

## Задача A. Fairland

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	512 мегабайт

В стране Справидляндии очень строгие законы, регулирующие то, как компании должны организовывать и оплачивать работу своих сотрудников:

- Каждая компания должна иметь ровно одного CEO, у которого нет менеджера.
- Каждый сотрудник, кроме CEO, должен иметь ровно одного менеджера. (Это значит, что структура компании представляет собой подвешенное дерево.)
- Пока сотрудник работает в компании, его менеджер никогда не меняется. Поэтому, если какой-то сотрудник покидает компанию, ее также вынуждены покинуть все сотрудники, для которых этот сотрудник был менеджером.
- CEO никогда не покидает компанию.
- Каждый сотрудник получает зарплату — некоторое количество справидляндских долларов в год. Зарплата сотрудника никогда не меняется.
- Разные сотрудники могут получать разную зарплату, и зарплата сотрудника не обязательно коррелирует с тем, где в структуре компании он находится.

Правительство Справидляндии только что приняло еще один дополнительный закон:

- Разность между наибольшей и наименьшей зарплатой во всей компании не должна превышать  $D$  долларов.

Мэри — CEO Fairland General Stuff Corporation, и она должна удостовериться, что ее компания исполняет новый закон. Для этого может потребоваться уволить некоторых сотрудников. У нее есть список всех сотрудников компании, их менеджеров и зарплат. Какое наибольшее число сотрудников она может оставить в компании, считая себя?

### Формат входных данных

Первая строка содержит целое число  $T$  — количество тестов. Затем идет  $T$  тестов. Каждый тест начинается со строки, содержащей два целых числа  $N$  и  $D$  — количество сотрудников и максимальная разрешенная разница в зарплате двух сотрудников. В следующей строке даны четыре целых числа  $S_0$ ,  $A_s$ ,  $C_s$ ,  $R_s$ , и в следующей строке даны еще четыре числа  $M_0$ ,  $A_m$ ,  $C_m$  and  $R_m$ . Эти восемь чисел определяют следующие последовательности:

$$S_{i+1} = (S_i * A_s + C_s) \bmod R_s$$
$$M_{i+1} = (M_i * A_m + C_m) \bmod R_m$$

Мэри имеет номер 0, остальные сотрудники пронумерованы от 1 до  $N - 1$ , включительно. Зарплата сотрудника  $i$  равна  $S_i$ . Для сотрудника номер  $i$  (кроме Мэри,  $i > 0$ ), менеджером является сотрудник номер  $M_i \bmod i$ . (Обратите внимание, что  $M_0$  не влияет на менеджера Мэри, у нее его нет!)

- $1 \leq T \leq 100$
- $0 \leq S_0 < R_s$
- $0 \leq M_0 < R_m$
- $0 \leq A_s, A_m \leq 1\,000$

- $0 \leq C_s, C_m \leq 10^9$

## Формат выходных данных

Для каждого теста, выведите одну строку, содержащую «Case # $x$ :  $y$ », где  $x$  — номер теста (начиная с 1), а  $y$  — наибольшее количество сотрудников, которых Мэри может оставить в компании, чтобы все законы были выполнены.

## Система оценки

Small dataset (3pt):

- $1 \leq N \leq 1000$
- $1 \leq D \leq 1000$
- $1 \leq R_s, R_m \leq 1000$

Large dataset (9pt):

