

## Содержание

<b>Must have</b>	<b>2</b>
Задача 10А. Почтовые отделения [0.1 sec, 256 mb]	2
Задача 10В. Гроб [0.1 sec, 256 mb]	3
Задача 10С. Свертка [0.5 sec, 256 mb]	4
<b>Обязательные задачи</b>	<b>5</b>
Задача 10D. Свертка [1 sec, 256 mb]	5
Задача 10E. Джентльмены [0.1 sec, 256 mb]	6
Задача 10F. Ship [3 sec, 256 mb]	7
Задача 10G. Наибольшая общая возрастающая [0.5 sec, 256 mb]	8
Задача 10H. Мостостроение [0.8 sec, 256 mb]	9
<b>Дополнительные задачи</b>	<b>10</b>
Задача 10I. Мостостроение [0.4 sec, 256 mb]	10
Задача 10J. Общая подпоследовательность [1.2 sec, 8 mb]	11
Задача 10K. Сервера [2.5 sec, 256 mb]	12
Задача 10L. Casino. Казино [0.4 sec, 256 mb]	13

---

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на GNU C++ компиляторы с суффиксом inc, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

## Must have

### Задача 10А. Почтовые отделения [0.1 sec, 256 mb]

Вдоль прямой дороги расположены деревни. Дорога представляется целочисленной осью, а расположение каждой деревни задается одним целым числом — координатой на этой оси. Никакие две деревни не имеют одинаковых координат. Расстояние между двумя деревнями вычисляется как модуль разности их координат.

В некоторых, не обязательно во всех, деревнях будут построены почтовые отделения. Деревня и расположенное в ней почтовое отделение имеют одинаковые координаты. Почтовые отделения необходимо расположить в деревнях таким образом, чтобы общая сумма расстояний от каждой деревни до ближайшего к ней почтового отделения была минимальной.

#### Формат входных данных

В первой строке содержатся два целых числа: количество деревень  $n$  ( $1 \leq n \leq 300$ ) и количество почтовых отделений  $m$  ( $1 \leq m \leq 30$ ),  $m \leq n$ . Вторая строка содержит  $n$  целых чисел в возрастающем порядке, являющихся координатами деревень. Для каждой координаты  $x$  верно  $1 \leq x \leq 10^4$ .

#### Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — общую сумму расстояний от каждой деревни до её ближайшего почтового отделения. Вторая строка должна содержать  $m$  целых чисел в возрастающем порядке. Эти числа являются искомыми координатами почтовых отделений. Если для заданного расположения деревень есть несколько решений, необходимо найти любое из них.

#### Пример

stdin	stdout
10 5	9
1 2 3 6 7 9 11 22 44 50	2 7 22 44 50

#### Замечание

В тему лекции. Самая сложная из обязательных задач.  
Заметьте, даже  $n^3$  заходит, главное — корректность.

**Задача 10В. Гроб [0.1 сек, 256 mb]**

*Генри Гроб* — швейцарский шахматист, международный мастер. Шахматный литератор. Художник.

Википедия

Генри Гроб решил сыграть в новую игру, отдалённо напоминающую шахматы. Изначально фишка стоит в корне подвешенного дерева, затем игроки по очереди передвигают её. За ход необходимо переставить фишку в одного из детей текущей вершины. Проигрывает тот, кто не может сделать ход.

Генри начинает первым. Определите, сможет ли он выиграть, если оба игрока играют оптимально.

**Формат входных данных**

Все числа в файле целые.

$1 \leq N \leq 10^5$  — количество вершин дерева.

Затем  $N$  чисел  $0 \leq p_i \leq N$  —  $i$ -ая вершина имеет родителя с номером  $p_i$ . Для корня дерева  $p_i = 0$ , для остальных вершин  $p_i \neq 0$ .

**Формат выходных данных**

«YES», если Гроб выигрывает, «NO» — в противном случае.

**Пример**

stdin	stdout
1 0	NO

**Замечание**

Чтобы хранить дерево, используйте массив векторов `vector<int> children[n];`

### Задача 10С. Свертка [0.5 сек, 256 mb]

Петя хочет сократить запись последовательности, состоящей из заглавных латинских букв. Для этого он может свернуть ее повторяющиеся подпоследовательности. Например, последовательность АААААААААВАВАВССD может быть записана как 10(A)2(BA)B2(C)D.

