1. Дерево каталогов

1 секунда 20 очков

Написать программу, которая получает описание дерева каталогов, где вложенность каталогов показана отступами, и выводит представление этого дерева, где для большей наглядности помимо отступов используются псевдографические символы.

Входные данные. На первой строке текстового файла `kata.sis` дано количество каталогов в дереве N ($1 \leqslant N \leqslant 1000$) а на каждой из следующих N строк — имя каталога, перед которым также может быть состоящий из пробелов отступ. Каждый каталог с отступом является дочерним каталогом ближайшего предыдущего каталога с меньшим отступом. Можно предполагать, что ни одно имя каталога не длиннее 50 знаков, а ни одна стока файла (включая отступ) не длиннее 250 знаков. Имя каталога не может содержать пробелов.

Выходные данные. В текстовый файл `kata.val` вывести в точности N строк: представление того же самого дерева каталогов, где шаг отступа равен в точности двум пробелам, а вложенность каталогов изображена графически, как в примере, приведенном ниже. Для представления вертикального отрезка линии использовать символ `|` (код 124), для горизонтальных отрезков символ `-` (код 45), в месте ответвления линий символ `+` (код 43), а в месте сгиба линии (без дополнительных ответвлений) символ `*` (код 42). Каталоги должны быть перечислены в том же порядке, в каком они были заданы во входном файле.

Пример.

```
kata.sis          kata.val
3                 Alpha
Alpha             *-Bravo
 Bravo              *-Charlie
  Charlie
```

Пример.

```
kata.sis          kata.val
8                 Alpha
Alpha             +-Bravo
 Bravo            | +-Charlie
  Charlie         | +-Delta
  Delta           | *-Echo
  Echo            *-Foxtrot
 Foxtrot          Golf
Golf              *-Hotel
    Hotel
```

Оценивание. В тестах с суммарной стоимостью 10 очков шаг отступа во входных данных равен в точности одному пробелу.

2. Марафон

1 секунда 40 очков

Зрительные места у трассы, где проводится марафон, пронумерованы $1 \ldots N$ по направлению от старта к финишу. Часть мест уже занята. K друзей хотят пойти смотреть соревнование вместе, и при этом находиться как можно ближе друг к другу.

Написать программу, которая подберет для друзей наилучшее расположение среди свободных мест. Оценкой качества расположения друзей является сумма разниц номеров занятых мест для каждой пары друзей в группе (чем она меньше — тем лучше).

Входные данные. На первой строке текстового файла `mare.sis` дано общее количество зрительных мест N ($1 \leqslant N \leqslant 1\,000\,000$), количество уже занятых мест M ($0 \leqslant M < N$), и количество друзей K ($1 \leqslant K \leqslant N - M$). На второй строке дано M разделенных пробелами чисел: номера уже занятых зрительных мест в порядке возрастания.

Выходные данные. На единственной строке текстового файла `mare.val` вывести два целых числа: минимальный и максимальный номера мест, которые должны занять друзья. Если найдется несколько одинаково хороших расположений, вывести то из них, которое ближе к финишу (где используются большие номера мест).

Пример.

mare.sis	mare.val
8 4 3 2 3 6 8	4 7

Друзья должны купить билеты на места 4, 5 и 7 (место 6 уже занято). Для мест 4 и 5 разница номеров равна 1, для мест 5 и 7 разница равна 2, а для мест 4 и 7 разница равна 3. Сумма всех трех попарных разниц (которая и является оценкой качества расположения друзей) равна 6.

Оценивание. В тестах с суммарной стоимостью 20 очков $N \leqslant 30\,000$ и $K \leqslant 300$. Среди них, в тестах с суммарной стоимостью 10 очков $N \leqslant 1000$ и $K \leqslant 100$.

3. Сочетания

1 секунда 40 очков

Как известно из комбинаторики, число подмножеств размером K элементов у множества, содержащего N элементов, равно

$$C(N,K) = \frac{N!}{K! \cdot (N-K)!},$$

где $N!$ обозначает произведение $1 \cdot 2 \cdot \ldots \cdot (N-1) \cdot N$, а также по определению $0! = 1$.

