

## Задача 2. UP

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	10 секунд
Максимальная оценка:	35 баллов

Для того, чтобы выдержать собеседование при приеме на работу в качестве программиста в некоторую компьютерную фирму, претендент должен уметь решать задачи по следующим четырем темам: динамическое программирование, перебор с возвратом, теория графов и вычислительная геометрия. Полностью подготовиться к тестовому собеседованию можно, решая задачи из пособия, изданного этой фирмой.

Уровень подготовки программиста по каждой из перечисленных тем оценивается своим индексом UP по  $L$ -балльной шкале от 1 до  $L$ . Для каждой из задач определены минимальные UP программиста по каждой из тем, необходимые для того, чтобы он смог решить данную задачу. Определены также UP, которых программист достигнет, решив ее. Если при этом по какой-то из тем он уже имел более высокий UP, то этот уровень не понизится. На решение задачи, повышающей хотя бы один из UP, программист тратит 2 часа, в противном случае - 1 час.

Требуется разработать учебный план, занимаясь по которому не более  $T$  часов, начинающий программист (UP=1 по всем темам) достиг бы уровня подготовки  $L$  по всем темам, прорешав за это время как можно больше различных задач.

### Входные данные

В первой строке входного файла задано число  $T$  - количество часов, которыми располагает претендент для подготовки к собеседованию. Во второй строке записано число  $L$  ( $2 \leq L \leq 16$ ), а в третьей строке - количество задач  $M$  ( $1 \leq M \leq 500$ ,  $2 \leq T \leq M$ ). Каждая из последующих  $M$  строк содержит 8 чисел, описывающих одну из задач. Первые четыре числа определяют UP претендента по всем темам, необходимые для того, чтобы он мог решить эту задачу. Следующие четыре числа задают UP, до которых в результате решения этой задачи повысятся те уровни подготовки претендента, которые были ниже.

### Выходные данные

При возможности достижения уровня подготовки  $L$  по всем темам в выходной файл вывести сначала максимальное количество задач в учебном плане, а затем - последовательность номеров задач, которые должен решить претендент. Задачи нумеруются в том порядке, в каком они описаны во входном файле, ни одна задача не может быть указана более одного раза. Если с помощью указанного во входном файле набора задач за время  $T$  начинающий программист не сможет достигнуть UP, равного  $L$  по всем темам, вывести в выходной файл одно число 0.

### Пример

input.txt	output.txt
7	4
5	2 1 4 3
6	
2 1 1 1 2 4 5 5	
1 1 1 1 3 1 1 1	
3 3 3 3 3 3 3 3	
1 3 1 1 5 5 5 5	
2 2 2 2 2 2 2 2	
1 2 3 4 2 3 4 5	