

А. Роботы-курьеры

Ограничение времени	20 секунд
Ограничение памяти	1Gb
Ввод	стандартный ввод
Вывод	стандартный вывод

В Иннополисе уже давно наступило будущее, и доставка заказа из кафе человеком - скорее редкость, чем правило. Большинство заказов в городе доставляют роботы-курьеры, и в этой задаче мы предлагаем вам поучаствовать в распределении роботов по заказам, происходящим в течение некоторого времени.

Представим город в виде карты размера $N \times N$. Для простоты предположим, что робот занимает ровно одну клетку и каждая клетка для него может быть либо проходимой, либо нет. За одну секунду робот может переместиться в любом из четырёх направлений (вверх/вниз/влево/вправо), если клетка, куда он хочет переместиться, свободна.

В начале теста вам нужно вывести количество роботов, которое вы хотите использовать для доставки заказов, и их изначальные координаты. Постройка каждого робота будет стоить $Cost_c$ рублей.

Далее будет произведено T итераций симуляции. Одна итерация представляет собой одну виртуальную минуту и состоит из 60 секунд. На каждой итерации вашей программе будет передано количество новых заказов, а в ответ программа должна сообщить, какие действия выполняет каждый робот (по 60 действий для робота).

За каждый успешно доставленный заказ вы получите $\max(0, MaxTips - DeliveryTime)$ рублей чаевых, где $MaxTips$ — максимальное количество чаевых для одного заказа, а $DeliveryTime$ — время с момента появления заказа до его доставки в секундах.

Итоговое количество очков, которое вы заработаете за один тест вычисляется по формуле $TotalTips - R \times Cost_c$, где $TotalTips$ — общее количество заработанных чаевых, R — количество использованных роботов, $Cost_c$ — цена постройки одного робота.

Значения $Cost_c$ и $MaxTips$ задаются в каждом тесте. Если вы заработали меньше чаевых, чем потратили на производство роботов, итоговое количество очков будет равно 0. Также вы получите 0 очков за тест в случае выполнения любого некорректного действия.

Формат ввода

Для чтения входных данных программа должна использовать стандартный ввод.

В первой строке ввода заданы три натуральных числа N , $MaxTips$ и $Cost_c$ ($N \leq 2\,000$, $MaxTips \leq 50\,000$, $Cost_c \leq 10^9$) — размер города, максимальное количество чаевых за заказ и цена постройки одного робота. Каждая из следующих N строк содержит N символов — карту города. Строки могут содержать два типа символов:

- ' #' - клетка занята препятствием.
- ' . ' - свободное пространство.

Затем вам на вход будет подано два целых натуральных числа T и D ($T \leq 100\,000$, $D \leq 10\,000\,000$) — количество итераций взаимодействия и суммарное количество заказов.

После этого вам необходимо вывести число R - количество роботов, которые вы разместите в городе. Роботов должно быть не менее, чем 1 и не более, чем 100. Затем выведите R пар целых чисел от 1 до N — координаты, где роботы будут изначально расположены.

Далее на каждой из T итераций мы сообщаем информацию о новых размещенных заказах. На каждой итерации сначала дано целое число k — количество новых курьерских заказов, затем k строк с числами $S_{row}, S_{col}, F_{row}, F_{col}$ — координаты начальной и конечной точки заказа ($1 \leq S_{row}, S_{col}, F_{row}, F_{col} \leq N$). Новый заказ может быть размещён в той же клетке, где уже находится 1 или более заказов. Время жизни заказа не ограничено.

Формат вывода

Для осуществления запросов программа должна использовать стандартный вывод.

На каждой итерации в ответ вы сообщаете нам о действиях каждого из своих роверов: R строк по 60 символов в каждой (один символ - одно действие, суммарно по 60 действий каждого ровера):

- U - движение на одну клетку вверх (уменьшить номер строки)
- L - движение на одну клетку влево (уменьшить номер столбца)
- D - движение на одну клетку вниз (увеличить номер строки)
- R - движение на одну клетку вправо (увеличить номер столбца)
- S - остаться на месте и ничего не делать
- T - остаться на месте и забрать самый старый заказ в текущей клетке
- P - остаться на месте и выдать заказ в текущей клетке

