

## ЛАБОРАТОРНАЯ №2

### Комбинаторика

Input file name:	input.txt
Output file name:	output.txt
Time limit (per test case):	1 sec
Memory limit (per test case):	64 MB
Score:	100 points

#### А. Суммарная полезность (тема: генерация всех подмножеств)

Дано  $n$  предметов, каждый из которых характеризуется весом  $w_i$  и полезностью  $p_i$ . Необходимо выбрать некоторый набор этих предметов так, чтобы суммарный вес этого набора не превышал  $W$ , а суммарная полезность была максимальна.

Ваша задача состоит в том, чтобы написать программу, решающую задачу о рюкзаке.

##### Формат входного файла

Первая строка входного файла содержит натуральные числа  $n$  ( $1 \leq n \leq 20$ ) и  $W$  ( $1 \leq W \leq 10^9$ ). Каждая из последующих  $n$  строк содержит описание одного предмета. Каждое описание состоит из двух чисел:  $w_i$  — веса предмета и  $p_i$  — его полезности ( $1 \leq w_i, p_i \leq 10^9$ ).

##### Формат выходного файла

В первой строке выходного файла выведите максимальную суммарную полезность.

##### Примеры

input.txt	output.txt
2 10 10 100 9 80	100
5 100 80 1000 50 550 50 550 50 550 50 550	1100
6 100 80 1000 50 550 50 550 50 550 50 550 100 1100	1100

#### В. Перестановки

Дана строка, состоящая из  $M$  попарно различных символов. Вывести все перестановки символов данной строки.

**Ограничения:**  $2 \leq M \leq 8$ , символы - буквы латинского алфавита и цифры.

**Ввод:** В первой строке файла находится исходная строка.

**Вывод:** В первой строке вывести количество перестановок, в следующих строках вывести в каждой строке по одной перестановке. Перестановки можно выводить в любом порядке. Повторений и строк, не являющихся перестановками исходной, быть не должно.

### Примеры

INPUT.TXT	OUTPUT.TXT
AB	2 AB BA
IOX	6 XOI OIX IXO XIO OXI IOX

**С. Пароль** (тема: размещение с повторениями). Сколько можно составить паролей с длиной N символов из заданного набора символов латинского алфавита и цифр 0 - 9, но с дополненными условиями:

- 1) символ C1 может использоваться в пароле любое количество раз, но обязательно должен быть на первом месте;
- 2) символ C2 должен быть использован в пароле ровно k раз.

#### Формат входных данных

В первой строке задан набор символов (не более 8 символов) из которых строится пароль, во второй длина пароля, в третьей строке задан символ C1, в четвёртой строке заданы C2 и k.

#### Формат выходных данных

##### Пример

input.txt	output.txt
PASSWORD1 6 P 1 3	490

## ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

### 1. Задача о восьми ферзях

Расположения восьми ферзей, которые не бьют друг – друга пронумерованы в лексикографическом порядке (симметричные решения считать решением). Найти k-е расположение.

**Ввод:** В первой строке файла находится целое число k.

**Вывод:** В первой строке вывести ответ.

#### Примеры

INPUT.TXT	OUTPUT.TXT
1	1 5 8 6 3 7 2 4

### Шахматная доска

8			+					
7						+		
6				+				
5		+						
4								+
3					+			
2							+	
1	+							
	1	2	3	4	5	6	7	8

**2. АВВА.** В алфавите племени Мумбо-Юмбо имеется только две буквы – **А** и **В**, а в каждом слове равное количество обеих букв. Например, АВВА – слово длины 4 из словаря племени.

Найдите для данного слова длины  $2k$  следующее за ним в лексикографическом порядке слово такой же длины или указывающий, что данное слово – последнее.

**Формат входных данных**

В первой строке задано слово.

**Формат выходных данных**

Напечатать следующее слово за заданным словом, а если заданное слово последнее, то напечатать 0.

**Пример**

input.txt	output.txt
АВАВААВВ	АВАВАВAB

**3. Жемчужное ожерелье.** Круглое ожерелье состоит из  $N$  жемчужин. Каждая жемчужина либо черного (Ч), либо белого (Б) цвета. Получите количество всевозможных вариантов ожерелий, которые можно составить из  $N$  жемчужин.

Ожерелья являются замкнутыми. Это означает, например, что два ожерелья, состоящие из четырех жемчужин: Б-Б-Б-Ч и Б-Ч-Б-Б являются одинаковыми.