Написать программу, которая для заданных целых чисел N, K и M находит максимальное целое число L, при котором $C(N,K)$ делится на M^L.

Входные данные. На первой строке текстового файла `komb.sis` находятся целые числа N, K и M ($0 \leqslant K \leqslant N < 2^{63}$, $2 \leqslant M \leqslant 10^{12}$).

Выходные данные. На единственную строку текстового файла `komb.val` вывести искомое L. Можно считать, что ответ попадает в область значений 64-битного целого числа со знаком.

Пример.

`komb.sis`	`komb.val`
`6 3 2`	`2`

$C(6,3) = 6!/(3! \cdot 3!) = 720/(6 \cdot 6) = 20$, что делится на $2^2 = 4$, но не делится на $2^3 = 8$.

Оценивание. В тестах с суммарной стоимостью 20 очков $C(N,K) < 2^{63}$. Из них в тестах с суммарной стоимостью 10 очков $N! < 2^{63}$.

1. Филателист

1 секунда 20 очков

У филателиста возникло подозрение, что среди его марок есть повторяющиеся экземпляры. Так как марок много, то искать среди них повторяющиеся вручную тяжело. Вместо этого филателист выложил все марки на стол в несколько рядов, сфотографировал цифровым фотоаппаратом, и надеется теперь использовать для поиска повторов компьютер.

Написать программу, которая найдет на фотографии все комплекты идентичных марок.

Входные данные. На первой строке текстового файла `filg.sis` дано количество рядов R ($1 \leqslant R \leqslant 50$) и колонок V ($1 \leqslant V \leqslant 10$) марок, а также высота K ($1 \leqslant K \leqslant 30$) и ширина L ($1 \leqslant L \leqslant 30$) изображения каждой марки. На каждой из следующих $K \cdot R$ строк записано в точности $L \cdot V$ символов. Все это представляет из себя изображение выложенных на столе друг рядом с другом марок, где каждой марке соответствует блок из $K \times L$ символов. Две марки считать одинаковыми, если их изображения идентичны (если при поточечном сопоставлении изображений двух марок, хотя бы один символ различается, марки нужно считать разными). Большие и маленькие буквы считать разными символами.

Выходные данные. В текстовый файл `filg.val` вывести найденные наборы повторяющихся марок. Каждую марку следует выводить в виде координат ее ряда и колонки на столе. Ряды нумеруются сверху вниз $1 \ldots R$, а колонки — слева направо $1 \ldots V$. Координаты всех одинаковых марок вывести одну за другой на одной строке. Координаты разных марок должны оказаться на разных строках. Порядок перечисления наборов и марок в каждом наборе не имеет значения.

Пример.

```
filg.sis
1 3 8 8
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
```

```
filg.val
1 1
1 2
1 3
```

Все марки различны, поэтому каждая марка выведена на отдельной строке.

Пример.

```
filg.sis
2 3 8 8
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
dddddddddaaaaaaaabbbbbbbb
dddddddddaaaaaaaabbbbbbbb
dddddddddaaaaaaaabbbbbbbb
ddddddddaaaxxaaabbbbbbbb
ddddddddaaaxxaaabbbbbbbb
dddddddddaaaaaaaabbbbbbbb
dddddddddaaaaaaaabbbbbbbb
dddddddddaaaaaaaabbbbbbbb
```

```
filg.val
1 1
1 2 2 3
1 3
2 1
2 2
```

Марка в 1-м ряду на 2-й колонке такая же, как и марка во 2-м ряду на 3-й колонке, поэтому они выведены на одной строке.

2. Последовательность

1 секунда 30 очков

Написать программу, которая принимает N натуральных чисел и находит максимальное число, которое можно получить, записав данные числа друг за другом в каком-то порядке.