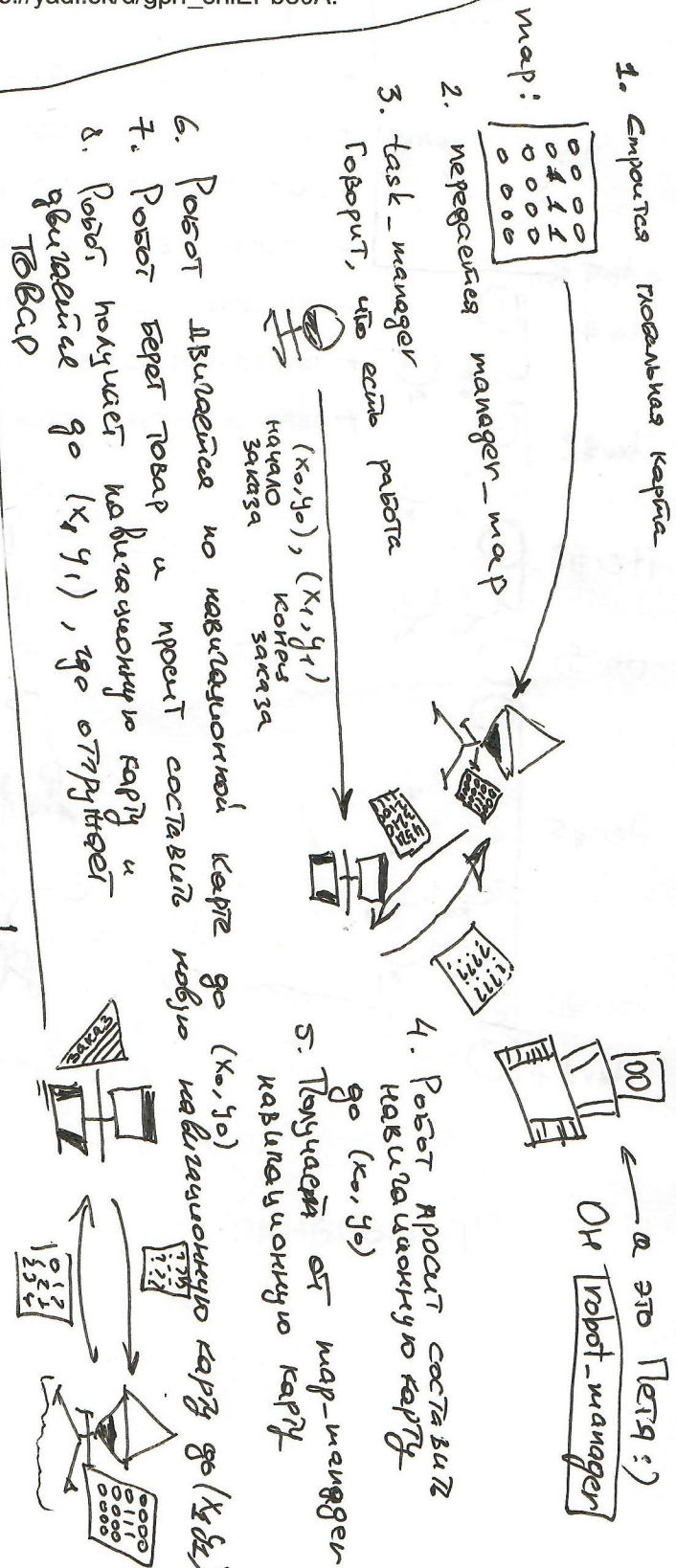
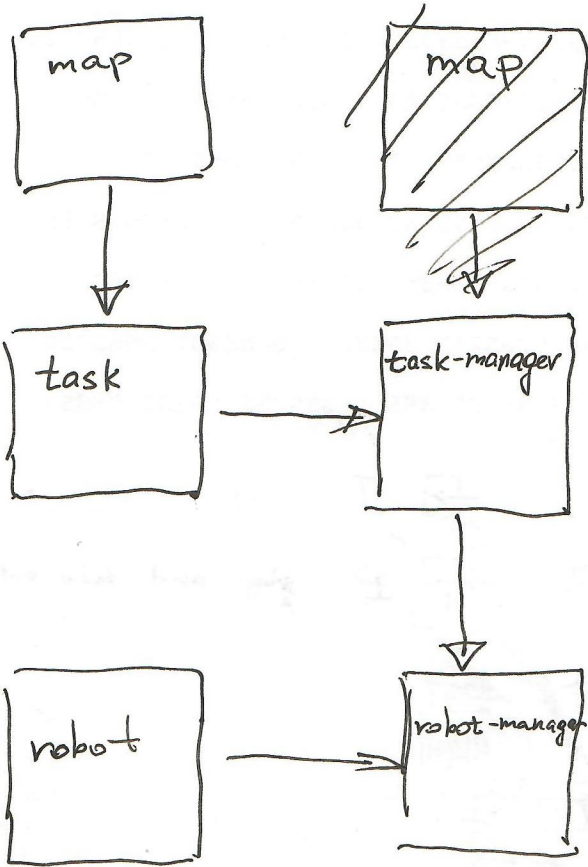
Роботы выполняют свои действия по очереди: сначала первое действие выполняет первый робот, затем второй и так далее до последнего робота. Потом первый робот выполняет второе действие, второй робот выполняет второе действие и так далее. В конце итерации каждый робот выполняет своё последнее действие и итерация заканчивается.

Несколько роботов могут занимать одну и ту же клетку.

Тестирующая система даст вашей программе прочитать свежие данные из входных данных только после того, как ваша программа вывела соответствующий запрос системе и выполнила операцию `flush`.

Вам предоставлены для ознакомления примеры тестовых данных, на которых будет тестироваться ваше решение. Гарантируется, что в системных тестах такие данные, как N , $MaxTips$, $Cost_c$, T и карта будут в точности такими же, как в ознакомительных тестах. Сами же заказы будут сгенерированы случайно, но тем же алгоритмом с тем же распределением, что и в ознакомительных тестах. Гарантируется, что значение D из системного теста будет отличаться от значения D в соответствующем ознакомительном тесте не более, чем на 1 процент.

Ознакомительные примеры доступны по ссылке: https://yadi.sk/d/gprf_snlEPb80A.



map-manager владеет картой

получает: начальную и конечную точки \oplus пустой блок для карты

выдает: навигационную карту

task-manager владеет системой

получает: запрос на заказ

выдает: (x_0, y_0) , (x_1, y_1) начало заказа, конец заказа

robot-manager владеет роботом

- навигационной картой
- началом и концом заказа

делает: 2 запроса на составление навигационных карт и запрос на заказ.