**Формат входных данных**

Содержит единственное целое число  $N$  ( $N \leq 8$ ).

**Формат выходных данных**

Первая строка количество полученных ожерелий. В следующих строках описание ожерелий.

**Пример**

input.txt	output.txt
3	4

**Примечание.** В данном примере можно получить следующие ожерелья:

ЧЧЧ

ЧЧБ

ЧББ

БББ

**4. Подмножество.** На плоскости задано множество  $N$  ( $1 < N \leq 20$ ) материальных точек. Найти такое подмножество этого множества, содержащее ровно  $M$  ( $M < N$ ) точек, центр тяжести которого находится наиболее близко к началу координат.

**Формат входных данных**

В первой строке – числа  $N$  и  $M$ . В последующих  $N$  строках точки, для которых заданы координаты и масса.

**Формат выходных данных**

Выдать номера точек в порядке следования, попавших в искомое множество.

**Пример**

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

**5. Стоимость.** Даны  $N$  предметов различной стоимости, из них требуется отобрать несколько предметов так, чтобы суммарная стоимость отобранных вещей составляла бы ровно  $S$  рублей. Напечатать всевозможные варианты.

**Формат входных данных**

В первой строке заданы два числа  $N$  и  $S$ . Во второй строке заданы  $N$  натуральных чисел  $p_1, p_2, \dots, p_n$  - стоимости предметов.

**Формат выходных данных**

Напечатать все полученные варианты решения задачи. Если это невозможно – напечатать -1.

**Пример**

input.txt	output.txt
5 12 2 4 7 6 5	2 4 6 7 5
4 16 12 5 13 2	-1

**6. План работы цеха.** В цехе имеется  $N$  станков и столько же рабочих. Мастер знает эффективность работы каждого рабочего на каждом станке. Ему необходимо распределить рабочих по станкам с учетом следующих ограничений:

- 1) ни один из рабочих не должен быть назначен на работу на самом неэффективном для него станке (иначе он не получит премию);
- 2) не менее четверти общего числа рабочих должны работать на станках, на которых эффективность их труда максимальна (надо успеть выполнить задание).

Найдите распределение рабочих по станкам, удовлетворяющее указанным условиям.

**Формат входных данных**

В первой строке задано число  $N$ , в следующих  $N$  строках заданы по  $N$  чисел, где  $i$ -е число определяет эффективность работы рабочего на  $i$ -м станке.

**Формат выходных данных**

Список номеров рабочих, где  $i$ -е число соответствует номеру рабочего стоящего на  $i$ -м станке.

**Пример**

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

**7. Купюры.** Составить алгоритм подсчета количества способов, которыми можно разменять  $S$  рублей медными купюрами достоинством 1, 5, 10, 50, 100, 500 рублей.

**Формат входных данных**

Содержит единственное целое число  $S$ .

### **Формат выходных данных**

Первая строка количество способов размена. Если невозможно напечатать -1.

### **Пример**

input.txt	output.txt
20	9

**8. Патронташ.** Имеется  $n$  патронташей **12** и **16** калибра на  $a_1, a_2, \dots, a_n$  патронов. Известно, что среди них есть  $k$  патронташей **12** калибра, а патронов **12** калибра в два раза больше чем патронов **16** калибра. Найти номера патронташей **12** калибра.

### **Формат входных данных**

В первой строке заданы два натуральных числа  $n$  и  $k$ . Во второй строке заданы  $n$  натуральных чисел  $a_1, a_2, \dots, a_n$ ,  $a_i$  – количество патронов в  $i$ -ом патронташе.

### **Формат выходных данных**

Напечатать номера выбранных патронташей в порядке следования.

### **Пример**

input.txt	output.txt
3 2 4 6 8	1 3
4 3 5 8 4 7	1 3 4

**9. Центр тяжести.** На плоскости задано множество  $N$  ( $1 < N \leq 20$ ) материальных точек. Найти такое подмножество этого множества, содержащее ровно  $M$  ( $M < N$ ) точек, центр тяжести которого находится наиболее близко к началу координат.

### **Формат входных данных**

В первой строке – числа  $N$  и  $M$ . В последующих  $N$  строках точки, для которых заданы координаты и масса.

### **Формат выходных данных**

Выдать номера точек в порядке следования, попавших в искомое множество.

