

### Замечание

Во всех задачах, если не указано особое, программа должна производить ввод со стандартного потока ввода или файл input.txt, выводить на стандартный поток или файл output.txt, ограничение по времени 2 с, ограничение по памяти 256МВ.

## Задача А. Цифровой корень

Цифровой корень натурального числа — сумма цифр от суммы цифр от суммы цифр ... числа. Например, для числа 345 цифровой корень 3 (345 -> 12 -> 3)

Ваша задача для заданного цифрового корня  $K$  найти  $N$ -значное число, цифровой корень которого равен  $K$ .

### Формат входных данных

Вводится два целых числа  $N$  и  $K$  ( $1 \leq N \leq 10^5, 1 \leq K \leq 9$ )

### Формат выходных данных

Требуется вывести искомое число без ведущих нулей.

### Пример

стандартный ввод	стандартный вывод
3 3	345

## Задача В. Перфекционист Глеб

Холодным мартовским утром Глеб играл на улице в игру: для начала он придумывает перестановку из чисел от 1 до  $N$  —  $a$ , также запоминая  $a_R$  - перестановку получающаяся из  $a$  разворотом. Потом он выбирает число  $K$ , и выписывает  $K$  раз подряд последовательность, полученную склеиванием  $a$  и  $a_R$ . Глеб очень любит чтобы у полученной последовательности длина наибольшей возрастающей подпоследовательности равнялась  $N$ . Помогите Глебу найти такое минимальное  $K$ .

### Формат входных данных

Первая строка входных данных содержит число  $N$  ( $1 \leq N \leq 300\,000$ )

Во второй строке записаны  $N$  целых чисел  $a_i$  ( $1 \leq a_i \leq n, a_i \neq a_j$  для  $i \neq j$ ) — загаданная перестановка.

### Формат выходных данных

В единственной строке выведите искомое  $K$ . Гарантируется, что такое  $K$  существует.

### Примеры

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

## Задача С. Связность графа

Однажды на день рождения Ивану подарили граф состоящий из  $N$  вершин в котором вершины с номерами  $x$  и  $y$  соединены ребром тогда и только тогда, когда  $\text{gcd}(x, y) \geq g$ , где

### Формат входных данных

Первая строка входных данных содержит три целых числа  $N, g, M$  и ( $1 \leq N, g, M \leq 200\,000$ ) — количество вершин в графе и параметр графа и количество запросов на связность.

Далее следуют  $m$  строк, описывающих запросы, каждый запрос задан двумя числами  $x$  и  $y$ , ( $1 \leq x, y \leq N, x \neq y$ ).

### Формат выходных данных

Выведите  $M$ , по одной на каждый запрос, соответственно «Yes», если вершины находятся в одной компоненте, и «No» иначе.

### Пример

стандартный ввод	стандартный вывод
6 2 4	Yes
2 3	Yes
4 3	No
5 1	Yes
2 6	

## Задача D. ЧПУ-фрезер

В этой задаче вам требуется написать программу для фрезерного станка с ЧПУ по изображению заготовки. Изображение задается в виде прямоугольника  $N \cdot M$ ,

где  $N$  – количество строк, а  $M$  – количество столбцов. Символ '.' означает, что эта клетка не должна быть обработана. Символ '\*' означает, что эта клетка должна быть отфрезерована.

Фреза станка изначально находится в точке с координатой  $(0, 0)$  (левый верхний угол картинке) и может за одну секунду переместиться в одну из соседних клеток (команды  $N$  - вверх,  $E$  - вправо,  $S$  - вниз,  $W$  - влево). Кроме того, в каждый момент времени фреза может находиться или в опущенном состоянии и тогда происходит процесс обработки, или в поднятом состоянии и тогда обработка не происходит. Подъем и опускание фрезы занимает 1 секунду. Обработка клетки происходит за одну секунду.

В процессе обработки заготовки происходит нагрев фрезы в опущенном состоянии на 1 градус за одну секунду и охлаждение на 1 градус в поднятом состоянии. Соответственно, команда  $D$  позволяет опустить фрезу и команда  $U$  позволяет поднять фрезу. Будем считать, что время опускания и поднятия фрезы пренебрежимо малы.

Начальная и минимальная температура 30 градусов. При нагреве больше 50 градусов фреза ломается. Если фреза выходит за границы изображения, то она ломается.

Ваша задача сгенерировать программу для фрезера из команд  $N$ ,  $E$ ,  $S$ ,  $W$ ,  $U$ ,  $D$ , которая получает заданное изображение.

Программа для фрезера оценивается по времени получения изображения. Программы которые приводят к поломке фрезы считаются неверными.

### Формат входных данных

Изображение описывается числами  $N$  и  $M$  ( $1 \leq N, M \leq 1000$ ), затем идет  $N$  строк по  $M$  символов '.' или '\*'

### Формат выходных данных

Вывести требуется программу из команд 'N','E','S','W','U','D' в одну строку без пробела, которую должен выполнить станок для получения изображения. Длина программы фрезера не должна превышать  $10^6$  символов.

### Система оценки

Итоговый балл за каждый тест и подзадачу будет выставляться в зависимости от лучшего полученного результата среди участников после олимпиады.