Входные данные. На первой строке текстового файла `jada.sis` находится целое число N ($1 \leqslant N \leqslant 1000$), а на второй строке находятся N разделённых пробелами неотрицательных целых чисел со значениями до 1 000 000.

Выходные данные. На единственную строку текстового файла `jada.val` вывести искомое максимальное число.

Пример.

jada.sis	jada.val
3 10 3 4	4310

Оценивание. В тестах с суммарной стоимостью 10 очков $N \leqslant 5$ и сумма длин данных чисел не превышает 19.

3. Делители

1 секунда 50 очков

Написать программу, которая находит наименьшее положительное целое число, у которого есть ровно K различных делителей.

Входные данные. На единственной строке текстового файла `jaga.sis` находится положительное целое число K.

Выходные данные. На единственную строку текстового файла `jaga.val` вывести искомое число. Можно считать, что ответ попадает в область значений 64-битного целого числа со знаком.

Пример.

jaga.sis	jaga.val
4	6

Число 6 делится на 1, 2, 3, 6, и легко проверить, что у всех меньших положительных целых чисел меньше различных делителей.

Оценивание. В тестах с суммарной стоимостью 25 очков ответ не превышает 1 000 000. Из них в тестах с суммарной стоимостью 15 очков ответ не превышает 10 000.

1. Радиовещание

1 секунда 20 очков

Телекоммуникационная фирма использует для распространения сигнала набор передатчиков. Приёмник клиента автоматически настраивается на тот передатчик, сигнал которого наиболее сильный в точке расположения приёмника. Как известно из физики, интенсивность сигнала падает пропорционально квадрату расстояния между передатчиком и приёмником.

Написать программу, которая принимает местоположения и мощности всех передатчиков и определяет передатчик, на который настроится приёмник, расположенный в заданной точке.

Входные данные. На первой строке текстового файла `levi.sis` находятся координаты клиента X и Y. На второй строке файла — число передатчиков N ($1 \leqslant N \leqslant 1000$). В каждой из следующих N строк находятся координаты соответствующего передатчика X_i и Y_i, а также его мощность P_i. Все координаты — целые числа с абсолютным значением не более 10 000, а мощности передатчиков — положительные целые числа, не превыщающие 1000. Можно считать, что клиент не находится в одной точке ни с одним передатчиком.

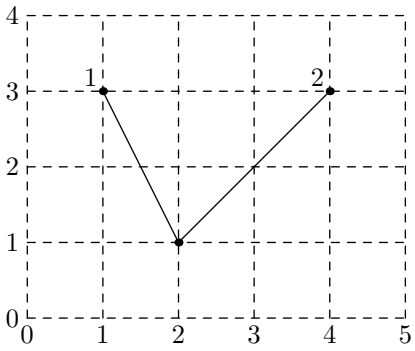
Выходные данные. На единственную строку текстового файла `levi.val` вывести номер того передатчика, сигнал которого наиболее сильный в точке расположения клиента. Если передатчиков с максимальным сигналом несколько, вывести любой из них. Передатчики пронумерованы числами $1 \ldots N$ в порядке их появления во входных данных.

Пример.

`levi.sis`
```
2 1
2
1 3 10
4 3 24
```

`levi.val`
```
2
```

На рисунке снизу видно, что расстояние от клиента до первого передатчика равно $\sqrt{5}$, и поэтому интенсивность его сигнала равна $10/5 = 2$, а расстояние до второго передатчика равно $\sqrt{8}$, и поэтому интенсивность его сигнала равна $24/8 = 3$. Следовательно приёмник настроится на второй передатчик, хоть он и дальше.



Пример.

`levi.sis`
```
2 1
2
1 3 12
4 3 20
```

`levi.val`
```
2
```

2. Филателист

1 секунда 40 очков

У филателиста возникло подозрение, что среди его марок есть два одинаковых. Так как марок много, то искать среди них повторяющиеся вручную тяжело. Вместо этого филателист выложил все марки на стол в несколько рядов, сфотографировал цифровым фотоаппаратом, и надеется теперь использовать для поиска повторов компьютер.

