

## 1. Кибер-Матрёшка

Суровой весной 2142 года, российскому учёному-программисту Потапу удалось изобрести искусственный интеллект! Но вскоре после изобретения Потап заметил, что его создание развивается эффективнее, если время от времени решает простые головоломки. И тогда решил Потап смастерить для своего детища Кибер-Матрёшку. Он написал простой генератор строк и предложил ИИ ответить на вопрос - содержит ли она строка другую в качестве подстроки? Да только не заладилось у ИИ с решением данной задачки и вам предстоит ему(ей) помочь.

На вход программы подается множество  $M$  из  $n$  строк, не содержащих пробельных символов. Строки нумеруются от 1 до  $n$ . Для каждой строки  $S$  из  $n$  необходимо найти все строки (их осталось  $n-1$ ), которые являются подстрокой  $S$ . Если строка  $S_1$  находится в строке  $S$  несколько раз, то нужно посчитать количество вхождений. Меньшие строки не должны пересекаться (накладываться друг на друга) в большей строке. В случае неоднозначности, когда большая строка равнозначно содержит две и более меньших строк одинаковой длины, то следует отдавать предпочтение той меньшей строке, у которой наименьший порядковый номер. Помимо этого следует учесть тот факт, что при существовании нескольких вариантов разбиения большей строки меньшими строками, верным считается тот вариант, у которого меньше количество оставшихся символов (которые не вошли ни в одну подстроку).

### Вход

В первой строке  $n$  - количество строк  $n = [1..55]$ .

Далее идут  $n$  строк длиной  $d$ , состоящих из не пробельных символов  $d = [1..1500]$

### Выход

$n$  строк, каждая из которых должна начинаться с номера  $k = [1..n]$ . После номера следует двоеточие (:) и пробел. Затем следует 0, если ни одна из  $n-1$  строк не является подстрокой строки  $k$ , или номера строк (через пробел, в порядке возрастания), которые являются подстроками строки  $k$ . В случае нескольких вхождений нужно повторить номер подстроки соответствующее количество раз.

Ограничение по времени: 30 секунд

Пример #1:

```
3
ol
lola
karolollola
```

Результат:

```
1: 0
2: 1
3: 1 1 2
```

Пример #2:

```
8
gap
pole
ole
tolek
gapole
gappole
yagapoletolekagappole
totolepoletgap
```

Результат:

1: 0  
2: 3  
3: 0  
4: 3  
5: 1 3  
6: 1 2  
7: 4 5 6  
8: 1 2 3

Пример #3:

5  
aa  
bb  
ab  
abbabaab  
aabbabaabb

Результат:

1: 0  
2: 0  
3: 0  
4: 1 2 3  
5: 1 1 2 2 3

## 2. Лишнее звено

Одну из мировых держав долгое время мучили коррупция и экономический террор. Наконец-то вышел долгожданный законопроект, предписывающий избавиться от всех посредников, которые не несут никакой экономической выгоды движению средств. В законопроекте чётко прописано: если посредник берёт товар лишь у одного посредника, и при этом передаёт товар только одному следующему посреднику, т.е. сам не является ни производителем, ни потребителем, то от него следует избавиться. Авторы законопроекта ждут экономического бума, подъема производства и наполнения казны. Однако сначала нужно вычислить посредников, а поскольку денег в казне пока нет, это решили поручить вам.

### Вход

Первая строка содержит N - число элементов торговой сети. Затем идут N строк, описывающих посредничество. В начале каждой строки идет имя посредника-поставщика, за ним - имена посредников-получателей, которым доставляет продукцию текущий посредник-поставщик. Все имена разделены пробелами. Торговая сеть не содержит циклов.

### Выход

Сначала выводятся имена лишних посредников, каждый с новой строки, упорядоченные по алфавиту. Затем в отдельной строке выводится символ "-", после чего выводится информация об оставшихся посредниках таким же образом, которым производился ввод информации. Имена на выходе должны быть упорядочены по алфавиту.