Формальной определение свернутой последовательности и соответствующей ей операции развертки дается следующим образом:

- Последовательность, которая содержит единственный символ от 'A' до 'Z' представляет из себя свернутую последовательность. При развертке такой последовательности получается она сама.
- Если  $S$  и  $Q$  — свернутые последовательности, то  $SQ$  также свернутая последовательность. Если при развертке строки  $S$  получается строка  $S'$ , а при развертке  $Q$  получается  $Q'$ , то при развертке  $SQ$  получается строка  $S'Q'$ .
- Если  $S$  — свернутая последовательность, то  $X(S)$  также свернутая последовательность, где  $X$  это десятичное представление целого числа большего единицы. Если при развертке строки  $S$  получается строка  $S'$ , то при развертке  $X(S)$  получается строка  $S'$ , повторенная  $X$  раз.

Петя хочет свернуть заданную последовательность таким образом, чтобы результат содержал наименьшее число символов.

#### Формат входных данных

Входной файл содержит непустую строку, состоящую из заглавных латинских букв. Длина строки не превышает 100 символов.

#### Формат выходных данных

В выходной файл выведите одну строку, содержащую наименьшую последовательность развертка которой даст строку, заданную во входном файле.

Если ответов несколько — выведите любой из них.

#### Пример

stdin	stdout
AAAAAAAAAABABABCCD	9(A)3(AB)CCD
NEERCYESYESYESNEERCYESYESYES	2(NEERC3(YES))

#### Замечание

Разобрано на паре.

## Обязательные задачи

### Задача 10D. Свертка [1 sec, 256 mb]

Петя хочет сократить запись последовательности, состоящей из заглавных латинских букв. Для этого он может свернуть ее повторяющиеся подпоследовательности. Например, последовательность АААААААААВАВАВССD может быть записана как 10(A)2(BA)B2(C)D.

Формальной определением свернутой последовательности и соответствующей ей операции развертки дается следующим образом:

- Последовательность, которая содержит единственный символ от 'A' до 'Z' представляет из себя свернутую последовательность. При развертке такой последовательности получается она сама.
- Если  $S$  и  $Q$  — свернутые последовательности, то  $SQ$  также свернутая последовательность. Если при развертке строки  $S$  получается строка  $S'$ , а при развертке  $Q$  получается  $Q'$ , то при развертке  $SQ$  получается строка  $S'Q'$ .
- Если  $S$  — свернутая последовательность, то  $X(S)$  также свернутая последовательность, где  $X$  это десятичное представление целого числа большего единицы. Если при развертке строки  $S$  получается строка  $S'$ , то при развертке  $X(S)$  получается строка  $S'$ , повторенная  $X$  раз.

Петя хочет свернуть заданную последовательность таким образом, чтобы результат содержал наименьшее число символов.

#### Формат входных данных

Входной файл содержит непустую строку, состоящую из заглавных латинских букв. Длина строки не превышает 1000 символов.

#### Формат выходных данных

В выходной файл выведите одну строку, содержащую наименьшую последовательность развертка которой даст строку, заданную во входном файле.

Если ответов несколько — выведите любой из них.

#### Пример

stdin	stdout
AAAAAAAAAABABABCCD	9(A)3(AB)CCD
NEERCYESYESYESNEERCYESYESYES	2(NEERC3(YES))

#### Замечание

Разобрано на паре.

Существует кубическое решение быстрее, чем за 0.2 секунды.

Если вы вдруг получили ОК решением за  $O(n^3)$ , обязательно поделитесь идеей.

### Задача 10Е. Джентльмены [0.1 сек, 256 mb]

Вспомним старый анекдот:

*Говорит как-то один джентльмен другому:*

- *А не перекинуться ли нам в картишки?*
- *Знаете, я не играл уже лет десять...*
- *А я – лет пятнадцать...*

*Так, слово за слово, решили они вспомнить молодость. Первый джентльмен попросил лакея принести колоду карт, и, перед тем, как начать раздачу, взвесил её в руке.*

– *Мне кажется, здесь не хватает одной карты... – начал он и передал колоду другому джентльмену.*

– *Да, девятки пик, – подтвердил тот.*

Дана неполная колода карт. Ваша программа должна определить, какие карты в колоде отсутствуют.

#### Формат входных данных

В первой строке дано целое положительное число, задающее вес данной неполной колоды в миллиграммах. Во второй строке – целое число  $N$ ,  $2 \leq N \leq 100$  – количество карт в полной колоде. Далее следует  $N$  строк, в каждой из которых записано целое число от 1 до 1000 – вес очередной карты в миллиграммах. Гарантируется, что суммарный вес всех карт в полной колоде превосходит вес неполной колоды.

#### Формат выходных данных

Если задача не имеет решения, выведите единственное число 0. Если решение существует, но не единственно, то выведите  $-1$ . Наконец, если можно однозначно восстановить, каких карт недостаёт в неполной колоде по сравнению с полной, то следует через пробел вывести номера отсутствующих карт в порядке возрастания.

