

Пустоши.

Орк вышел из Оргриммара в Пустоши и собирается за ближайшие семь недель, покروшив неисчислимое количество кабанов, прокачаться до пятидесятого уровня. Он бесстрашно собирает на себя пак из N кабанов. Предположим, что у них есть номера от 0 до $N - 1$. Ему, конечно же, нужно их всех победить. У орка пока нет AoE спеллов, поэтому ему приходится биться с кабанамии по очереди. К сожалению, живые кабаны вполне себе способны атаковать его одновременно.

У каждого кабана есть две важные характеристики: количество причиняемого урона в секунду (dps) и количество очков жизни (hp). У орка dps равен 1, т.е. за секунду он способен у одного кабана уменьшить количество очков жизни на единицу. Когда у кабана количество очков жизни доходит до 0, он умирает и не способен атаковать.

Вам дано два набора чисел, каждый из N чисел: dps и hp . Кабан с номером i наносит dps_i урона в секунду и имеет hp_i очков жизни. Мы будем считать, что количество очков жизни у орка достаточное, чтобы их всех победить. Найдите общее количество урона, которое нанесут кабаны в течение битвы.

Предупреждение:

Пожалуйста, не огорчайте орка тем фактом, что очень скоро кабаны перестанут приносить ему опыт.

Входные данные:

- на первой строке одно целое число: N ;
- на второй строке N целых чисел через пробел: dps_i ;
- на третьей строке N целых чисел через пробел: hp_i .

Выходные данные:

- на единственной строке одно число: полученный урон.

Ограничения:

- $1 \leq N \leq 30$;
- $1 \leq dps_i \leq 30$;
- $1 \leq hp_i \leq 30$.

Примеры:

Дано	Надо
2 1 2 3 4	17
3 1 1 1 1 3 1	10
10 1	55
6 20 12 10 10 23 10 5 7 7 5 7 7	1767
6 5 7 7 5 7 7 20 12 10 10 23 10	1998
10 30 2 7 4 7 8 21 14 19 12 2 27 18 19 14 8 25 13 21 30	11029
1 1 1	1

Победители.

Почти в любой аркаде есть список победителей, где показываются имена и баллы игроков, набравшие больше всего баллов. Поскольку вам скоро писать ЕГЭ, эта задача будет только о баллах.

Рейтинги в списке победителей отображаются в неубывающем порядке. Места вычисляются обычно по номеру в таком отсортированном списке, но если несколько человек получили одинаковые баллы, то место обычно вычисляется как место первого участника с таким количеством баллов. Например, для вот таких баллов в списке 100 90 90 80 места участников будут 1 2 **2** 4.

Вам будет дано общее количество записей в списке победителей P , список длины K с количествами набранных баллов уже сыгравших игроков S и баллы только что сыгравшего игрока N . Найдите и выведите место этого игрока в списке победителей. Если игрок набрал недостаточно баллов, чтобы попасть в список победителей, выведите "GG" (кавычки для ясности). Заметьте, что если все места в списке уже заняты, игрок попадёт в него только если он набрал больше баллов, чем кто-либо в списке.

Входные данные:

- на первой строке три целых числа через пробел: N K P ;
- на второй строке набор целых чисел через пробел: S .

Выходные данные:

- на единственной строке: целое число, номер в списке победителей, или "GG".

Ограничения:

- $1 \leq P \leq 50$;
- $0 \leq K \leq P$;
- $0 \leq S_i \leq 2000000000$;
- S отсортированы в невозрастающем порядке.
- $0 \leq N \leq 2000000000$.

Примеры:

Дано	Надо
90 3 10 100 90 80	2
0 0 50	1
1 10 10 10 9 8 7 6 5 4 3 2 1	GG
1 10 10 10 9 8 7 6 5 4 3 3 0	10
1337 50 50 2000000000 19539 19466 19146 17441 17002 16348 16343 15981 15346 14748 14594 13752 13684 13336 13290 12939 12208 12163 12133 11621 11119 10872 10710 10390 9934 9296 8844 8662 8653 8168 7914 7529 7354 6016 5428 5302 5158 4853 4538 4328 3443 3222 2107 2107 1337 951 586 424 31	46

Нелинейный порядок.

Скульптор Артемий, разглядывая свои поделия, обнаружил, что хоть они и стоят в одной линии, но всё же несколько хаотично. У каждой скульптуры есть своя высота, и сейчас они стоят просто вразнобой. Артемий попробовал отсортировать их в порядке неубывания высоты, но оказалось, что это тоже не очень красиво. Он решил внести свою изюминку: поставить их слегка не по порядку.

