

## Задача А. Усердные бобры

Имя входного файла: `beavers.in`  
Имя выходного файла: `beavers.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайты

Бобры — уникальные животные. Больше всего они известны своим умением строить дамбы на реках.

Рассмотрим подробно процесс постройки дамбы. Исходно бобер стоит перед длинным рядом деревьев. Будем считать этот ряд бесконечным в обе стороны. Каждое дерево в ряду может быть либо *хорошим*, либо *плохим*. Исходно все деревья хорошие. Но после того, как бобер погрызет хорошее дерево, оно может стать плохим, и наоборот, после того как бобер погрызет плохое дерево, оно может стать хорошим.

Бобер может быть в одном из пяти настроений: он может быть сердитым (*angry*), усердным (*busy*), творческим (*creative*), отчаявшимся (*despaired*) и утомленным (*exhausted*). Исходно бобер сердит.

Когда бобер стоит перед деревом, в зависимости от своего настроения и того, хорошее ли дерево перед ним, он делает следующие действия:

- грызет дерево, при этом возможно оно меняется с плохого на хорошее или наоборот;
- меняет свое настроение на одно из возможных пяти (настроение бобра может остаться тем же);
- перемещается либо к соседнему дереву налево, либо к соседнему дереву направо.

Также при некоторой комбинации настроения бобра и состояния дерева перед ним, бобер может наконец решить, что плотина готова, и становится счастливым (*happy*). Может случиться, однако, что бобер так никогда и не станет счастливым.

Вам заданы правила, по которым действует бобер. Выясните, станет ли этот бобер счастливым.

### Формат входного файла

Входной файл содержит десять слов, разделенных пробелами. Каждое слово состоит из трех символов, эти слова задают поведение бобра. Первое слово задает поведение сердитого бобра на хорошем дереве, второе слово задает поведение сердитого бобра на плохом дереве, третье слово задает поведение усердного бобра на хорошем дереве, и т.д., последнее слово задает поведение утомленного бобра на плохом дереве.

Первый символ каждого слова равен '1', если бобер делает дерево плохим, либо '0', если бобер делает дерево хорошим. Второй символ равен 'R', если бобер перемещается направо, либо 'L', если бобер перемещается налево. Наконец, третий символ равен либо букве от 'A' до 'E' и описывает новое настроение бобра, либо 'H', если бобер становится счастливым.

### Формат выходного файла

Выведите "happy beaver", если бобер становится счастливым, или "unhappy beaver", если бобер не становится счастливым.

### Примеры

<code>beavers.in</code>	<code>beavers.out</code>
<code>1RB 1LC 1RC 1RB 1RD 0LE 1LA 1LD 1RH 0LA</code>	<code>happy beaver</code>
<code>ORA ORA ORA ORA ORA ORA ORA ORA ORA ORA</code>	<code>unhappy beaver</code>

## Задача В. Бюджет

Имя входного файла: `budget.in`  
Имя выходного файла: `budget.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Всем известно, что основной целью любого крупного правительственного проекта во Флатландии является освоение бюджетных средств (разумеется, по назначению). В настоящее время во Флатландии ведется работа над  $m$  национальными проектами,  $i$ -й проект может освоить  $s_i$  миллионов Флатландских тугриков в день.

Правительство Флатландии планирует выделить  $n$  грантов для финансирования проектов, каждый по  $p$  миллионов Флатландских тугриков. Деньги  $i$ -го из грантов будут доступны для освоения, начиная с дня  $r_i$ . Когда очередной грант становится доступен для освоения, его можно отдать некоторому проекту. Этот проект осваивает деньги гранта в течение  $\lceil p/s_i \rceil$  дней. Проект не может одновременно осваивать деньги нескольких грантов.

Премьер-министр Флатландии господин Тупиков хочет выяснить, за какое время удастся освоить все деньги, выделенные в рамках грантов. Помогите ему выяснить самый ранний день, когда можно полностью освоить все деньги грантов.

### Формат входного файла

Первая строка входного файла содержит числа  $m$ ,  $n$  и  $p$  ( $1 \leq m \leq 100$ ,  $1 \leq n \leq 100$ ,  $1 \leq p \leq 10^9$ ). Вторая строка содержит  $m$  целых чисел:  $s_1, s_2, \dots, s_m$  ( $1 \leq s_i \leq 10^9$ ). Третья строка содержит  $n$  целых чисел:  $r_1, r_2, \dots, r_n$  ( $1 \leq r_i \leq 10^9$ ).