#### Примеры

stdin	stdout
270 4 100 110 170 200	2 4
270 4 100 110 160 170	-1
270 4 100 120 160 180	0

### Задача 10F. Ship [3 sec, 256 mb]

A ship has been assigned to transport cargo across the river Daugava (Western Dvina) that has the capacity to carry at most  $t$  tons of cargo. There is a hangar on the right bank of the river that contains  $n$  cargo containers to be transported by the ship. The weight of each container is an integer number of tons and it does not exceed  $t$  tons. The containers are arranged in a single line from the start to the end of the hangar. A container can be carried onto the ship by a liftloader that can pick only a container either at the start or at the end of the line. The total weight of the containers placed on a ship cannot exceed the capacity of the ship. None of the containers are allowed to be opened, thus it is impossible to redistribute the cargo among the containers. It is also expensive to move the containers so once the container has been picked up by the liftloader, it is then directly carried onto the ship. Only then the liftloader can pick another container from the hangar. Once the ship is loaded, it can transport the containers to the left bank and return to the right bank empty.

Write a program that calculates the minimum possible number of trips the ship has to make to transport all of the containers to the left river bank!

#### Формат входных данных

The first line of input contains two integers  $t$  and  $n$  ( $1 \leq t \leq 10^9$ ,  $1 \leq n \leq 10^4$ ), the capacity of the ship and the number of containers in the hangar, accordingly. The next line contains  $n$  integers, the weight of the containers in tons. The  $i$ -th number  $w_i$  ( $1 \leq w_i \leq t$ ) denotes the weight of the  $i$ -th container from the start of the hangar. On each line the numbers are separated by spaces.

#### Формат выходных данных

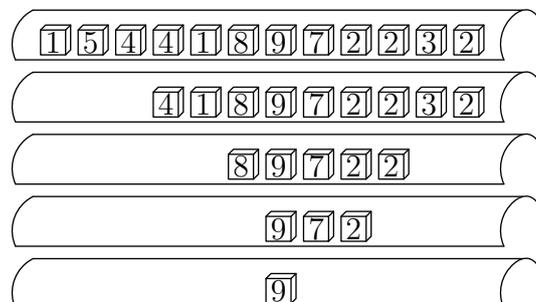
Output a single integer: the minimum possible trip number for the ship to transport all of the containers to the other river bank.

#### Примеры

stdin	stdout
10 12 1 5 4 4 1 8 9 7 2 2 3 2	5

#### Замечание

The following figure shows the remaining containers in the hangar before and after each trip in the given example.



#### Замечание

Задача с лекции. Вы знаете её условие, поэтому английский вам не помеха.

**Задача 10G. Наибольшая общая возрастающая [0.5 sec, 256 mb]**

Даны две последовательности чисел —  $a$  и  $b$ . Нужно найти наибольшую общую возрастающую подпоследовательность. Более формально: такие  $1 \leq i_1 < i_2 < \dots < i_k \leq a.n$  и  $1 \leq j_1 < j_2 < \dots < j_k \leq b.n$ , что  $\forall t : a_{i_t} = b_{j_t}, a_{i_t} < a_{i_{t+1}}$  и  $k$  максимально.

**Формат входных данных**

На первой строке целые числа  $n$  и  $m$  от 1 до 3 000 — длины последовательностей. Вторая строка содержит  $n$  целых чисел, задающих первую последовательность. Третья строка содержит  $m$  целых чисел, задающих вторую последовательность. Все элементы последовательностей — целые неотрицательные числа, не превосходящие  $10^9$ .

**Формат выходных данных**

Выведите одно целое число — длину наибольшей общей возрастающей подпоследовательности.

**Пример**

stdin	stdout
6 5 1 2 1 2 1 3 2 1 3 2 1	2

**Замечание**

Разобрана на паре.

### Задача 10Н. Мостостроение [0.8 sec, 256 mb]

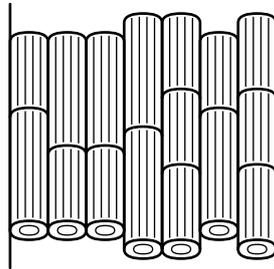
В деревне Гадюкино регулярно идут проливные дожди, в результате чего речка Вонючка, которую обычно можно просто перешагнуть, выходит из берегов. Чтобы можно было перейти разлившуюся реку, планируется построить плавучий мост из бревен, оставшихся от строительства бани бизнесмена, поселившегося неподалеку.

Все оставшиеся бревна имеют одинаковую толщину. При этом есть  $x$  бревен длины  $a$  и  $y$  бревен длины  $b$ .