Вам дан набор чисел s , представляющих высоты статуй. Нужно помочь Артемию переупорядочить статуи так, чтобы они стояли не в неубывающем порядке. Переставьте числа в наборе так, чтобы они образовывали словарно наименьшую последовательность, и выведите их. Если переупорядочить их так невозможно, выведите "NO" (кавычки здесь для ясности).

Матчасть:

Последовательность чисел $s_0 s_1 s_2 \dots s_{N-1}$ называется отсортированной в порядке неубывания, если $s_i \leq s_{i+1}$ для любого i от 0 до $N-2$ включительно. (Последовательность из одного числа всегда считается отсортированной по неубыванию.) Например, последовательности 1 2 3, 1 1 1 и 2 2 4 4 6 6 отсортированы в неубывающем порядке, а последовательности 1 2 1, 7 4 6 и 3 2 1 — нет.

Последовательность чисел A называется словарно меньшей, чем B , если A содержит меньшее число в первой позиции с левого края, в которой они различаются. Например, 1 2 3 словарно меньше, чем 1 3 2, поскольку $2 < 3$.

Входные данные:

- на первой строке одно целое число: N , количество статуй;
- на второй строке набор из N целых чисел через пробел: s .

Выходные данные:

- на единственной строке:
 - либо набор целых чисел: искомая последовательность высот статуй;
 - либо строка "NO".

Ограничения:

- $1 \leq N \leq 50$;
- $1 \leq s_i \leq 1000$.

Примеры:

Дано	Надо	Комментарий
2 1 2	2 1	Есть только одна последовательность не в неубывающем порядке.
3 1 2 3	1 3 2	Есть одна неубывающая, а 1 3 2 — словарно минимальная из оставшихся.
3 7 2 2	2 7 2	Статуи могут быть одинаковой высоты.
1 1000	NO	Одно число всегда отсортировано в неубывающем порядке.
2 1 1	NO	Так тоже бывает.
4 1 2 4 3	1 2 4 3	
17 2 8 5 1 10 5 9 9 3 10 5 6 6 2 8 2 10	1 2 2 2 3 5 5 5 6 6 8 8 9 10 9 10 10	

Плохой день.

В Флатландии есть Прямоугольный парк, в котором все дорожки и газоны выровнены по точкам сетки с шагом 1 на обычной прямоугольной системе координат. Крис увидел своего друга, окликнул его, но тот не услышал. Тогда Крис побежал к нему по прямой через газон. Но с другой стороны газона его ожидал строгий и раздосадованный садовник. Крис не заметил, как протоптался по нескольким аккуратно высаженным растениям. Друга он не догнал, так теперь ещё и садовник требует возместить парку убытки. Плохой день, короче говоря.

Посчитать убытки оказалось тоже непросто. Известно, что растения были высажены в точках с целыми координатами. Пересекая газон, Крис сначала наступил на растение в точке (x_1, y_1) , а в самом конце на растение в точке (x_2, y_2) . Известно также, что в каждой целочисленной точке, принадлежащей прямоугольнику, стороны которого параллельны осям координат, а две вершины совпадают с координатами начальной и конечной точкой досадного пути Криса, находилось по одному растению. Сколько растений было затоптано?

Входные данные:

- на первой строке четыре числа через пробел: x_1 y_1 x_2 y_2 .

Выходные данные:

- на единственной строке одно число: количество затоптанных растений.

Ограничения:

- $0 \leq x_1, y_1, x_2, y_2 \leq 50$;
- начальная и конечная точка — разные.

Примеры:

Дано	Надо
1 1 5 5	5
0 0 1 1	2
50 48 0 0	3
0 0 42 36	7

Шифрование.

Феликс просто параноик, когда дело доходит до безопасности его переписки. Он использует шифр подстановки, чтобы зашифровать свои письма. Каждая буква в письме заменяется на соответствующую букву из алфавита подстановки. Алфавит подстановки это перестановка 26 букв английского алфавита. В рамках этой задачи алфавит будет состоять только из строчных букв 'a' – 'z'.

Например, если письмо Феликса это "hello", и его шифр превращает 'h' в 'q', 'e' в 'w', 'l' в 'e' и 'o' в 'r', закодированное письмо будет "qweer". Если шифр превращает 'h' в 'a', 'e' в 'b', 'l' в 'c' и 'o' в 'd', то закодированное письмо будет "abccd".

Зная изначальный текст письма, зашифруйте сообщение шифром подстановки с таким алфавитом подстановки, чтобы получившееся сообщение было словарно наименьшим. В примере выше второй шифр как раз создаёт словарно наименьший результат "abccd".