### **Пример**

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

## **10. Бутылки**

В цех вторичной переработки поступают бутылки  $N$  ( $1 \leq N \leq 8$ ) видов: А, В, С, ... (первые  $N$  заглавных букв латинского алфавита). Бутылки поступают на переработку партиями из  $N$  контейнеров, причем в каждом контейнере могут находиться бутылки различных видов. Перед вторичной переработкой бутылок специальные рабочие сортируют их по видам таким образом, чтобы после сортировки в каждом из поступивших контейнеров остались бутылки только одного вида. В каждом из контейнеров может помещаться неограниченное количество бутылок.

**Требуется** написать программу, которая определяет минимальное количество перемещений бутылок, обеспечивающих их сортировку по видам, причем за каждое перемещение можно переместить только одну бутылку из одного контейнера в другой.

### **Формат входных данных**

Входной файл состоит из N+1 строк. В первой строке записано число N. Во второй строке располагаются разделенные пробелами N целых числа, соответствующие количеству бутылок вида A, B, C, ... в первом контейнере. В последующих строках содержится аналогичная информация для второго, третьего, ..., N-го контейнеров соответственно. Известно, что количество бутылок в каждом из контейнеров не превосходит 32767.

**Формат выходных данных**

Выходной файл должен состоять из двух строк. В первой располагается последовательность из символов A, B, C, ..., которая определяет какого вида бутылки находятся после сортировки в 1-м, 2-м, ..., N-м контейнерах. Во второй строке располагается число, определяющее искомое количество перемещений бутылок.

Если возможно несколько вариантов ответа, то необходимо выдать любой из них.

**Пример**

INPUT.TXT	OUTPUT.TXT
3 12 6 20 23 9 45 21 31 13	ACB 92
4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	ABCD 102

**11. Слово.** Задается некоторое слово (например, LOOP). Из всех его букв составляются различные другие слова, возможно бессмысленные (POOL, OOLP и т.д.). По заданному слову из этого набора составить следующее за ним по алфавиту слово из этого набора.

**Формат входных данных**

Задано исходное слово.

**Формат выходных данных**

Следующее слово в лексикографическом порядке.

**Пример**

input.txt	output.txt
PAROLE	PEALOR

**12. Перестановка**

По номеру определить перестановку.

**Формат входных данных**

Входной файл содержит два целых числа n и k – количество элементов в перестановке и номер перестановки.

**Формат выходных данных**

Входной файл должен содержать соответствующую перестановку.

**Примеры**

INPUT.TXT	OUTPUT.TXT
8 37021	8 3 4 5 6 1 2 7

**13. Перестановка**

По перестановке определить его номер в лексикографическом перечислении всех перестановок множества  $\{1, 2, \dots, n\}$ .

**Формат входных данных**

Входной файл содержит в первой строке одно число n – количество элементов в перестановке, во второй – перестановку.

**Формат выходных данных**

Входной файл должен содержать номер заданной перестановки в лексикографическом перечислении всех перестановок.

**Примеры**

INPUT.TXT	OUTPUT.TXT
8	37022
8 3 4 5 6 1 7 2	

**14. Число.** Написать программу генерации всех двоичных чисел с ненулевой первой цифрой, состоящей из **n** единиц и **m** нулей.

**Формат входных данных**

В первой строке заданы два натуральных числа **n** и **m**.

**Формат выходных данных**

Полученные двоичные числа в лексикографическом порядке.

**Пример**

input.txt	output.txt
3 2	10011 10101 10110 11001 11010 11100

**15. Размен.** Составить алгоритм получения способа размена **S** рублей медными купюрами достоинством 1, 5, 10, 50, 100, 500 рублей.

**Формат входных данных**

Содержит единственное целое число **S**.

**Формат выходных данных**

Первая строка количество способов размена. Если невозможно напечатать **-1**.

**Пример**

input.txt	output.txt
20	9

**16. Выражение.** Даны **N** целых чисел  $X_1, X_2, \dots, X_N$ . Расставить между ними знаки "+" и "-" так, чтобы значение получившегося выражения было равно заданному целому **S**.

**Ограничения:**  $2 \leq N \leq 24$ ,  $0 \leq X_i \leq 50\,000\,000$ ,

$1\,000\,000\,000 \leq S \leq 1\,000\,000\,000$ , время 3 с.

