Задача A. Дерево Ли Чао

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дано множество из n полуинтервалов $y = a_i x + b_i$ (где $l_i \leqslant x < r_i$) и q запросов вида:

- `0` l r a b. Добавить в множество полуинтервал $y = ax + b$ ($l \leqslant x < r$)
- `1` p. Найти минимальное значение y в точке $x = p$ по полуинтервалам из множества. Если такого y не существует, вывести «`INFINITY`».

Формат входных данных

В первой строке указана пара чисел n и q ($1 \leqslant n, q \leqslant 2 \cdot 10^5$).

В следующих n строках вводятся по 4 целых числа, задающих полуинтервалы: l_i, r_i, a_i, b_i ($-10^9 \leqslant l_i < r_i \leqslant 10^9$, $|a_i| \leqslant 10^9$, $|b_i| \leqslant 10^{18}$).

Затем следуют q запросов вида:

- `0` l r a b ($-10^9 \leqslant l < r \leqslant 10^9$, $|a| \leqslant 10^9$, $|b| \leqslant 10^{18}$)
- `1` p ($|p| \leqslant 10^9$)

Формат выходных данных

Для каждого запроса типа 1 выведите ответ на него в отдельной строке.

Примеры

стандартный ввод	стандартный вывод
2 8 -3 3 -1 -1 0 7 0 1 1 -1 1 -2 1 0 1 2 0 -4 2 0 -10 1 -2 1 0 1 2	0 1 -1 -3 -10 -10 -3
1 2 -10 0 0 0 1 0 1 -1	INFINITY 0

Задача B. Коды, сохраняющие порядок

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Двоичный код — это код, где каждому символу сопоставляется последовательность из единиц и нулей. Код называется префиксным, если ни одно кодовое слово не является префиксом другого. Код называется сохраняющим порядок, если лексикографический порядок кодовых слов совпадает с алфавитным порядком символов.

Рассмотрим текст над алфавитом, содержащим n символов, в котором a_1 раз встречается первый символ, a_2 раз встречается второй символ, $\cdots$, a_n раз встречается n-й символ. Длина текста после кодирования его префиксным кодом, где первому символу сопоставлена строка длины l_1, второму — строка длины l_2, и т. д., будет равна $a_1 \cdot l_1 + a_2 \cdot l_2 + \cdots + a_n \cdot l_n$.

Требуется найти сохраняющий порядок префиксный код, минимизирующий длину закодированного текста.

Формат входных данных

Первая строка содержит число n — число символов в алфавите ($2 \leqslant n \leqslant 2000$). Следующая строка содержит n целых чисел — сколько раз каждый символ встречается в тексте: $a_1, a_2, \cdots, a_n$. Числа положительные и не превосходят 10^9.

Формат выходных данных

Выведите n двоичных последовательностей — искомый код.

Пример

стандартный ввод	стандартный вывод
5 1 8 2 3 1	00 01 10 110 111

Задача C. Задача «ИЛИ»

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Дан массив a из n целых чисел. Стоимость отрезка массива — побитовое «ИЛИ» его элементов. Пусть вы разбили массив на k непустых отрезков и посчитали сумму их стоимостей. Какое максимальное значение вы можете получить?

Формат входных данных

Первая строка содержит два целых числа n и k ($1 \leqslant k \leqslant n \leqslant 2 \cdot 10^5$).

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i < 2^{30}$) — элементы массива a.

Формат выходных данных

Выведите одно целое число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
5 2 1 4 3 4 8	20

Задача D. Ракообразные

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

После детального изучения вьюнков Галапагосских островов Чарльз Дарвин занялся изучением ракообразных. Для этого он выловил несколько крабов. А именно, у него есть N крабов, которых он упорядочил по разным параметрам и присвоил им номера от 1 до N. Выяснилось, что все крабы агрессивны, краб i имеет уровень агрессии A_i, где A_i положительно.

У Дарвина есть K аквариумов, в которые он планирует поместить крабов, чтобы они жили в естественной среде. Однако он уже потратил достаточно много времени, чтобы упорядочить крабов, поэтому он не хочет изменять их порядок слишком сильно. Он решил, что поместит первых T_1 крабов в первый аквариум, следующих T_2 крабов во второй и так далее, последних T_K крабов в последний аквариум. Разумеется, должно выполняться равенство $T_1 + T_2 + \cdots + T_K = N$, а каждое число T_i должно быть неотрицательным, но других ограничений нет, Дарвин может выбрать любые значения T_i.

Но у крабов тоже есть чувства, в частности, страх. На самом деле крабы жутко боятся друг друга, особенно слишком агрессивных особей. Говоря конкретно, краб i боится всех других крабов, которые живут с ним в одном аквариуме, и количество страха, которое он испытывает по отношению к крабу j, равно его уровню агрессии A_j. Таким образом, общее количество страха, которое испытывает краб i, равно сумме уровней агрессии A_j для всех номеров j, других крабов, живущих в том же аквариуме, что и краб i.