### Формат выходного файла

Первая строка выходного файла должна содержать одно целое число — минимальный день, к которому можно полностью освоить все деньги грантов.

### Примеры

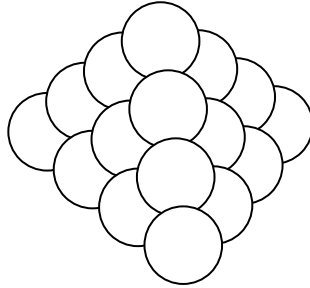
<code>budget.in</code>	<code>budget.out</code>
2 4 22	17
2 5	
1 3 8 12	

Одна из возможных оптимальных схем освоения устроена следующим образом: грант 1 отдается первому проекту, который осваивает его в течение 11 дней. Остальные гранты отдаются второму проекту, грант 2 осваивается в течение дней 3–7, грант 3 в течение дней 8–12 и грант 4 в течение дней 13–17. Заметим, что грант 4 появляется в день 12, но назначается только в день 13.

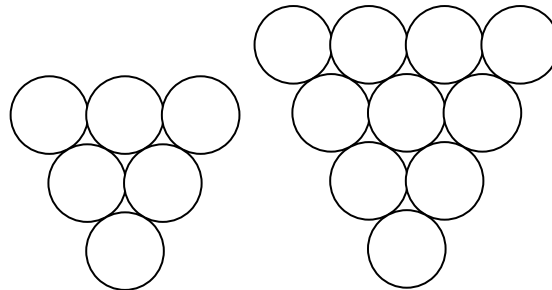
## Задача С. Конфеты Дяди Федора

Имя входного файла:	candies.in
Имя выходного файла:	candies.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайта

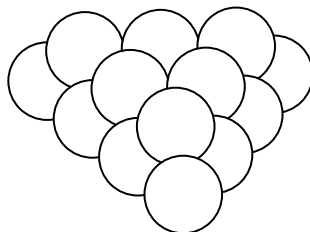
В погожий субботний день небезызвестные жители Простоквашино по традиции собрались на чаепитии у Дяди Федора. На этот раз Дядя Федор открыл ящик с заморскими шарообразными шоколадными конфетами, а Почтальон Печкин в силу своей аккуратности построил из всех конфет пирамидку в форме правильного тетраэдра.



Каждый слой пирамидки представляет из себя правильный треугольник, выложенный из конфет, причем сторона его содержит на одну конфету меньше, чем сторона предыдущего слоя. Например, так выглядят третий и четвертый сверху слои пирамидки.



Галчонку Хватайке очень не понравилось распределение конфет по слоям (верхний слой состоит из одной конфеты, а последний может состоять, к примеру, из десяти!), и он решил немного исправить ситуацию. Приведя всех в заблуждение и ослабив бдительность фразой «Кто там?» он целиком съел несколько верхних слоев пирамидки.



Этот поступок заметили и дружно начали ловить заметно растолстевшего галчонка. Разумеется, что в суете стол опрокинули, и все конфеты рассыпались по полу.

Операцию по поиску и сбору конфет провели Шарик и Матроскин, которые нашли  $n$  конфет, и заинтересовались, не потерялись ли еще конфеты. Так как никто не помнит, сколько конфет было, да и анализ внешности и состояния Хватайки ничего не дал, будем считать, что конфеты не потерялись, если их можно сложить в усеченную правильную пирамидку, то есть пирамидку без нескольких верхних слоев. Помогите жителям Простоквашино понять, потерялись ли столь ценные заморские конфетки!

## Формат входного файла

Входной файл содержит единственное число  $n$  ( $0 \leq n \leq 10^{17}$ ) — количество найденных конфет.

## Формат выходного файла

Если из  $n$  конфет можно сложить усеченную пирамидку, то выведите в выходной файл слово YES, иначе — выведите слово NO.

## Примеры

candies.in	candies.out
0	YES
1	NO
3	YES
6	YES
16	YES

## Задача D. Разложение графа

Имя входного файла: `factor.in`  
Имя выходного файла: `factor.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Рассмотрим неориентированный граф  $G$ . Его  $r$ -фактором называется подграф  $G$ , содержащий все его вершины, и некоторое подмножество ребер  $G$ , при этом степень каждой вершины должна быть равна  $r$  (такой граф называется  $r$ -регулярным).