**Формат входных данных**

В первой строке находятся числа **N** и **S**. В следующей строке - **N** чисел через пробел.

**Формат выходных данных**

Если получить требуемый результат невозможно, вывести "No solution", если можно, то вывести равенство. Если решение не единственное, вывести любое.

**Примеры**

input.txt	output.txt
3 10 15 25 30	15+25-30=10
2 100 10 10	No solution

**17. Последовательность.** Получить последовательность  $a_1, a_2, \dots, a_n$  цифр 0, 1, 2, в которой нет смежных одинаковых членов (например, последовательность 2,0,1,1,... не годится, так как рядом расположены два одинаковых члена 1).

**Формат входных данных**

В первой строке задано одно натуральное число **n**.

**Формат выходных данных**

Напечатать в лексикографическом порядке все последовательности удовлетворяющие вышеуказанному условию.

**Пример**

input.txt	output.txt
3	010 012 020 021 101 102 120 121 201 202 210 212

**18. Центр тяжести.** На плоскости задано множество  $N$  ( $1 < N \leq 20$ ) материальных точек. Найти такое подмножество этого множества, содержащее ровно  $M$  ( $M < N$ ) точек, центр тяжести которого находится наиболее близко к началу координат.

**Формат входных данных**

В первой строке – числа  $N$  и  $M$ . В последующих  $N$  строках точки, для которых заданы координаты и масса.

**Формат выходных данных**

Выдать номера точек в порядке следования, попавших в искомое множество.

**Пример**

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

**19. Раскопки**

Во время недавних раскопок на Марсе были обнаружены листы бумаги с таинственными символами на них. После долгих исследований учёные пришли к выводу, что надписи на них на самом деле могли быть обычными числовыми равенствами. Если бы этот вывод оказался верным, это доказало бы не только то, что на Марсе много лет назад были разумные существа, но и то, что они уже умели считать...

Ученые смогли понять, что в этом случае означают найденные символы, и перевели эти равенства на обычный язык – язык цифр, скобок, знаков арифметических действий и равенства. Кроме того, из других источников было получено веское доказательство того, что марсиане знали только три операции – сложение, умножение и вычитание (марсиане никогда не использовали «унарный минус»: вместо «-5» они писали «0-5»). Также ученые доказали, что марсиане не наделяли операции разным приоритетом, а просто вычисляли выражения (если в них не было скобок) слева направо: например,  $3+3*5$  у них равнялось 30, а не 18.

К сожалению, символы арифметических действий марсиане почему-то наносили специальными чернилами, которые, как оказалось, были не очень стойкими, и поэтому в



найденных листках между числами вместо знаков действий были пробелы. Если вся вышеизложенная теория верна, то вместо этих пробелов можно поставить знаки сложения, вычитания и умножения так, чтобы равенства стали верными. Например, если был найден лист бумаги с надписью «18=7 (5 3) 2», то возможна такая расстановка знаков: «18=7+(5-3)\*2» (помните про то, в каком порядке марсиане вычисляют выражения!). В то же время, если попался лист с надписью «5=3 3», то марсиане явно не имели в виду числового равенства, когда писали это...

Вы должны написать программу, находящую требуемую расстановку знаков или сообщаящую, что таковой не существует.

### **Формат входных данных**

Первая строка входного файла состоит из натурального (целого положительного) числа, не превосходящего  $2^{30}$ , знака равенства, и последовательности натуральных чисел (не более десяти), произведение которых также не превосходит  $2^{30}$ . Некоторые группы чисел (одно или более) могут быть окружены скобками. Длина входной строки не будет превосходить 80 символов, и других ограничений на количество и вложенность скобок нет. Между двумя соседними числами, не разделенными скобками, всегда будет хотя бы один пробел, во всех остальных местах может быть любое (в том числе и 0) число пробелов (естественно, внутри числа пробелов нет).

### **Формат выходных данных**

В выходной файл необходимо вывести одну строку, содержащую полученное равенство (т.е., исходное равенство со вставленными знаками арифметических действий). В случае если требуемая расстановка знаков невозможна, вывести строку, состоящую из единственного числа «-1». Выходная строка не должна содержать пробелов.