- $1 \leq N \leq 10^6$
- $1 \leq D \leq 10^6$
- $1 \leq R_s, R_m \leq 10^6$

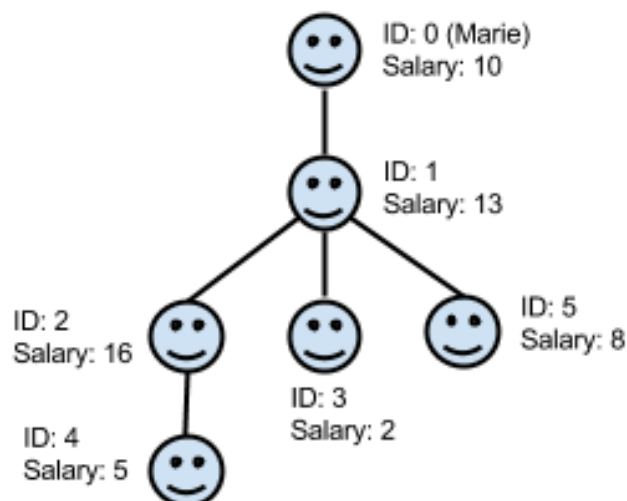
## Пример

стандартный ввод	стандартный вывод
3 1 395 18 246 615815 60 73 228 14618 195 6 5 10 1 3 17 5 2 7 19 10 13 28 931 601463 36 231 539 556432 258	Case #1: 1 Case #2: 3 Case #3: 5

## Замечание

In Case #1, the company has only a CEO and no other employees, but it does not violate any of the laws, so no changes need to be made.

Here is the org chart for Case #2:



The optimal strategy is to save employees 0, 1, and 5 (who have salaries of 10, 13, and 8, respectively). It is not possible to save employee 2, for example, because her salary is more than 5 away from employee 0's salary of 10; since employee 0 cannot be laid off, employee 2 must be laid off (along with all employees who report to her).

If you want to check your sequences for employees 1 through 5, they are:

*S*: 13, 16, 2, 5, 8

*M*: 17, 3, 13, 14, 16

Manager numbers:  $17 \bmod 1 = 0$ ,  $3 \bmod 2 = 1$ ,  $13 \bmod 3 = 1$ ,  $14 \bmod 4 = 2$ ,  $16 \bmod 5 = 1$

## Задача B. Smoothing Window

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	512 мегабайт

Адамма изучает температуру окружающей среды. Каждую минуту она записывает текущую температуру как целое число, создавая длинную последовательность из целых чисел:  $x_1, x_2, \dots, x_N$ . (Адамма использует свою специальную шкалу для измерения температуры, а не одну из известных вам, например Цельсия или Кельвина, поэтому значения могут быть большие или отрицательные!) Она часто рисует графики этой температуры на своем компьютере.

Сегодня утром она решила вычислить скользящее среднее этого списка, чтобы построить сглаженный график. Она использует сглаживающее окно длины  $K$ , это значит, что она преобразует последовательность длины  $N$  значений температуры в последовательность длины  $(N - K + 1)$  средних значений температуры:  $s_1, s_2, \dots, s_{N-K+1}$ . Каждое  $s_i$  равно средней величине значений  $x_i, x_{i+1}, \dots, x_{i+K-1}$ . Исходные значения  $x_i$  — целые, но некоторые из значений  $s_i$  могут быть рациональными.

К сожалению, Адамма забыла сохранить исходную последовательность температур! И теперь она хочет узнать ответ на такой вопрос — какова была разность между наибольшей и наименьшей температурой? Другими словами, она хочет вычислить  $\max(x_1, \dots, x_N) - \min(x_1, \dots, x_N)$ . Но она знает только  $N, K$  и сглаженную последовательность.

После некоторых раздумий, Адамма поняла, что может существовать несколько корректных ответов. В таком случае, она хочет узнать наименьший возможный ответ среди всех подходящих исходных последовательностей, которые могли привести к такой сглаженной последовательности.

### Формат входных данных

Первая строка содержит одно целое число  $T$  — количество тестов. Затем идет  $T$  тестов. Первая строка теста содержит два целых числа  $N$  и  $K$ . Вторая строка теста содержит последовательность целых чисел  $sum_1, sum_2, \dots, sum_{N-K+1}$ .  $s_i = sum_i / K$ .