Дарвин не хочет подвергать своих крабов слишком большому стрессу, ведь чувства крабов имеют значение. Поэтому он пытается минимизировать суммарное количество страха, которое будут испытывать все крабы, выбрав распределение их по K аквариумам. К сожалению, в 30-е (1830-е, как вы понимаете) у людей еще не было особо мощных компьютеров, так что Дарвину приходится обратиться к вам за помощью. Пожалуйста, помогите ему, написав программу, которая вычислит минимальное возможное суммарное количество страха, которое будут испытывать крабы, по заданным N, K и A_i.

Формат входных данных

Первая строка стандартного потока ввода содержит два положительных целых числа N и K ($1 \leqslant K \leqslant N \leqslant 10^5$) — количество крабов и количество аквариумов. Следующая строка содержит N положительных целых чисел: $A_1, A_2, \ldots, A_N$ ($1 \leqslant A_i \leqslant 10^7$) — уровни агрессии крабов.

Формат выходных данных

В единственной строке стандартного потока вывода следует вывести неотрицательное число: минимальное возможное суммарное количество страха, которое будут испытывать крабы.

Примеры

стандартный ввод	стандартный вывод
8 4 8 1 2 3 9 1 9 1	32
6 3 10 3 8 5 4 7	37

Замечание

В этом примере 8 крабов, которые будут жить в 4 аквариумах. Оптимальное распределение такое: краб 1 в аквариуме 1; крабы 2, 3 и 4 в аквариуме 2; крабы 5 и 6 в аквариуме 3; крабы 7 и 8 в аквариуме 4. Краб 1 не испытывает страха; краб 2 испытывает 5 единиц страха; краб 3 испытывает 4 единицы страха; краб 4 испытывает 3 единицы страха; краб 5 испытывает 1 единицу страха; краб 6 испытывает 9 единиц страха; краб 7 испытывает 1 единицу страха, и краб 8 испытывает 9 единиц страха. Суммарное количество страха, испытываемого крабами, равно $0+5+4+3+1+9+1+9 = 32$.

Задача E. Очередная задача минимизации

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан массив из n чисел $a_1 \ldots a_n$. Стоимостью подотрезка элементов в массиве назовем количество неупорядоченных пар различных позиций внутри подотрезка, содержащих одинаковые элементы. Разбейте массив на k непересекающихся непустых подотрезков таких, что сумма их стоимостей минимальна. Каждый элемент массива должен попасть ровно в один подотрезок.

Формат входных данных

Первая строка содержит два целых числа n и k $(2 \leqslant n \leqslant 10^5,\ 2 \leqslant k \leqslant \min(n, 20))$ — размер массива и количество отрезков, на которые надо его разбить.

Следующая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ $(1 \leqslant a_i \leqslant n)$ — элементы массива.

Формат выходных данных

Выведите одно число — минимальную стоимость разбиения массива на подотрезки.

Примеры

стандартный ввод	стандартный вывод
7 3 1 1 3 3 3 2 1	1
10 2 1 2 1 2 1 2 1 2 1 2	8
13 3 1 2 2 2 1 2 1 1 1 2 2 1 1	9

Замечание

В первом примере оптимально разбить последовательность на три подпоследовательности: $[1]$, $[1, 3]$, $[3, 3, 2, 1]$. Стоимости равны 0, 0 и 1, поэтому ответ равен 1.

Во втором примере оптимально разбить подпоследовательность на две половины. Стоимость каждой половины равна 4.

В третьем примере оптимально разбить следующим образом: $[1, 2, 2, 2, 1]$, $[2, 1, 1, 1, 2]$, $[2, 1, 1]$. Стоимости равны 4, 4, 1.

Задача F. Красные и синие лампочки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Есть n лампочек с номерами $1, 2, \ldots, n$. Надо ровно m из них сделать красными, а все остальные — синими, чтобы максимизировать следующую величину: по всем $i \in [1, n-1]$: если лампочки i и $i+1$ разного цвета, то добавить к ответу a_i.

Формат входных данных

Первая строка содержит два числа — n и m ($2 \leqslant n \leqslant 2 \cdot 10^5, 1 \leqslant m \leqslant n-1$).

Вторая строка содержит $n-1$ число $a_1, a_2, \ldots, a_{n-1}$ ($1 \leqslant a_i \leqslant 10^9$).

Формат выходных данных

Выведите максимальную сумму, которую можно получить, если ровно m лампочек красные.

Примеры

стандартный ввод	стандартный вывод
6 2 3 1 4 1 5	11
7 6 2 7 1 8 2 8	10
11 7 12345 678 90123 45678901 234567 89012 3456 78901 23456 7890	46207983

Задача G. Честный дележ

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

У вас есть массив неотрицательных целых чисел $a_1, a_2, \ldots, a_n$.

Вам нужно разделить его на k непустых подотрезков: $[1; b_1], [b_1 + 1; b_2], \ldots, [b_{k-1} + 1; n]$.