### **Примеры**

<b>h.in</b>	<b>h.out</b>
18=7 (5 3) 2	18=7+(5-3)*2
5= 3 3	-1

**20. Ожерелье.** Круглое ожерелье состоит из  $N$  жемчужин. Каждая жемчужина либо черного (Ч), либо белого (Б) цвета. Получите количество всевозможных вариантов ожерелий, которые можно составить из  $N$  жемчужин. Ожерелья являются замкнутыми. Это означает, например, что два ожерелья, состоящие из четырех жемчужин: Б-Б-Б-Ч и Б-Ч-Б-Б являются одинаковыми.

### **Формат входных данных**

Содержит единственное целое число  $N$  ( $N \leq 8$ ).

### **Формат выходных данных**

В первой строке напечатать количество полученных ожерелий.

### **Пример**

<b>input.txt</b>	<b>output.txt</b>
3	4

Примечание. В данном примере можно получить следующие ожерелья:

ЧЧЧ, ЧЧБ, ЧББ, БББ

**21. Фестиваль.** В клубе любителей кино состоит  $N$  ( $N \leq 20$ ) членов. Ежегодно на Международный фестиваль в Канн посылается  $k$  ( $k \leq N$ ) членов киноклуба. Одна делегация не должна ездить дважды, т.е. каждый год состав делегации должен хотя бы на одного человека отличаться от состава любой предыдущей делегации. Напечатайте списки всех возможных делегаций.

### **Формат входных данных**

В первой строке заданы количество членов клуба  $N$  и количество человек в делегации  $k$ . В следующих строках фамилии членов клуба в лексико-графическом порядке.

**Формат выходных данных**

В первой строке напечатайте количество всевозможных делегаций, а со второй строки – делегации в лексикографическом порядке. Фамилии членов делегации печатать через пробел.

**Пример**

input.txt	output.txt
5 3 Васильев Иванов Петров Тарасов Харитонов	10 Васильев Иванов Петров Васильев Иванов Тарасов Васильев Иванов Харитонов Васильев Петров Тарасов Васильев Петров Харитонов Васильев Тарасов Харитонов Иванов Петров Тарасов Иванов Петров Харитонов Иванов Тарасов Харитонов Петров Тарасов Харитонов

**22. Центр тяжести.** На плоскости задано множество  $N$  ( $1 < N \leq 20$ ) материальных точек. Найти такое подмножество этого множества, содержащее ровно  $M$  ( $M < N$ ) точек, центр тяжести которого находится наиболее близко к началу координат.

**Формат входных данных**

В первой строке – числа  $N$  и  $M$ . В последующих  $N$  строках точки, для которых заданы координаты и масса.

**Формат выходных данных**

Выдать номера точек в порядке следования, попавших в искомое множество.

**Пример**

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

**23. Перестановка**

По перестановке определить его номер в лексикографическом перечислении всех перестановок множества  $\{1, 2, \dots, n\}$ .

**Формат входных данных**

Входной файл содержит в первой строке одно число  $n$  – количество элементов в перестановке, во второй – перестановку.

**Формат выходных данных**

Входной файл должен содержать номер заданной перестановки в лексикографическом перечислении всех перестановок.

**Примеры**

INPUT.TXT	OUTPUT.TXT
8 8 3 4 5 6 1 7 2	37022

**24. Число.** Написать программу генерации всех двоичных чисел с ненулевой первой цифрой, состоящей из  $n$  единиц и  $m$  нулей.

**Формат входных данных**

В первой строке заданы два натуральных числа  $n$  и  $m$ .

**Формат выходных данных**

Полученные двоичные числа в лексикографическом порядке.

**Пример**

input.txt	output.txt
3 2	10011 10101 10110 11001 11010 11100

**25. Размен.** Составить алгоритм получения способа размена  $S$  рублей медными купюрами достоинством 1, 5, 10, 50, 100, 500 рублей.

**Формат входных данных**

Содержит единственное целое число  $S$ .

**Формат выходных данных**

Первая строка количество способов размена. Если невозможно напечатать **-1**.

**Пример**

input.txt	output.txt
20	9

**26. Выражение.** Даны  $N$  целых чисел  $X_1, X_2, \dots, X_N$ . Расставить между ними знаки "+" и "-" так, чтобы значение получившегося выражения было равно заданному целому  $S$ .

**Ограничения:**  $2 \leq N \leq 24$ ,  $0 \leq X_i \leq 50\,000\,000$ ,

$1\,000\,000\,000 \leq S \leq 1\,000\,000\,000$ , время 3 с.

**Формат входных данных**

В первой строке находятся числа  $N$  и  $S$ . В следующей строке -  $N$  чисел через пробел.