Рассмотрим  $k$ -регулярный граф  $G$ , пусть  $\lambda$  — разбиение числа  $k$  на слагаемые:  $k = a_1 + a_2 + \dots + a_m$ . Тогда  $\lambda$ -разложением графа  $G$  называется набор графов  $G_1, G_2, \dots, G_m$ , такой что  $G_i$  представляет собой  $a_i$ -фактор графа  $G$ , и каждое ребро исходного графа принадлежит ровно одному из графов  $G_i$ .

Для полного графа с четным числом вершин  $K_{2n}$  и разбиения  $\lambda$  числа  $2n - 1$  на слагаемые постройте  $\lambda$ -разложение  $K_{2n}$ .

### Формат входного файла

Первая строка входного файла содержит  $n$  ( $1 \leq n \leq 100$ ). Вторая строка содержит число  $m$ , и затем числа  $a_1, a_2, \dots, a_m$  ( $1 \leq a_i \leq 2n - 1$ ,  $\sum a_i = 2n - 1$ ).

### Формат выходного файла

Выведите  $m$  описаний графа. Описание  $G_i$  должно содержать  $na_i$  ребер. Разделяйте описания графов пустыми строками.

### Примеры

<code>factor.in</code>	<code>factor.out</code>
3	1 4
3 1 2 2	2 5
	3 6
	1 2
	2 3
	3 4
	4 5
	5 6
	6 1
	1 3
	3 5
	5 1
	2 4
	4 6
	6 2

## Задача Е. Здоровое питание

Имя входного файла: `food.in`  
Имя выходного файла: `food.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Как и всем студентам, Володе Ядолову необходимо здоровое питание. Каждый день, вот уже несколько лет, Володя ест на завтрак две булочки и сок.

К несчастью для Володи, булочки не вечны и портятся через  $k$  дней хранения, поэтому каждое утро он ходит в магазин и закупается в нем свежими булочками, причем из-за инфляции цены на булочки постоянно меняются. Благодаря своей хитрости Володе удалось выяснить у продавца магазина цены булочек на ближайшие  $m$  дней. Теперь Володя хочет составить план закупок завтраков, которыми он будет питаться в последующие  $m$  дней. В связи с финансовым кризисом и природной жадностью Володя хочет потратить на булочки наименьшее возможное количество денег.

Помогите ему в этом!

### Формат входного файла

В первой строке входного файла находятся два числа:  $m$  — число дней на которые надо разработать план закупок, и  $k$  — срок хранения булочек ( $1 \leq m, k \leq 100\,000$ ). Во второй строке дано  $m$  натуральных чисел  $c_i$  — цена булочки в  $i$ -ый день ( $1 \leq c_i \leq 1\,000\,000$ ).

### Формат выходного файла

В первой строке входного файла выведите число  $C$  — количество денег, необходимое Володе. Во второй строке выведите  $m$  чисел  $d_i$  — количество булочек, покупаемых Володей в  $i$ -ый день.

### Примеры

<code>food.in</code>	<code>food.out</code>
2 1	6
1 2	2 2
3 2	10
3 1 2	2 4 0

## Задача F. Налеее-во!

Имя входного файла: leftturn.in  
Имя выходного файла: leftturn.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Прошло около двух недель, и разведчик Смит снова получил срочное и ответственное задание. Наскоро попрощавшись с десятком новоиспеченных подруг и наспех отдав распоряжения своему дворецкому, в три часа ночи он уже находился в аэропорту.

И вот он, все тот же плац. Моросит мелкий дождик, грязи где по колено, а где — по грудь, но это не смущает ни бравого сержанта, бодрым голосом отдающего команды, ни солдата Перетеркина, их беспрекословно выполняющего. Вчера, занимаясь подметанием плаца, рядовой Перетеркин один лом сломал, а второй потерял. В качестве воспитательной меры ему было назначено три часа строевых тренировок, и первый час этого достойного занятия уже подходил к концу.

Задание разведчика Смита состоит в том, чтобы выяснить, насколько точно выполняет строевые приемы среднестатистический рядовой армии потенциального противника. Для этого он через равные промежутки времени фотографирует плац из укрытия, а команды записывает на портативный шпионский диктофон с направленным микрофоном. Поначалу Смит думал, что задание будет легким, но когда число снимков перевалило за третью сотню, до него постепенно стало доходить, что он так легко не отделается.