Пример #1:

5

1 2  
2 3 4  
4 5  
3  
5

Результат:

4  
-  
1 2  
2 3 5  
3  
5

Пример #2:

12  
vasa pete vanya  
pete sasha  
sasha masha katya lena  
masha jora lera  
jora vanya  
vanya  
katya lera  
lera vova  
vova  
lena misha  
misha vova  
jeka sasha

Результат:

jora  
katya  
lena  
misha  
pete  
-  
jeka sasha  
lera vova  
masha lera vanya  
sasha lera masha vova  
vanya  
vasa sasha vanya  
vova

### 3. Цепная реакция

В своем дневнике "Моё нечаянное путешествие. Туда и Обратно и что случилось После" хоббит Бильбо Бэггинс описал историю, которая по счастливой случайности чуть не закончилась трагедией. Однажды, когда гномы

устроили привал на ночь и по своему обыкновению зажарили несколько сочных кабанчиков, выпили пару бочонков хмельного эля и уже хотели ложиться спать, прибежали часовые Кили и Фили. Они сообщили, что недалеко от лагеря была замечена большая стая диких варгов. Гном Бомбур, славящийся своими пиротехническими навыками, предложил покинуть и заминировать лагерь, чтобы испугать и сбить со следа варгов. Он нарисовал карту местности размером  $N * N$  клеток. Затем он рассчитал количество зарядов взрывчатки  $K$ , их координаты на карте (номер строки  $R$ , номер столбца  $C$ , нумерация начинается с 1) и мощности. Мощность взрывчатки обозначается некоторой цифрой, например  $L$ , и означает, что при взрыве клетка, в которой заложен заряд, а также  $L-1$  клеток слева, справа, сверху и снизу будут взорваны, причем, если в любой соседней клетке содержится заряд, то он тоже взрывается – так начинается цепь взрывов. Каждый заряд взрывается только один раз. Когда все заряды были расположены по местам, Бомбур понял, что забыл установить детонаторы на зарядах, но времени почти не осталось - стая уже близко. Вам предстоит рассчитать минимальное количество детонаторов, необходимое, чтобы взорвались все заложенные заряды. Судьба похода и всего Средиземья в Ваших руках!

Вход:

$N$   
 $K$   
 $R1 C1 L1$   
 $R2 C2 L2$   
...  
 $Rk Ck Lk$

Выход:

минимальное количество детонаторов

Пример #1:

5  
5  
3 5 1  
3 1 3  
4 4 2  
4 2 2  
3 2 2

Результат:

3

Пример #2:

7  
5  
7 5 1  
6 4 4  
1 3 3  
1 4 1  
7 4 4

Результат:

2

Пример #3:

10  
10  
10 7 5  
10 5 3  
9 3 4  
5 5 2  
4 4 4  
10 4 2  
9 5 5  
10 8 4  
9 1 1  
9 8 1

Результат:

2

#### 4. Поиски самолёта авиакомпании Malaysia Airlines

Для того, чтобы отыскать пропавший самолёт авиакомпании Malaysia Airlines была предпринята попытка задействовать спутник, для дискретного сканирования области возможного местонахождения потерпевшего крушение лайнера. Первые поиски дали ошеломляющий результат: оказывается, заданный район - **огромный коралловый риф, в котором мог затеряться самолёт**.

Если верить словам капитана Фербера, что: "Красивый самолёт летает хорошо, а некрасивый - плохо", то лайнер авиакомпании Malaysia Airlines, был не таким уж и красивым самолётом. По этой причине предположили, что силуэт самолёта может быть представлен шестью кусками, т.е. любым из вариантов:

```
.....  
..%.. ..%.. ..%... ..%...  
.%%%. ..%.. .%%%. .%%%.  
..%.. .%%%. ..%... ..%...  
..%.. ..%.. .....  
.....
```

Вам предстоит написать программу, которая будет обнаруживать присутствует ли силуэт пропавшего самолёта (самолетов) на снятом участке поверхности. Спутник каждый раз фотографирует участок поверхности моря и подаёт снимок на вход программе.

Следует также заметить, что заказчики программы попросили учесть тот факт, что самолёт может лежать на краю захваченного спутником участка поверхности планеты. В таком случае должно учитываться условие, что если самолёт находится на краю, но не в углу, то для полного силуэта может не хватать одной детали самолёта (где каждая отдельная возможная деталь обозначается на снимке знаком '%'). Если самолёт находится в углу снимка, то может не хватать сразу двух деталей, по одной детали на каждое из направлений (слева или справа, и сверху или снизу). На выходе программа должна печатать координаты возможного нахождения самолёта. Начала координат расположены в верхнем левом углу представленного снимка. Каждый самолёт представлен координатами его крестовины (пересечения крыльев и корпуса). Координаты всех



Результат:

9, 0  
2, 2  
8, 4  
9, 8  
6, 9  
8, 9  
9, 9

## 5. Босиком по радуге

Любимая игра 10й старшей группы детского сада "Радуга-дуга" - это Попрыгушки по разноцветным дорожкам. В этом году ребята устроили турнир на звание "Главного прыгуна", предварительно выбрав семерых самых достойных участников. По правилам каждый прыгун выбирает себе цвет и может прыгать только по плиткам выбранного цвета и только вперед по диагонали. Выигрывает тот, кто сможет допрыгать до конца дорожки. Если же у кого-то из участников обрывается цветовая диагональ, то он выбывает из соревнования. Все дорожки, по которым прыгают участники, имеют ширину в 7 плиток и раскрашены в цвета радуги в хаотичном порядке. Итак, семеро смелых вышли на старт. Кому же достанется звание Главного прыгуна?