- $1 \leq T \leq 100$
- $2 \leq K \leq N$
- $-10\,000 \leq sum_i \leq 10\,000$

### Формат выходных данных

Для каждого теста, выведите одну строку, содержащую «Case # $x$ :  $y$ », где  $x$  — номер теста (начиная с 1), а  $y$  — наименьшая возможная разность между максимальной и минимальной температурой.

### Система оценки

Small dataset (6pt):

- $2 \leq N \leq 100$

Large dataset (7pt):

- $2 \leq N \leq 1\,000$
- $2 \leq K \leq 100$

### Пример

стандартный ввод	стандартный вывод
3	Case #1: 5
10 2	Case #2: 0
1 2 3 4 5 6 7 8 9	Case #3: 12
100 100	
-100	
7 3	
0 12 0 12 0	

## Замечание

In Case #1, the smoothed sequence is:

0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5

The integer sequence that gives the smallest difference is:

0, 1, 1, 2, 2, 3, 3, 4, 4, 5

Note that the sequence:

0.5, 0.5, 1.5, 1.5, 2.5, 2.5, 3.5, 3.5, 4.5, 4.5

Would give the same smoothed sequence with a maximum difference of 4, but this is not a valid answer because the original temperatures are known to have been integers.

In Case #2, all we know is that the sum of the 100 original values was  $-100$ . It's possible that all of the original values were exactly  $-1$ , in which case the difference between the largest and smallest temperatures would be 0, which is as small as differences get!

In Case #3, the original sequence could have been:

$-4, 8, -4, 8, -4, 8, -4$

## Задача C. Runaway Quail

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	512 мегабайт

О нет — ваши  $N$  домашних перепелок разбежались! Вы сейчас находитесь в точке 0 на прямой,  $i$ -я перепелка начинает бежать с некоторой ненулевой (положительной или отрицательной) точки  $P_i$  на этой прямой, в метрах, и будет всегда бежать от вас с постоянной целой скоростью  $S_i$  метров в секунду. Вы можете бежать с постоянной скоростью  $Y$  метров в секунду, и вы можете мгновенно изменять направление движения, когда захотите. Обратите внимание, что перепелки убегают постоянно бегут от вас, даже если вы сейчас не движетесь по направлению к ним. Когда вы оказываетесь в одной точке с какой-нибудь перепелкой, вы ловите ее (это не требует дополнительного времени).

Чему равно минимальное количество секунд, которое вам понадобится, чтобы поймать всех перепелок?

### Формат входных данных

Первая строка содержит одно целое число  $T$  — количество тестов. Затем даны  $T$  тестов. Первая строка теста содержит два целых числа  $Y$  и  $N$  — ваша скорость и количество перепелок. В следующей строке даны  $N$  целых чисел  $P_i$  — исходные позиции перепелок. В следующей строке даны  $N$  целых чисел  $S_i$  — скорости перепелок.

- $1 \leq T \leq 100$
- $2 \leq Y \leq 1000$
- $-10^7 \leq P_i \leq 10^7$ ;  $P_i \neq 0$
- $1 \leq S_i < Y$

### Формат выходных данных

Для каждого теста, выведите одну строку, содержащую «Case # $x$ :  $y$ », где  $x$  — номер теста (начиная с 1), а  $y$  — минимальное количество секунд, которое понадобится, чтобы поймать всех перепелок.

Ответ будет считаться правильным, если его абсолютная или относительная погрешность от правильного ответа не превышает  $10^{-6}$ .

### Система оценки

Small dataset (8pt):

- $1 \leq N \leq 25$

Large dataset (15pt):

- $1 \leq N \leq 500$

### Пример

стандартный ввод	стандартный вывод
2 4 3 -3 -6 -9 3 2 1 2 2 1 -1 1 1	Case #1: 3.0000000000000000 Case #2: 5.0000000000000000

## Замечание

In Case #1, you can run to the left and catch all three quail at the same time, 12 meters to the left of the starting position, which takes 3 seconds.