Немного поразмыслив, Смит — а он парень умный — понял, что стоящую перед ним задачу он может свести к следующей: даны начальное положение солдата на плацу и последовательность команд, которые солдат исполняет, требуется найти положение солдата, которое он займет после исполнения всех этих команд. Но как же решать эту задачу? И тут Смита осенило, и, как только сержант сделал паузу, достал из-за пазухи телефон и стал набирать знакомый номер...

Известно, что на данной строевой тренировке отрабатывались следующие команды: «налево», «направо», «кругом» и «шаг вперед». Представим плац прямоугольным полем размера  $N \cdot M$ , а солдата на нем — фигурой, занимающей три подряд идущие смежные клетки. Далее проиллюстрировано выполнение команд.

Команда «налево»

```
.....  
..... \...  
.. \-/. .. |...  
..... /...  
.....
```

Команда «направо»

```
.....  
..... /...  
.. \-/. .. |...  
..... \...  
.....
```

Команда «кругом»

```
.....  
.. \-/. .. /-\...  
.....
```

Команда «шаг вперед»

```
.....  
.....  
.. \-/  
.. \-/  
.....  
.....
```

## Формат входного файла

В первой строке входного файла содержатся три целых числа  $N$ ,  $M$ ,  $K$  ( $1 \leq N, M, K \leq 100$ ). Далее следуют  $N$  строк по  $M$  символов каждая — описание исходного положения солдата на плаце.

Следующие  $K$  строк содержат команды, отданные солдату. По причине того, что разведчик Смит отчитывается перед начальством на английском языке, команды будут также даны на английском языке. В таблице приведены переводы используемых команд.

Налево	Left turn
Направо	Right turn
Кругом	About turn
Шаг вперед	Forward step

Гарантируется, что при выполнении команд солдат остается в пределах границ плаца.

## Формат выходного файла

В выходной файл выведите  $N$  строк по  $M$  символов каждая — описание положения солдата на плаце после выполнения команд.

## Примеры

leftturn.in	leftturn.out
4 7 4	.....
.....	.. \-/ .....
.....	.....
.. /-\.	.....
.....	
Left turn	
Right turn	
About turn	
Forward step	



## Задача G. Языки

Имя входного файла: `lingvo.in`  
Имя выходного файла: `lingvo.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Федор решил изучать новый язык и первым делом выучил названия цифр. Чтобы закрепить свои знания, он выбирает число и находит в нем последнюю в алфавитном порядке цифру. Однако Федор еще не до конца уверен в своих знаниях и поэтому нуждается в проверке.

Напомним, что слово  $a$  идет раньше слова  $b$  в алфавитном порядке, если либо слово  $a$  является префиксом  $b$ , либо первые несколько (возможно ноль) символов у них совпадают, а следующий символ в слове  $a$  идет раньше в алфавите соответствующего символа в слове  $b$ .

Напишите программу, которая поможет Федору проверить свои знания. Задано число. Найдите в нем цифру, название которой идет по алфавиту позже названий других цифр. Названия цифр заданы во входном файле.

### Формат входного файла

В первой строке входного файла через пробел заданы названия цифр от 0 до 9 на языке, который изучает Федор. Длина названий цифр не более 50 символов. Во второй строке записано число  $n$  ( $1 \leq n \leq 1000$ ). Следующие  $n$  строчек содержат числа, для которых Федор хочет узнать ответ. В следующих  $n$  строках записано по одному числу  $a_i$  ( $0 \leq a_i \leq 10^9$ ,  $a_i$  не имеет ведущих нулей).

### Формат выходного файла

В  $i$ -ой строке выходного файла, выведите одно слово — последнее по алфавиту название цифры, которая есть в числе  $a_i$ .