**Вход**

В первой строке число  $L$  - длина плиточного поля (ширина поля  $W$  равна семи плиткам). Далее на отдельных строках перечислены имена участников соревнования. Далее идут  $W$  строк по  $L$  символов в каждой. При этом символы друг от друга отделены пробелами. Каждый символ отвечает за цвет плитки, 'r' - красный, 'o' - оранжевый, 'y' - желтый, 'g' - зеленый, 'l' - голубой, 'b' - синий, 'v' - фиолетовый.

**Выход**

Имя победителя (гарантируется, что победитель всегда один, и что он всегда есть).

Пример #1:

Вход:

```
11
Nikita
Vova
Denis
Andrey
Galya
Valeriya
Lena
r l r l r y r y r b l
y r l r y r y r y r b
o y o y o l o l b y b
g o y o v o l y l b y
b v y v r v y o r v g
v l v y g y v g v g r
l o l g y g g v g r l
```

Результат:

Vova

Пример #2:

Вход:

```
15
Igor
Rita
Sonya
Masha
Misha
Kolya
Petya
v b r g b v r v b g b r b g b
g o g o g o v o g o g o g o y
o g o v o g o g o v o g o b o
b l v r y v g r y r y r y v r
l b r y r y r y r y r y r y v
y r y b o r y v o v o v o r o
r y b o b o v o v o v o v o y
```

Результат:

Sonya

## 6. Паровозики

После внеочередных порций громких разоблачений спецслужбами было установлено точное расположение Эдварда Сноудена. Он подключился к общественной сети на одном из жд вокзалов и передавал секретные данные. После отправки всех секретных архивов Эдвард решил скрыться, уехав так далеко, как только позволит его материальное положение. На его поиски брошены лучшие силы. Вам предстоит определить все возможные варианты его местонахождения, исходя из оставшихся средств на его банковской карте.

### Вход

В первой строке записывается станция, где был обнаружен Эдвард. Во второй строке - количество средств на его банковской карте. Далее следует информация о районной железнодорожной сети: количество жд станций, названия жд станций (на отдельных строках), матрица стоимости пути от одной станции до другой.

Матрица стоимости пути получена таким образом:

- строки соответствуют  $i$ -ым станциям
- столбцы соответствуют  $j$ -ым станциям
- двигаться можно только с  $j$ -ой станции на  $i$ -тые станции за указанную стоимость.
- если стоимость движения равна нулю, значит между соответствующими станциями либо вообще нет пути, либо нельзя добраться без пересадок.

### Выход

Названия самых дальних станций, до которых может добраться Эдвард за имеющиеся у него средства. Порядок вывода станций такой же как их порядок на входе.

Пример #1:

Вход:

Mostovaya  
55  
5  
Mostovaya  
Kozhevino  
Ermlino  
Rotor  
Kotelnya  
0 0 0 0 0  
39 0 0 0 0  
16 0 0 0 0  
0 0 34 0 0  
0 0 51 0 0

Результат:  
Kozhevino  
Rotor

Пример #2:

Вход:  
Svetloe  
30  
7  
Sortirovochnaya  
Svetloe  
Prigorki  
Pahota  
Kanal  
Kishi  
Kosmos  
0 0 0 0 0 0 0  
17 0 0 0 0 0 0  
0 31 0 0 0 0 0  
0 28 0 0 0 0 0  
45 0 0 0 0 0 0  
54 0 0 0 0 0 0  
0 0 0 0 0 9 0

Результат:  
Pahota

## 7. Обед

Все хорошо в Тушино - новый корпус, просторные помещения, лес рядом и бассейн. Одно плохо - в столовой всегда мало еды, преподавателями и студентам часто не удается пообедать. Особенно тяжело преподавателю

на лекции - ведь он затрачивает много энергии, стараясь разбудить студентов. Если отпустишь их пораньше, это идет в ущерб предмету, отпустишь позже - другие группы и преподаватели все съедят.

