1. Разбиение прямоугольника

1 секунда 20 очков

Написать программу, которая разделяет состоящий из $N \times M$ клеток прямоугольник на K кусочков так, что каждый кусочек в свою очередь является прямоугольником площадью минимум в 2 клетки. Найденные K кусочков должны закрывать весь прямоугольник, не должно остаться ни одной клетки.

Входные данные. В первой строке текстового файла `rt.sis` дано количество рядов N ($1 \leqslant N \leqslant 100$) и столбцов M ($1 \leqslant M \leqslant 100$) прямоугольника. Во второй строке файла указано количество кусочков K ($1 \leqslant K \leqslant \frac{N \cdot M}{2}$).

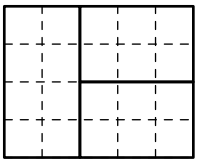
Выходные данные. В текстовой файл `rt.val` вывести ровно N строк, в каждой ровно M целых чисел $1 \ldots K$ так, что при соединении клеток, находящихся на позициях, указанных одним и тем же числом, получилось бы требуемое разбиение. Если возможных разбиений несколько, вывести любое из них. Можно предполагать, что в каждом тесте существует как минимум одно возможное разбиение.

Пример.

`rt.sis`
```
4 5
3
```

`rt.val`
```
1 1 2 2 2
1 1 2 2 2
1 1 3 3 3
1 1 3 3 3
```

Соответствующее примеру разбиение:



2. Автобусные билеты

1 секунда 40 очков

Автобусная фирма продаёт T различных типов билетов, каждый из которых действителен определённое количество дней, начиная с дня покупки, и имеет определённую цену. Пассажиру известно про следующие N дней, в какие дни он будет ездить на автобусе, а в какие нет. Написать программу, которая находит, когда и какие билеты следует купить пассажиру, чтобы совершить все необходимые поездки за минимальную цену.

Входные данные. В первой строке текстового файла `bp.sis` дано количество типов билетов T ($1 \leqslant T \leqslant 100$), и в каждой из следующих T строк задано описание одного типа в виде K_i H_i ($1 \leqslant K_i \leqslant 100$, $1 \leqslant H_i \leqslant 100$), где K_i означает длительность в днях i. го типа билетов, а H_i — его цену. Известно, что все типы билетов имеют различные сроки действия.

В следующей строке файла дана длительность плана поездок N ($1 \leqslant N \leqslant 10\,000$), а в следующей строке — N целых чисел, где 1 обозначает день, в который пассажир собирается ехать на автобусе, и 0 обозначает день, в который он не собирается ехать на автобусе.

Выходные данные. В первую строку текстового файла `bp.val` вывести количество покупаемых билетов M и их суммарную стоимость H. В каждую из следующих M строк вывести два целых числа P_i и T_i ($1 \leqslant P_i \leqslant N$, $1 \leqslant T_i \leqslant T$), которые означают, что в день P_i пассажир покапает билет типа T_i (который будет действителен в дни $P_i \ldots P_i + K_{T_i} - 1$). Если планов с минимальной стоимостью несколько, вывести любой из них. Билеты выводить в порядке их покупки.

Пример.

`bp.sis`
```
3
1 4
2 6
5 12
6
1 1 0 0 1 0
```

`bp.val`
```
2 10
1 2
5 1
```

3. Странствующий политик

1 секунда 40 очков

В ходе предвыборной кампании политику необходимо посетить N городов, каждый ровно один раз, и вернуться домой. Его партия согласна оплатить только K билетов на самолёт, остальные он должен оплатить сам. Естественно, политик собирается предоставить партии K самых дорогих билетов и сам заплатить за самые дешёвые.

Написать программу, которая составит политику маршрут, при котором ему придётся заплатить из своего кармана как можно меньше.

Входные данные. В первой строке текстового файла `rp.sis` дано количество посещаемых городов N ($3 \leqslant N \leqslant 11$) и количество оплачиваемых партией билетов K ($0 \leqslant K \leqslant N$). Города обозначены числами $1 \ldots N$, и политик живёт в 1-ом городе.

Во второй строке файла дано количество авиалиний M ($N \leqslant M \leqslant \frac{N \cdot (N-1)}{2}$), и в каждой из следующих M строк дано три целых числа: описание одной авиалинии в виде A_i B_i C_i ($1 \leqslant A_i, B_i \leqslant N$, $0 \leqslant C_i \leqslant 10\,000$), которое означает, что i-ая линия направлена из города A_i в город B_i и цена билета за этот рейс C_i. Известно, что между любыми двумя городами не более одной линии, и на всех линиях самолёты летят в обе стороны.

Выходные данные. В текстовой файл `rp.val` вывести ровно $N + 1$ строк: обозначения городов в порядке их посещения. Маршрут должен начинаться и заканчиваться в городе номер 1, каждый из оставшихся городов посещать ровно один раз и стоить политику как можно меньше. Если маршрутов с минимальной стоимостью несколько, вывести любой из них. Можно предполагать, что в любом тесте найдётся хотя бы один маршрут, проходящий через все города.

Пример.

`rp.sis`	`rp.val`
`4 2`	`1`
`6`	`2`
`1 2 4`	`3`
`1 3 2`	`4`
`1 4 4`	`1`
`2 3 1`	
`2 4 2`	
`3 4 1`	

Другая возможность была бы пройти тот же маршрут в противоположном порядке: $1 \to 4 \to 3 \to 2 \to 1$. Маршрут $1 \to 3 \to 2 \to 4 \to 1$ был бы в целом дешевле, но в этом случае политику пришлось бы больше заплатить из своего кармана.