Матчасть:

Строка A называется словарно меньшей, чем B, если A содержит ранее идущий в алфавите символ в первой позиции с левого края, в которой они различаются. Например, "abc" словарно меньше, чем "acb", поскольку 'b' идёт раньше, чем 'c'.

Входные данные:

- на единственной строке: сообщение msg.

Ограничения:

- msg содержит от 1 до 50 символов включительно;
- msg содержит только строчные буквы английского алфавита 'a' – 'z'.

Примеры:

Дано	Надо
hello	abccd
abcd	abcd
topcoder	abcdbefg
encryption	abcdefghijkl

Калькулятор.

Виктор любит калькуляторы. Особенно большие старые калькуляторы, у которых на экране горят яркие красные цифры. Но у них есть серьёзная проблема: они быстро разряжаются. Скорость разряда батарей зависит от того, как много сегментов нужно зажечь на экране, чтобы показать число. Виктор хотел бы проверить, с какой скоростью числа разряжают его калькулятор.

0 123456789

Входные данные:

- на первой строке одно целое число: N .

Выходные данные:

- на единственной строке одно число: количество сегментов, которые зажгутся на экране, когда Виктор наберёт число N .

Ограничения:

- все числа целые;
- $0 \leq N \leq 10^{50}$;
- число может начинаться с одного и более нулей.

Примеры:

Дано	Надо
13579	21
02468	28
73254370932875002027963295052175	157

Электроника ИМ-02.

Волк пытается поймать падающие с неба фрукты. Игра плоская. Для простоты будем считать, что Волк и фрукты это точки на плоскости. Волк начинает в точке с координатами $(0, 0)$. Волк может двигаться вдоль оси Ox со скоростью не больше 1 единицы расстояния в секунду. Например, ему нужно не менее 3 секунд, чтобы переместиться из точки $(-2, 0)$ в $(1, 0)$.

Нам заранее известны координаты x_i и y_i фруктов в момент времени 0. Все фрукты падают вниз с постоянной скоростью в 1 единицу расстояния в секунду. Таким образом, фрукт, находящийся в данный момент времени в точке (a, b) будет находиться в точке $(a, b-t)$ через t секунд.

Волк поймает фрукт, если он будет находиться в то же время в той же точке, где и фрукт. Сможет ли Волк поймать все фрукты? Выведите "YES" (кавычки для ясности) если да и "NO" если нет.

Входные данные:

- на первой строке одно целое число: N , количество фруктов;
- на второй строке N пар целых чисел через пробел: $x_i y_i$.

Выходные данные:

- на единственной строке: "YES" или "NO".

Ограничения:

- $1 \leq N \leq 50$;
- $-1000 \leq x_i \leq 1000$;
- $0 \leq y_i \leq 1000$.

Примеры:

Дано	Надо
3 -1 1 1 3 0 4	YES
1 -3 2	NO
3 -1 1 1 2 0 4	NO
3 0 9 -1 1 1 3	YES
8 70 141 -108 299 52 402 -70 280 84 28 -29 363 66 427 -33 232	NO
15 -175 320 -28 107 -207 379 -29 72 -43 126 -183 445 -175 318 -112 255 -183 445 -31 62 -25 52 -66 184 -114 247 -116 245 -66 185	YES
4 0 0 0 0 0 0 0 0	YES

Шаг вперёд.

Тайлер устроил погром в школьном театре, и теперь приговорён к 200 часам обязательных работ в этой школе. Ему нужно за день убрать k помещений. Однако он также помогает Норе готовиться к Смотру искусств, а поэтому хотел бы потратить на это как можно меньше времени. Тайлер уже знает, сколько времени нужно на уборку каждого помещения в школе, и хотел бы узнать, сколько времени ему придётся потратить.

Входные данные:

- на первой строке одно целое число: k ;
- на второй строке набор целых чисел через пробел: количество времени (в секундах) на уборку каждого из помещений школы.

Выходные данные:

- на единственной строке одно число: минимальное время на уборку k помещений в секундах.

Ограничения:

- в школе от 1 до 10000 помещений;
- каждое помещение убирается от 1 до 1000 секунд;
- k от 1 до количества помещений.

Примеры:

Дано	Надо
2 1 5 3 4	4
3 1 5 4	10
1 2 2 4 5 3	2
39 973 793 722 573 521 568 845 674 595 310 284 794 913 93 129 758 108 433 181 163 96 932 703 989 884 420 615 991 364 657 421 336 801 142 908 321 709 752 346 656 413 629 801	20431