Вам нужно рассчитать оптимальное время окончания лекции - самое позднее время до официального окончания (или равное ему) (14:05), отпустив группу в которое, преподаватель успевает пообедать. После ухода группы преподавателю нужно еще 2 мин, чтобы закрыть класс. Если какая-то группа людей численностью  $x$  пришла в столовую одновременно, а там осталось всего  $y$  порций ( $x > y$ ), то считается, что никому из них не удалось пообедать. Если группу обычно отпускают раньше, но в этот раз есть возможность отпустить её позже, то нужно отпустить позже.

### **Вход**

В первой строке идет  $n$  - целое число порций обеда, которые есть в столовой ( $n < 1000$ ). Во второй строке целое число  $m$  - количество групп, которые занимаются ( $m < 10$ ). Далее идут  $m$  строк, описывающих группу. Описание группы состоит из  $k$  - количества студентов ( $k < 31$ ),  $t$  - расстояния от класса до столовой в минутах ходьбы ( $t < 11$ ),  $d$  - момент времени в формате hh:mm, в который обычно эту группу отпускают ( $13:44 < d < 14:06$ ). В последней строке идет целое число  $z$  - номер группы (нумерация начинается с 1), преподавателю которой нужно помочь.

### **Выход**

Оптимальное время, когда преподаватель должен отпустить группу в формате hh:mm.

#### **Пример #1:**

##### **Вход**

```
30
3
9 2 14:05
9 3 14:05
9 5 14:05
3
```

##### **Выход**

```
14:05
```

#### **Пример #2:**

##### **Вход**

```
30
3
9 2 14:05
9 3 14:05
10 5 14:05
3
```

##### **Выход**

```
14:02
```

## **8. Индивидуальный план**

Жизнь одного преподавателя сложилась вполне удачно: предмет ему всегда нравился, и вдруг студенты начали по-настоящему учиться, а тут еще открыли вторую столовую и зарплату неожиданно повысили в 10 раз. Одно плохо - каждый год надо заполнять Индивидуальный план. Это такой документ, где перечислены основные виды деятельности преподавателя - чтение лекций, проведение лабораторных, проверка домашних работ, прием

экзамена и т.д. По каждому виду деятельности нужно рассчитать часы, следуя методичке Минобрнауки. Видов деятельности в методичке очень много. Видов деятельности у преподавателя тоже много, хотя и меньше, чем в методичке. Проблема в том, что часов в итоге должно получиться не слишком много и не слишком мало.

Напишите программу, которая рассчитает Индивидуальный план. Если получается слишком много часов, то нужно выкидывать какие-то виды деятельности, так, чтобы уложиться в заданный диапазон. Если существует несколько способов заполнить Индивидуальный план, нужно выбрать тот, который дает больше часов.

#### **Вход**

В первой строке идет  $n$  - число типов деятельности, заложенных в методичке Минобрнауки ( $1 \leq n \leq 1000$ ). Далее идет строка в которой через пробел перечислены количества часов  $h[i]$ , заложенных на тип деятельности  $i$ ,  $i$  от 1 до  $n$ ,  $0 < h[i] < 101$ .

В следующей строке  $m$  - число типов деятельности, которые умеет делать преподаватель ( $m < n$ ).

В следующей строке  $m$  чисел, разделенных пробелами - номера типов деятельности преподавателя, соответствующих методичке.

В последней строке два целых числа  $min$  и  $max$ , разделенных пробелами - минимальное и максимальное количество часов.

#### **Выход**

Оптимальное количество часов. В интервале  $[min, max]$  ( $0 < min, max < 100$ ). Если не получается заполнить план, то вывести -1.

#### **Пример #1:**

##### **Вход:**

```
10
3 3 5 6 7 8 1 8 9 5
4
1 3 5 9
8 10
```

##### **Выход:**

```
10
```

#### **Пример #2:**

##### **Вход:**

```
10
3 3 5 6 7 8 1 8 9 5
4
1 3 5 9
30 35
```

##### **Выход:**

```
-1
```

## 9. LRF

Все течет, все меняется. Где прошло 5 лет, там и 10, где 10, там и 20. К 2034 году все осознали важность информационных технологий и открытых систем. Открылся факультет linux-систем, на который принимают только после сдачи профильного ЕГЭ по linux или при наличии значка "Готов к компиляции ядра" 1 степени. Первые три года студенты учатся уму-разуму: языкам программирования, процессу разработки, основным

алгоритмам, структурам данных и подходам, а затем разбредаются по кафедрам и в ходе курсовых и дипломных проектов находят свое место в Open Source мире - кто-то тестирует и разрабатывает ядро, кто-то - драйверы для новых транспортных систем, кто-то - стандартные библиотеки для новых языков программирования, кто-то совершенствует СУБД. Диплом выдается только тому, кто сделал 12 коммитов кода в основную ветку проекта, подтвержденных мэйнтейнером, или закрыл 12 багов.