### Пример

стандартный ввод	стандартный вывод
5 5 ...* ...* *...* ...* *****	EEEEDSUWWWSDUEEEEDSUWWWSDUEEEED

### Замечание

В задаче есть несколько подзадач с файлам изображений, для которых требуется сгенерировать и отправить на проверку файл с программой. Отдельной подзадачей требуется отправить программу на одном из языков программирования, которая для произвольного изображения получает программу для станка.

При проверке файлов с изображениями будет доступен протокол проверки корректности программы для станка и время работы для программы не приводящей к поломке.

Подзадача со сдачей программы на языке программирования будет проверяться на всех тестовых изображениях во время олимпиады и на некотором наборе тестов после окончания олимпиады.

## Задача Е. Квадрокоптеры

Однажды Федор решил провести чемпионат мира по программированию. Ясно что такое соревнование нельзя провести без торжественного открытия, а открытие без красочного шоу. К счастью у Федора есть  $N$  квадрокоптеров и он решил построить из них заданную фигуру в зале над участниками.

Управление квадрокоптера осуществляется следующим образом. Изначально в квадрокоптер загружается программа, которая каждый момент времени решает куда должен переместиться квадрокоптер - влево, вправо, вверх, вниз или остаться на месте (эти команды кодируются числами от 1 до 5 соответственно). Каждый момент времени каждый квадрокоптер принимает решение о перемещении. После этого все квадрокоптеры одновременно совершают это перемещение.

У каждого квадрокоптера есть 10 байт памяти которую он может изменять каждый момент времени. У каждого квадрокоптера есть сенсоры которые позволяют считать состояние памяти соседних (по сторонам) квадрокоптеров или квадрокоптеров в той же клетке.

Представим зал в виде прямоугольной сетки, к некоторым клеткам которой находятся квадрокоптеры. Изначально они находятся в неизвестных различных ко-

ординатах, но гарантируется что начальная конфигурация является связной по сторонам фигурой.

Вам требуется написать программу которая выполняется на квадрокоптере.

При формировании фигуры квадрокоптерами важно только относительное положение квадрокоптеров относительно друг-друга. То есть требуется чтобы в какой-то момент времени относительная конфигурация должна совпадать с заданной.

### Формат входных данных

Каждый момент времени на вход программе подается неизменяемая целевая конфигурация квадрокоптеров: число  $N$  ( $1 \leq N \leq 20$ ) и  $N$  пар целых положительных чисел (каждое из которых до 100) - точки в которых могут находиться квадрокоптеры, чтобы сформировать нужную конфигурацию.

Следующие 10 чисел задают состояние памяти текущего квадрокоптера, на котором выполняется программа - числа от 0 до 255, разделенные пробелами

Затем следует число соседних квадрокоптеров  $M$ . Следующие  $M$  строк задают состояние каждого из квадрокоптеров - первое число от 1 до 5 где он находится (слева, справа, сверху, снизу, в той же клетке) и 10 чисел от 0 до 255 - состояние его памяти.

### Формат выходных данных

Требуется вывести текущее действие квадрокоптера - число от 1 до 5, затем 10 байт - новое состояние памяти

### Замечание

Для каждого хода для каждого квадрокоптера запускается решение с данными, описанными во входном формате. Программа делающая один ход не должна предполагать что можно как-то сохранить данные кроме как в 10 байтах памяти квадрокоптера.

Пример ввода для одного хода:

```
4
0 0
1 1
2 2
3 3
0 1 200 0 0 0 0 0 0 0
3
1 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0
```

Пример вывода:

```
2
0 1 199 0 0 0 0 0 0 0
```

## Ответы и критерии проверки заключительного этапа.

Тесты, проверяющие программы, и ответы находятся в архиве тестирующей системы:

<https://ejudge.cs.msu.ru/lomo18.tar.gz>

Протоколы проверки и технические баллы доступны по логину и паролю участника на сайте

[https://ejudge.cs.msu.ru/ej/client?contest\\_id=104&locale\\_id=1](https://ejudge.cs.msu.ru/ej/client?contest_id=104&locale_id=1)

[https://ejudge.cs.msu.ru/ej/client?contest\\_id=105&locale\\_id=1](https://ejudge.cs.msu.ru/ej/client?contest_id=105&locale_id=1)

Критерии проверки (10-11 классы):

1 - 101 тест, по 1 баллу за тест, кроме первого, за него 0.

2 - 72 теста, по 1 баллу за первые 44 теста, за остальные 2 балла.

3 - 55 тестов, по 1 баллу за первые 10 тестов, остальные по 2.

4 - D1-D5 50 баллов, по 10 баллов за каждую в зависимости от полученного ответа, критерии на число баллов в checker\_1.cpp - checker\_5.cpp. D - 50 баллов - сумма по каждому тесту, отмасштабированная согласно valuer.py.

5 - 20 тестов, с разбалловкой test\_score\_list = "2 2 2 2 2 4 4 4 4 2 2 2 2 2 4 4 4 4", максимальный балл 100 за полное решение.

6 - 10 тестов по 10 баллов.

В 5-9 классе аналогично без 5-й задачи.

Оценка выставлялась по следующей схеме: в 10-11 классах балл, полученный из критериев делился на 431, затем умножался на 100 и округлялся до ближайшего целого, а в 5-9 классах балл, полученный из критериев делился на 350, затем умножался на 100 и округлялся до ближайшего целого.