Построенный мост должен состоять из  $l$  рядов, каждый из которых составлен из одного или нескольких бревен. Пилить бревна нельзя, так как последняя пила утонула при разливе Вонючки.

Главный инженер хочет построить мост максимальной возможной ширины, при этом ширина моста определяется по минимальной ширине ряда бревен.

Например, если нужно построить мост из семи рядов, и при этом есть шесть бревен длины 3 и десять бревен длины 2, то можно построить мост ширины 5.



#### Формат входных данных

Входной файл содержит пять натуральных чисел:  $x$ ,  $a$ ,  $y$ ,  $b$  и  $l$ . Все числа не превышают 150. Общее количество бревен не меньше  $l$ .

#### Формат выходных данных

Выведите в выходной файл одно число — максимальную возможную ширину моста.

#### Примеры

stdin	stdout
6 3 10 2 7	5
10 7 20 9 25	9

## Дополнительные задачи

### Задача 10I. Мостостроение [0.4 sec, 256 mb]

Давным давно, в 2009-м году...

---

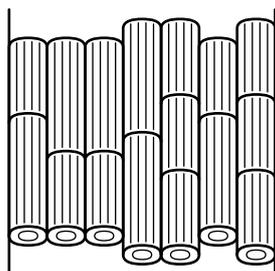
В деревне Зайкино регулярно идут проливные дожди, в результате чего речка Дубровка, которую обычно можно просто перешагнуть, выходит из берегов. Чтобы можно было перейти разлившуюся реку, планируется построить плавучий мост из брёвен, оставшихся от строительства бани бизнесмена, поселившегося неподалёку.

Все оставшиеся брёвна имеют одинаковую толщину. При этом есть  $x$  брёвен длины  $a$  и  $y$  брёвен длины  $b$ .

Построенный мост должен состоять из  $l$  рядов, каждый из которых составлен из одного или нескольких брёвен. Пилить брёвна нельзя, так как последняя пила утонула при разливе Дубровки.

Главный инженер хочет построить мост максимальной возможной ширины. Ширина моста определяется по минимальной ширине ряда брёвен в нём.

Например, если нужно построить мост из семи рядов, и при этом есть шесть брёвен длины 3 и десять брёвен длины 2, то можно построить мост ширины 5.



#### Формат входных данных

Ввод состоит из одного или нескольких тестовых случаев. Каждый тестовый случай состоит из пяти целых положительных чисел  $x$ ,  $a$ ,  $y$ ,  $b$  и  $l$ . Каждое число не превосходит 500. Общее количество брёвен в каждом тестовом случае не меньше  $l$ .

Обозначим  $d = \max(x, a, y, b, l)$ . Гарантируется, что сумма  $d$  по всем тестам не превосходит 5000.

#### Формат выходных данных

Для каждого тестового случая на отдельной строке выведите одно число — максимальную возможную ширину моста.

#### Пример

stdin	stdout
6 3 10 2 7	5
10 7 20 9 25	9
106 126 135 28 137	112

**Задача 10J. Общая подпоследовательность [1.2 сек, 8 mb]**

Даны две строки, состоящих из маленьких латинских букв. Нужно найти их наибольшую общую подпоследовательность.

**Формат входных данных**

На первой строке первая строка.

На второй строке вторая строка.

Длины строк от 1 до 5000.

**Формат выходных данных**

Максимальную по длине общую подпоследовательность на отдельной строке. Если ответов несколько, выведите любой. Если ответ пуст, перевод строки выводить все равно нужно.

**Пример**

stdin	stdout
abacabadabacaba	bccb
dbdccbdb	

**Замечание**

Неужели, всё-таки придётся писать линию памяти?

### Задача 10К. Сервера [2.5 sec, 256 mb]

Компьютерная сеть в некотором доме строилась по принципу присоединения нового компьютера к последнему из уже подключенных. Никакие два компьютера, будучи подключенными в сеть, между собой дополнительно никак не связывались. Таким образом, в сеть были объединены последовательно  $N$  компьютеров. Соседи обменивались информацией между собой, но в какой-то момент поняли, что им нужны прокси-серверы. Компьютерное сообщество дома решило установить прокси-серверы ровно на  $K$  компьютеров. Осталось только решить, какие именно компьютеры выбрать для этой цели. Главным критерием является ежемесячная стоимость обслуживания серверами всех компьютеров.

Для каждого компьютера установлен тариф его обслуживания, выраженный в рублях за метр провода. Стоимость обслуживания одного компьютера каким-то сервером равна тарифу компьютера, умноженному на суммарную длину провода от этого компьютера до сервера, которым он обслуживается.