**Формат выходных данных**

Если получить требуемый результат невозможно, вывести "No solution", если можно, то вывести равенство. Если решение не единственное, вывести любое.

**Примеры**

input.txt	output.txt
3 10 15 25 30	15+25-30=10
2 100 10 10	No solution

**27. Последовательность.** Получить последовательность  $a_1, a_2, \dots, a_n$  цифр 0, 1, 2, в которой нет смежных одинаковых членов (например, последовательность 2,0,1,1,... не годится, так как рядом расположены два одинаковых члена 1).

**Формат входных данных**

В первой строке задано одно натуральное число  $n$ .

**Формат выходных данных**

Напечатать в лексикографическом порядке все последовательности удовлетворяющие вышеуказанному условию.

**Пример**

input.txt	output.txt
3	012 021 102 120 201

**28. Центр тяжести.** На плоскости задано множество  $N$  ( $1 < N \leq 20$ ) материальных точек. Найти такое подмножество этого множества, содержащее ровно  $M$  ( $M < N$ ) точек, центр тяжести которого находится наиболее близко к началу координат.

**Формат входных данных**

В первой строке – числа  $N$  и  $M$ . В последующих  $N$  строках точки, для которых заданы координаты и масса.

**Формат выходных данных**

Выдать номера точек в порядке следования, попавших в искомое множество.

**Пример**

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

**29. Ожерелье.** Круглое ожерелье состоит из  $N$  жемчужин. Каждая жемчужина либо черного (Ч), либо белого (Б) цвета. Получите количество всевозможных вариантов ожерелий, которые можно составить из  $N$  жемчужин.

Ожерелья являются замкнутыми. Это означает, например, что два ожерелья, состоящие из четырех жемчужин: Б-Б-Б-Ч и Б-Ч-Б-Б являются одинаковыми.

**Формат входных данных**

Содержит единственное целое число  $N$  ( $N \leq 8$ ).

**Формат выходных данных**

В первой строке напечатать количество полученных ожерелий.

**Пример**

input.txt	output.txt
3	4

Примечание. В данном примере можно получить следующие ожерелья:

ЧЧЧ, ЧЧБ, ЧББ, БББ

**30. Морской бой.** Самолет находится на палубе неподвижного авианосца и готов в любой момент к взлету. Задача игрока в этой миссии состоит в уничтожении  $N$  кораблей противника. После уничтожения всех кораблей самолет должен вернуться обратно на авианосец.

Для простоты будем считать плоской поверхность моря, где располагается авианосец. Введем прямоугольную декартову систему координат и разместим авианосец в начале координат. Каждый из кораблей в начальный момент игры находится в некоторой точке  $(x, y)$ , и сразу после начала игры движется равномерно и прямолинейно так, что его вектор скорости равен  $(V_x, V_y)$ .

Конструктивные особенности самолета таковы, что он может двигаться с любой скоростью, не превосходящей  $U$ . Для того, чтобы сбросить бомбу, которая была специально придумана для этой игры, самолету необходимо находиться непосредственно над кораблем. Корабли считаются точками, т.е. размером кораблей можно пренебречь. Считается также, что самолет может мгновенно взлететь с палубы авианосца, и время падения бомбы на цель равно нулю.

Требуется написать программу, определяющую минимальное время, за которое игрок сможет уничтожить все корабли и возвратить самолет обратно на авианосец.

**Формат входных данных**

Первая строка входного файла содержит число  $N$ , определяющее количество кораблей ( $1 \leq N \leq 9$ ). Вторая строка входного файла содержит целое число  $U$  ( $1 \leq U \leq 10000$ ), задающее скорость самолета в метрах в секунду. Последующие  $N$  строк описывают все корабли. Каждая строка содержит четыре целых числа  $x$ ,  $y$ ,  $V_x$ ,  $V_y$ , не превосходящих 10000 по модулю и определяющих начальные координаты и скорость корабля, соответственно. Координаты кораблей заданы в метрах, скорости – в метрах в секунду.

Гарантируется, что самолет летит быстрее, чем плавает любой из кораблей.

**Формат выходных данных**

В первой строке выходного файла выведите минимальное время, требуемое на выполнение миссии. Требуемая точность – не менее  $10^{-3}$ .

**Пример**

input.txt	output.txt
1 1000 10 10 0 0	0.0283