Вся бюрократия - учебные планы, индивидуальные планы, методички давно отменена и все забыли, как её нужно считать. Даже оценки перестали ставить - учится только тот, кто хочет.

Однако в 2035 году пришло время понять, какой проект пользуется самой большой популярностью среди студентов. Студент может учиться на кафедре ядра, или на кафедре файловых систем, но заниматься он может любым проектом, например графической библиотекой для fortran2030.

#### **Вход**

В первой строке идет число  $n$  - количество групп студентов ( $0 < n < 1000$ ). Далее идет описание групп. Для каждой группы в первой строке задается количество  $m$  студентов в ней ( $0 < m < 100$ ). Далее идет  $m$  строк, в каждой из которых фамилия студента и проект, которым он(она) занимается - через пробел.

#### **Выход**

Вывести распределение студентов по проектам, начиная с самого популярного проекта, и заканчивая самым забытым в формате: название проекта, количество студентов, процент от общего числа. Эти три поля должны быть разделены пробелами. Проценты выводить до одного знака после точки: 10.3%. Если какие-то проекты одинаково популярны, вывести их по алфавиту.

#### **Пример #1:**

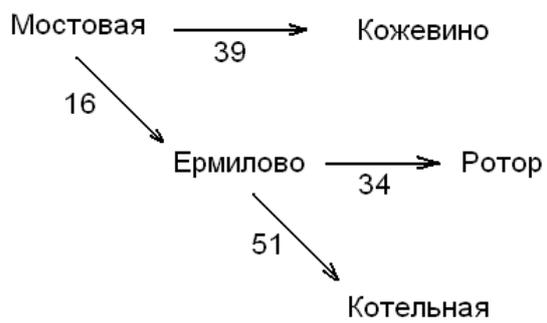
##### **Вход:**

```
2
3
Ivanov c++22
Petrenko perl
Sidorevich z++
5
Nazarov      perl
Vykanov perl
Dremlev ruberoid
Zvi-Mark ProstoDB
Keylow  c++22
```

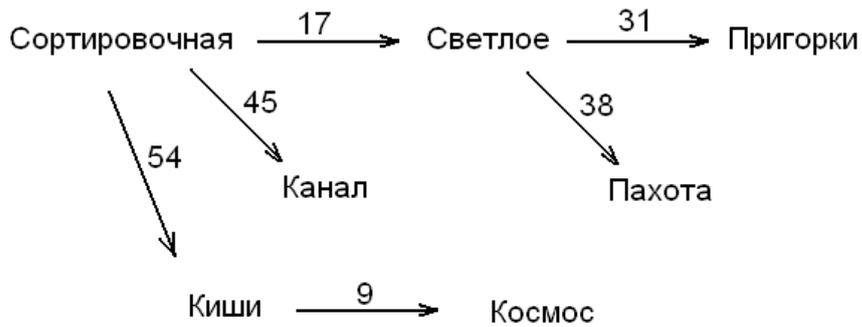
##### **Выход:**

```
perl 3 37.5%
c++22 2 25.0%
ProstoDB 1 12.5%
ruberoid 1 12.5%
z++ 1 12.5%
```

## Пояснения к задаче «Паровозики»



Откуда →	Mostovaya	Kozhevino	Ermilovo	Rotor	Kotelnaya
Куда ↓	Mostovaya	Kozhevino	Ermilovo	Rotor	Kotelnaya
Mostovaya	0	0	0	0	0
Kozhevino	39	0	0	0	0
Ermilovo	16	0	0	0	0
Rotor	0	0	34	0	0
Kotelnaya	0	0	51	0	0



Откуда →	Sortirovochnaya	Svetloe	Prigorki	Pahota	Kanal	Kishi	Kosmos
Куда ↓	Sortirovochnaya	Svetloe	Prigorki	Pahota	Kanal	Kishi	Kosmos
Sortirovochnaya	0	0	0	0	0	0	0
Svetloe	17	0	0	0	0	0	0
Prigorki	0	31	0	0	0	0	0
Pahota	0	28	0	0	0	0	0
Kanal	45	0	0	0	0	0	0
Kishi	54	0	0	0	0	0	0
Kosmos	0	0	0	0	0	9	0