Ваша задача написать программу, которая выберет такие компьютеры, чтобы установить на них прокси-серверы, что общие затраты на обслуживание всех компьютеров были бы минимальными

#### Формат входных данных

В первой строке входного файла записано два целых числа  $N$  и  $K$  — количество компьютеров в сети и количество прокси-серверов, которые нужно установить ( $1 \leq K \leq N \leq 2000$ ).

Все компьютеры в сети пронумерованы числами от 1 до  $N$  по порядку подключения.

Во второй строке записано одно целое число  $T_1$  — тариф обслуживания первого компьютера.

В следующих  $N - 1$  строках записано через пробел по два целых неотрицательных числа  $L_i, T_i$  — информация об остальных компьютерах в сети по порядку номеров.  $L_i$  — длина провода, соединяющего  $i$  — компьютер с соседним с меньшим номером,  $T_i$  — тариф обслуживания данного компьютера ( $2 \leq i \leq N$ ). Все  $L_i$  и  $T_i$  от 0 до  $10^6$ .

#### Формат выходных данных

В первую строку выходного файла необходимо вывести одно целое число — минимальную стоимость обслуживания всех компьютеров всеми серверами. Во второй строке должны быть записаны через пробел  $K$  номеров компьютеров, на которые необходимо установить серверы. При существовании нескольких вариантов размещения разрешается вывести любой.

#### Пример

stdin	stdout
3 1 10 2 2 3 3	19 1
3 2 10 2 2 3 3	4 1 3

#### Замечание

Вы же были на лекции? Значит, всё просто ; -)

### Задача 10L. Casino. Казино [0.4 sec, 256 mb]

Вновь открытое казино предложило оригинальную игру.

В начале игры крупье выставляет в ряд несколько фишек разных цветов. Кроме того, он объявляет, какие последовательности фишек игрок может забирать себе в процессе игры. Далее игрок забирает себе одну из заранее объявленных последовательностей фишек, расположенных подряд. После этого крупье сдвигает оставшиеся фишки, убирая разрыв. Затем игрок снова забирает себе одну из объявленных последовательностей и так далее. Игра продолжается до тех пор, пока игрок может забирать фишки.

Рассмотрим пример. Пусть на столе выставлен ряд фишек 'rrrgggbbb', и крупье объявил последовательности 'rg' и 'gb'. Игрок, например, может забрать фишки 'rg', лежащие на третьем и четвёртом местах слева. После этого крупье сдвинет фишки, и на столе получится ряд 'rrggbbb'. Ещё дважды забрав фишки 'rg', игрок добьётся того, что на столе останутся фишки 'bbb' и игра закончится, так как игроку больше нечего забрать со стола. Игрок мог бы действовать и по-другому — на втором и третьем ходах забрать не последовательности 'rg', а последовательности 'gb'. Тогда на столе остались бы фишки 'rrb'. Аналогично, игрок мог бы добиться того, чтобы в конце остались ряды 'rrr' или 'rbb'.

После окончания игры полученные фишки игрок меняет на деньги. Цена фишки зависит от её цвета.

Требуется написать программу, определяющую максимальную сумму, которую сможет получить игрок.

#### Формат входных данных

В первой строке входного файла записано число  $K$  ( $1 \leq K \leq 26$ ) — количество цветов фишек. Каждая из следующих  $K$  строк начинается со строчной латинской буквы, обозначающей цвет. Далее в той же строке через пробел следует целое число  $X_i$  ( $1 \leq X_i \leq 150$ ,  $i = 1..K$ ) — цена фишки соответствующего цвета.

В  $(K + 2)$ -ой строке описан ряд фишек, лежащих на столе в начале игры. Ряд задается  $L$  строчными латинскими буквами ( $1 \leq L \leq 150$ ), которые обозначают цвета фишек ряда.

В следующей строке содержится число  $N$  ( $1 \leq N \leq 150$ ) — количество последовательностей, которые были объявлены крупье. В следующих  $N$  строках записаны эти последовательности. Гарантируется, что сумма длин этих  $N$  строк не превосходит 150 символов, и все они непустые.

#### Формат выходных данных

В выходной файл выведите единственное целое число — максимальную сумму денег, которую может получить игрок.

**Пример**

stdin	stdout
6	16
a 1	
b 4	
d 2	
x 3	
f 1	
e 3	
fxeeabadd	
2	
aba	
ed	

**Замечание**

Привет с РОИ 2005. Казалось бы обычная динамика по подотрезкам...  
Не стесняйтесь при TL оптимизировать память, добавлять `bitset`.