — в правом верхнем.

### Формат выходных данных

Для каждого тестового примера в отдельной строке выведите одно число — вероятность того, что Джо в эту ночь заснёт дома, а не на улице, с точностью не хуже  $10^{-6}$ .

### Пример

<i>или стандартный ввод</i>	<code>drunk.out</code> <i>или стандартный вывод</i>
2	0.3333333
2 5 0 2 0 3 1	0.1111111
2 6 0 1 0 3 2	

## Задача E. Joe Got in Trouble!

Имя входного файла: `joe.in` или *стандартный ввод*  
Имя выходного файла: `joe.out` или *стандартный вывод*  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Неуловимый Джо попал в засаду! Проснувшись с утра в степи, он обнаружил вокруг себя банду Одноруких. Но основная неприятность была не в этом, а в том, что у Джо с собой не оказалось патронов... Остаётся уповать на собственную репутацию.

Однорукие знают, что Джо — отличный стрелок и при малейшем движении стреляет безошибочно, так что, пока все однорукие находятся в поле зрения Джо, он в безопасности (из этого же следует, что бандиты всё время находятся на тех же местах, на которых их застало появление Джо). В случае, если хотя бы один бандит находится вне поля зрения ковбоя, он сразу же стреляет в Джо и убивает его.

По словам Неуловимого Джо, угол обзора для него составляет 180 градусов. Так это или нет — проверять никто не пытался (в таком случае на Диком Западе предпочитают верить на слово).

Задача Джо — добраться живым до секретного тайника с патронами. Джо может в процессе движения поворачиваться на произвольный угол и двигаться в любом направлении (в том числе и спиной вперёд).

Выясните, сможет ли Джо добраться до тайника, и, если сможет, выведите наименьшую длину «безопасного» пути.

### Формат входных данных

В первой строке входного файла задано количество тестовых примеров  $T$  ( $1 \leq T \leq 100$ ). Каждый тестовый пример начинается с двух целых чисел  $x_j$  и  $y_j$  — начального положения Джо. Во второй строке заданы два целых числа  $x_a$  и  $y_a$  — координаты тайника с патронами. В третьей строке задано одно целое число  $n$  ( $1 \leq n \leq 1000$ ) — количество Одноруких бандитов. В последующих  $n$  строках заданы по два целых числа — в  $i$ -й строке заданы координаты  $i$ -го бандита  $x_i$  и  $y_i$ . Все координаты (Джо, тайника и каждого из бандитов) попарно различны и по модулю не превышают  $10^4$ .

### Формат выходных данных

Выведите одно число — длину кратчайшего пути, следуя по которому, Джо останется в живых и доберётся до патронов. Если добраться до патронов живым Джо не сможет, выведите  $-1$ . Длину следует выводить с точностью не хуже  $10^{-3}$ .

### Пример

<code>joe.in</code> или стандартный ввод	<code>joe.out</code> или стандартный вывод
2	5.657
0 0	-1
4 0	
2	
2 2	
2 -4	
2 2	
4 2	
3	
3 3	
3 0	
1 2	

## Задача F. Play-Off

Имя входного файла:	<code>playoff.in</code> или <i>стандартный ввод</i>
Имя выходного файла:	<code>playoff.out</code> или <i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Неуловимый Джо решил принять участие в Открытом Чемпионате Дикого Запада по спортивной стрельбе. В этом году организаторы придумали несколько новых конкурсов, в частности, поединок пяти лучших американских стрелков с пятью лучшими китайскими. Но основным мероприятием всё равно остаётся чемпионат. Перед началом турнира Джо собирается проверить, справедлива ли жеребьёвка.

В чемпионате, который проходит по системе play-off, принимает участие  $2^n$  игроков. Перед первым раундом игроки каким-то образом разбиваются на пары. Далее проходят состязания в каждой паре. Победитель переходит в следующий раунд, а проигравший с пулевыми ранениями выбывает из турнира. Во втором раунде в первую пару попадают победители первой и второй пар первого раунда, во вторую пару попадают победители третьей и четвёртой пары и так далее. Все последующие раунды проходят по этой же схеме. Турнир продолжается до тех пор, пока не останется один участник. То есть всего будет  $n$  раундов.

Джо известен рейтинг всех участников. Он считает жеребьёвку справедливой, если каждый участник сможет занять место в соответствии со своим рейтингом (сильнейший участник занял первое место, следующий по силе — второе и так далее).

Победитель раунда номер  $n$  занимает первое место, проигравший второе. Выбывшие в  $n - 1$  раунде занимают 3 — 4 места, выбывшие в  $n - 2$  раунде занимают 5 — 8 места и так далее.

### Формат входных данных

Первая строка входного файла содержит одно целое число  $n$  ( $1 \leq n \leq 20$ ). Игроки нумеруются числами от 1 до  $2^n$ . Будем считать, что участник с номером  $i$  имеет рейтинг  $i$ . Участник с рейтингом 1 должен занять первое место, с рейтингом 2 — второе и так далее.

### Формат выходных данных

Выходной файл должен содержать  $2^n$  целых чисел — одну из возможных жеребьёвок. Первые два числа должны обозначать номера игроков в первой паре, следующие два числа — номера игроков во второй паре и так далее.

### Примеры

<code>playoff.in</code> или <i>стандартный ввод</i>	<code>playoff.out</code> или <i>стандартный вывод</i>
2	1 3 2 4