In Case #2, one optimal strategy is to run to the left until the second quail is caught at  $-2$  m, which takes one second, and then run to the right in pursuit of the first quail, which you will catch at 6 m, taking four more seconds.

## Задача D. Log Set

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	512 мегабайт

*Степенным множеством* множества  $S$  назовем множество всех подмножеств  $S$  (включая пустое и само  $S$ ). По множеству получить его степенное множество просто, но в этой задаче мы будем проделывать обратную операцию.

Мы начали с множества  $S$  целых, не обязательно различных, чисел. Нашли его степенное множество, и затем заменили каждый элемент в степенном множестве на сумму элементов в нем. Получили множество  $S'$ . Например, если  $S = \{-1, 1\}$ , его степенное множество равно  $\{\{\}, \{-1\}, \{1\}, \{-1, 1\}\}$ , и  $S' = \{0, -1, 1, 0\}$ .  $S'$  может содержать повторяющиеся значения, поэтому, если  $S$  содержит  $N$  элементов, то  $S'$  содержит ровно  $2^N$  элементов.

Дано значения различных элементов в  $S'$  и их количества, можете ли вы определить исходное множество  $S$ ? Гарантируется, что  $S$  существует. Если существует несколько различных ответов, выведите лексикографически минимальный. Чтобы сравнить два множества лексикографически, рассмотрим их элементы в возрастающем порядке и сравним лексикографически полученные два массива.

### Формат входных данных

Первая строка содержит одно целое число  $T$  — количество тестов. Затем даны  $T$  тестов. Первая строка теста содержит одно целое число  $P$  — количество различных значений в  $S'$ . Следующая строка содержит  $P$  различных целых чисел  $E_1, E_2, \dots, E_P$  — значения, которые присутствуют в  $S'$ , отсортированные в возрастающем порядке. Следующая строка содержит  $P$  чисел  $F_1, F_2, \dots, F_P$  — количество раз, которое соответствующее значение присутствует в  $S'$ .

- $1 \leq T \leq 100$
- $1 \leq P \leq 10\,000$
- $1 \leq F_i$

### Формат выходных данных

Для каждого теста, выведите одну строку, содержащую «Case #x: », где  $x$  — номер теста (начиная с 1), после чего элементы исходного множества  $S$  в неубывающем порядке.

### Система оценки

Small dataset (6pt):

- $S$  содержит от 1 до 20 элементов
- $0 \leq E_i \leq 10^8$

Large dataset (19pt):

- $S$  will содержит от 1 до 60 элементов
- $-10^{10} \leq E_i \leq 10^{10}$



## Пример

стандартный ввод	стандартный вывод
5	Case #1: 1 2 4
8	Case #2: 1 1 1
0 1 2 3 4 5 6 7	Case #3: 0 0 1 3
1 1 1 1 1 1 1 1	Case #4: -1 1
4	Case #5: -2 1 1
0 1 2 3	
1 3 3 1	
4	
0 1 3 4	
4 4 4 4	
3	
-1 0 1	
1 2 1	
5	
-2 -1 0 1 2	
1 2 2 2 1	

## Замечание

Note that Cases #4 and #5 are not within the limits for the Small dataset.

In Case #4,  $S = \{-1, 1\}$  is the only possible set that satisfies the conditions. (Its subsets are  $\{\}$ ,  $\{-1\}$ ,  $\{1\}$ , and  $\{-1, 1\}$ . Those have sums 0,  $-1$ , 1, and 0, respectively, so  $S'$  has one copy of  $-1$ , two copies of 0, and one copy of 1, which matches the specifications in the input.)

For Case #5, note that  $S = \{-1, -1, 2\}$  also produces the same  $S' = \{-2, -1, -1, 0, 0, 1, 1, 2\}$ , but  $S = \{-2, 1, 1\}$  is earlier than  $\{-1, -1, 2\}$ , since at the first point of difference,  $-2 < -1$ . So  $-1 -1 2$  would not be an acceptable answer.  $1 -2 1$  would also be unacceptable, even though it is the correct set, because the elements are not listed in nondecreasing order.