Написать программу, которая проанализирует фотографию и удостоверится, что все марки в наборе уникальны, или, в противном случае, укажет на пару одинаковых марок.

Входные данные. На первой строке текстового файла `filp.sis` дано количество рядов R ($1 \leqslant R \leqslant 50$) и колонок V ($1 \leqslant V \leqslant 10$) марок. На каждой из следующих $8 \cdot R$ строк записано в точности $8 \cdot V$ символов. Все это представляет из себя изображение выложенных на столе друг рядом с другом марок, где каждой марке соответствует блок из 8×8 символов. Две марки считать одинаковыми, если их изображения идентичны (если при поточечном сопоставлении изображений двух марок, хотя бы один символ различается, марки нужно считать разными). Большие и маленькие буквы считать разными символами.

Выходные данные. Если среди марок нет повторяющихся, в текстовый файл `filp.val` вывести на единственной строке слово `POLE`. Если же найдется пара одинаковых марок, вывести на первой строке координаты первой марки в паре, а на второй строке — координаты второй марки. Координаты каждой марки указать в виде двух чисел — ряда и колонки марки на столе. Ряды нумеруются сверху вниз $1 \ldots R$, а колонки — слева направо $1 \ldots V$. Можно предполагать, что либо все марки уникальны, либо среди них есть в точности одна пара одинаковых марок.

Пример.

`filp.sis`

```
1 3
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
```

`filp.val`

```
POLE
```

Все марки различны.

Пример.

`filp.sis`

```
2 3
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
aaaaaaaabbbbbbbbcccccccc
ddddddddaaaaaaaabbbbbbbb
ddddddddaaaaaaaabbbbbbbb
ddddddddaaaaaaaabbbbbbbb
ddddddddaaaxxaaabbbbbbbb
ddddddddaaaxxaaabbbbbbbb
ddddddddaaaaaaaabbbbbbbb
ddddddddaaaaaaaabbbbbbbb
ddddddddaaaaaaaabbbbbbbb
```

`filp.val`

```
1 2
2 3
```

Марка в 1-м ряду на 2-й колонке такая же, как и марка во 2-м ряду на 3-й колонке.

3. Марафон

1 секунда 40 очков

Зрительные места у трассы, где проводится марафон, пронумерованы $1 \ldots N$ по направлению от старта к финишу. Часть мест уже занята. K друзей хотят пойти смотреть соревнование вместе, и при этом находиться как можно ближе друг к другу.

Написать программу, которая подберет для друзей наилучшее расположение среди свободных мест. Оценкой качества расположения друзей является разница минимального и максимального номеров занятых группой мест (чем она меньше — тем лучше).

Входные данные. На первой строке текстового файла `marp.sis` дано общее количество зрительных мест N ($1 \leqslant N \leqslant 1\,000\,000$), количество уже занятых мест M ($0 \leqslant M < N$), и количество друзей K ($1 \leqslant K \leqslant N - M$). На второй строке дано M разделенных пробелами чисел: номера уже занятых зрительных мест в порядке возрастания.

Выходные данные. На единственной строке текстового файла `marp.val` вывести два целых числа: минимальный и максимальный номера мест, которые должны занять друзья. Если найдется несколько одинаково хороших расположений, вывести то из них, которое ближе к финишу (где используются большие номера мест).

Пример.

`marp.sis`	`marp.val`
`8 4 3` `2 3 6 8`	`4 7`

Друзья должны купить билеты на места 4, 5 и 7 (место 6 уже занято).

Оценивание. В тестах с суммарной стоимостью 30 очков $N \leqslant 30\,000$ и $K \leqslant 300$. Среди них, в тестах с суммарной стоимостью 20 очков $N \leqslant 3000$. А среди них, в свою очередь, в тестах с суммарной стоимостью 10 очков $N \leqslant 300$.