### Примеры

<code>lingvo.in</code>
<code>nulo unu du tri kvar kvin ses sep ok nau</code>
<code>3</code>
<code>123</code>
<code>456</code>
<code>908</code>
<code>lingvo.out</code>
<code>unu</code>
<code>ses</code>
<code>ok</code>
<code>lingvo.in</code>
<code>zero un deux trois quatre cinq six sept huit neuf</code>
<code>3</code>
<code>123</code>
<code>456</code>
<code>908</code>
<code>lingvo.out</code>
<code>un</code>
<code>six</code>
<code>zero</code>

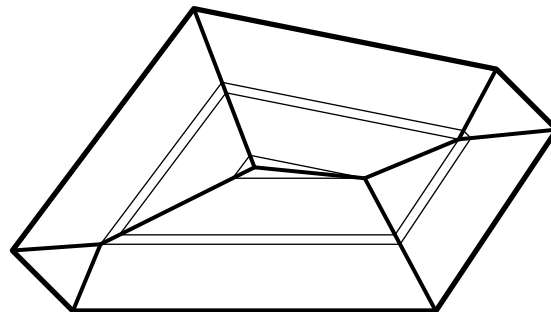
## Задача Н. Опора для крыши

Имя входного файла: `roof.in`  
Имя выходного файла: `roof.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Фермер Сергей планирует построить крышу своему свинарнику. Основание свинарника имеет форму выпуклого многоугольника с  $n$  вершинами. Перед тем как настилать крышу, Сергей должен построить опору крыши. Опора крыши состоит из нескольких прочных балок, которые поддерживают крышу. Сергей хочет, чтобы проекция опоры крыши на пол свинарника представляла собой *скелет* многоугольника, который представляет собой свинарник.

Скелет многоугольника определяется следующим образом. Будем сжимать многоугольник, двигая его стороны внутрь с равной скоростью в направлении, перпендикулярном соответствующей стороне. По мере того, как стороны двигаются таким образом, вершины также будут двигаться внутрь многоугольника, при этом скорость перемещения вершины зависит от угла при ней. Если в процессе сжатия многоугольника две вершины совпадают, то соответствующая сторона исчезает и две соседние с ней стороны становятся соседними между собой. Скелетом многоугольника называется фигура, которую вычерчивают вершины многоугольника в процессе сжатия. Можно показать, что эта фигура состоит из нескольких отрезков. Если в результате сжатия у многоугольника остается одна сторона, то она тоже включается в его скелет.

Ниже приведен пример многоугольника и его скелета.



Фермер Сергей хочет узнать, балки какого суммарного размера ему необходимо изготовить для опоры крыши. По заданной форме свинарника выясните суммарную длину отрезков в его скелете.

### Формат входного файла

Первая строка входного файла содержит целое число  $n$  — количество вершин у основания свинарника ( $3 \leq n \leq 100$ ). Следующие  $n$  строк содержат по два целых числа — координаты вершин свинарника в порядке их обхода. Многоугольник который представляет собой выпуклый, невырожденный, никакие три его вершины не лежат на одной прямой. Координаты не превышают  $10^4$  по абсолютной величине.

### Формат выходного файла

Выведите в выходной файл одно вещественное число — суммарную длину всех ребер скелета свинарника. Ответ должен быть выведен с точностью не меньше  $10^{-6}$ .

### Примеры

<code>roof.in</code>	<code>roof.out</code>
6	17.258949429319017
0 0	
6 0	
8 3	
7 4	
2 5	
-1 1	

## Задача I. Сумма степеней

Имя входного файла: `sumpower.in`  
Имя выходного файла: `sumpower.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Мальчик Сережа очень любит изучать числовые последовательности. Он уже изучил все числовые последовательности, встречающиеся в книгах, и теперь просит всех своих знакомых придумать ему последовательность. Также он попросил придумать числовую последовательность и своего младшего брата Ромочку, который умеет считать только до шести. Разумеется, Ромочка (после долгих рассказов Сережи о том, что такое последовательность) предложил Сереже изучить последовательность 1, 2, 3, 4, 5, 6. Но Сережа уже неоднократно изучал эту последовательность, и немного изменил ее. Теперь он изучает последовательность натуральных чисел, такую, что каждый ее член  $k$  является делителем числа  $1 + 2^k + 3^k + 4^k + 5^k + 6^k$ .

Сережа посчитал несколько первых членов последовательности, после чего решил, что лучше напишет программу, которая за него посчитает член последовательности с номером  $n$ . Напишите и Вы такую программу.

### Формат входного файла

Входной файл содержит единственное число  $n$  ( $1 \leq n \leq 70$ ).

### Формат выходного файла

В выходной файл выведите искомый член последовательности.

### Примеры

<code>sumpower.in</code>	<code>sumpower.out</code>
1	1
2	3