## Задача E. River Flow

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	512 мегабайт

Вы живете в городе, расположенном на берегах Бинарной реки. Вода в реку собирается из нескольких притоков, которые начинаются высоко в горах. К сожалению для вашего города, в горах живет несколько фермеров, которые используют немного воды из притоков реки для полива.

Давным-давно, город заключил с фермерами сделку, разрешающую тем использовать воду из реки для полива раз в два дня. На один день они используют воду для полива, а на другой они дают ей течь дальше. Результат этой сделки был катастрофичен! Из-за того, что поливы фермеров были синхронизованы, на один день река полностью пересыхала, а на другой — подтапливала город.

Чтобы решить эту проблему, горожане снова пришли к фермерам и попросили каждого выбрать некоторую целую степень 2 (дело же происходит на берегах Бинарной реки, в конце концов) между 1 и  $D$ , включительно. И переключать использование воды (либо переставать использовать, если до этого использовал, либо начинать использовать, если до этого не использовал) раз во столько дней. Несколько фермеров могли выбрать одно и то же число, и некоторые числа мог не выбрать никто. Идея была в том, чтобы чтобы использование воды стало более равномерным, и наводнения случались реже.

Все это произошло давным-давно, но недавно вы и другие горожане стали подозревать, что фермеры не придерживаются соглашения. (Вы даже не уверены, сколько именно фермеров сейчас живет в горах!) Единственные данные, которые у вас есть — это измерения количества воды, которое протекало через город в каждый из  $N$  подряд идущих дней. Могут ли фермеры придерживаться договора?

По каждому притоку течет поток величины 1, а потом в реке равен сумме потоков в притоках, которые не используются для полива в текущий день. Каждый приток использует максимум один фермер. Обратите внимание, что фермеры должны были начать придерживаться договора уже давно, однако они могли начать это делать в разные дни.

### Формат входных данных

Первая строка содержит одно целое число  $T$  — количество тестов. Далее даны  $T$  тестов. Каждый тест начинается со строки, содержащей два целых числа  $N$  и  $D$ . Следующая строка содержит  $N$  целых чисел  $d_i$  — величина потока на  $i$ -й день.

- $1 \leq T \leq 50$
- $D$  will be a power of 2
- $1 \leq D \leq \lfloor \frac{N}{2} \rfloor$

### Формат выходных данных

Для каждого теста, выведите одну строку, содержащую «Case # $x$ :  $M$ », где  $x$  — номер теста (начиная с 1), а  $M$  — наименьшая возможное количество фермеров, которые могут жить в горах и использовать воду для полива, придерживаясь договора.

Если вы уверены, что не существует способа объяснить наблюдаемые величины действиями фермеров, придерживающихся договора, выведите «CHEATERS!» вместо числа.

### Система оценки

Small dataset (10pt):

- $1 \leq N \leq 50$
- $0 \leq d_i \leq 5$

Large dataset (17pt):

- $1 \leq N \leq 5\,000$
- $0 \leq d_i \leq 1\,000$

### Пример

стандартный ввод	стандартный вывод
4	Case #1: 0
5 2	Case #2: CHEATERS!
2 2 2 2 2	Case #3: 2
6 2	Case #4: 3
1 1 1 0 0 0	
8 4	
2 1 1 0 0 1 1 2	
8 4	
0 1 1 3 1 2 2 2	

### Замечание

Case #1 is consistent with two tributary streams with no farmers drawing from them.

Case #2 could a single tributary stream being diverted every 4 days. However,  $D$  is 2 in this case, so this farmer is breaking the agreement.

Case #3 could be two farmers each with a diversion cycle of 4 days.

Case #4 could be three farmers with diversion cycles of 1, 2 and 4 days.