Обозначим сумму на i-м отрезке как s_i и максимум на i-м отрезке как m_i. Ваша задача сделать $|s_i - s_{i+1}| \leqslant \max(m_i, m_{i+1})$ для всех $1 \leqslant i \leqslant k - 1$.

Формат входных данных

В первой строке находятся два целых числа n и k: размер массива и необходимое количество отрезков ($3 \leqslant k \leqslant n \leqslant 100\,000$).

В следующей строке находится n целых чисел $a_1, a_2, \ldots, a_n$: данный массив ($0 \leqslant a_i \leqslant 50\,000$).

Формат выходных данных

Если разделить массив указанным образом возможно, выведите “`Yes`” на первой строке и $k - 1$ целых чисел $b_1, b_2, \ldots, b_{k-1}$, разделенных пробелами на второй строке.

Числа должны удовлетворять $1 \leqslant b_1 < b_2 < \ldots < b_{k-1} < n$.

Также неравенства $|s_i - s_{i+1}| \leqslant \max(m_i, m_{i+1})$ должны быть выполнены для всех $1 \leqslant i \leqslant k - 1$.

Если существует несколько возможных решений, выведите любое.

Если разделение невозможно, выведите “`No`” в единственной строке.

Пример

стандартный ввод	*стандартный вывод*
5 3 17 18 17 30 35	Yes 2 4

Задача H. Тренерский выбор Тимы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Тима с недавних пор начал тренировать баскетбольные команды. Капитаном команды всегда выбирается игрок с максимальным ростом. Также он придумал свою формулу *несовместимости* игроков в одной команде. Формула зависит от роста всех игроков в команде. *Несовместимость* игроков в одной команде равняется сумме разниц роста между всеми игроками и ростом капитана. Более формально, пусть $h_1, h_2, ..., h_m$ это рост игроков команды, $mx = max(h_1, h_2, ..., h_m)$, тогда *несовместимость* $= \sum_{i=1}^{m} mx\text{-}h_i$.

У Тимы есть n игроков выстроенных в ряд, i-й из них имеет рост a_i. Он хочет разбить всех на k команд, каждый игрок должен быть в ровно одной команде и команда должна состоять из игроков, которые составляют непрерывный отрезок в ряду. Тима хочет собрать команды так, чтобы суммарная *несовместимость* была минимальной. Помогите Тиме разбить на команды оптимально.

Формат входных данных

Первая строка входных данных содержит два целых чисел n и k ($1 \leqslant n \leqslant 10^5$, $1 \leqslant k \leqslant min(n, 20)$) — количество игроков в ряду и количество команд. Вторая строка содержит n целых чисел a_i ($1 \leqslant a_i \leqslant 10^6$) — рост i-го игрока слева в ряду в сантиметрах.

Формат выходных данных

Выведите единственное целое число — ответ на задачу.

Система оценки

Данная задача содержит шесть подзадач, в каждой подзадаче выполняются ограничения из условий:

1. $n \leqslant 100$, $k = 1$. Оценивается в 5 баллов.
2. $n \leqslant 2000$. Оценивается в 11 баллов.
3. $k = 2$. Оценивается в 8 баллов.
4. $k = 3$. Оценивается в 15 баллов.
5. $a_i \leqslant a_{i+1}$, для всех $1 \leqslant i < n$. Оценивается в 19 баллов.
6. Ограничения из условия задачи. Оценивается в 42 баллов.

Примеры

стандартный ввод	стандартный вывод
7 3 6 4 1 5 3 2 2	7
5 2 4 1 5 5 6	5
9 2 3 7 4 1 3 2 4 6 7	22

Задача I. Убираем со стола

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мегабайта

После праздников на вашем столе осталось очень много тарелок, которые предстоит убрать. Стол можно представить как полосу из ячеек, которые пронумерованы от левого края стола последовательными целыми числами, начиная с 1.

Каждая тарелка занимает ровно одну ячейку. Всего тарелок ровно n штук, i-я тарелка лежит в ячейке с номером x_i.

У левого края стола находится раковина, куда нужно перенести все тарелки. Будем считать, что раковина находится в ячейке с номером 0. Исходно вы находитесь около раковины.

Для того, чтобы перенести тарелки в раковину, вы можете произвольное количество раз выполнять любое из трех возможных действий. Пусть вы сейчас находитесь около ячейки с номером x. Тогда:

- Можно переместиться от ячейки номер x к произвольной ячейке, затратив на это ровно $|x - y|$ секунд, где y — номер ячейки, к которой вы перемещаетесь.
- Если для некоторого целого k $(0 < k < x)$ во всех ячейках x, $x - 1$, $\ldots$, $x - k + 1$ есть по тарелке, а ячейка $x - k$ — пустая, то можно сдвинуть все тарелки в ячейках от $x - k + 1$ до x включительно на одну ячейку в сторону раковины. В этом случае вы также передвинетесь на одну ячейку ближе к раковине. Такое действие занимает одну секунду. Обратите внимание, что это действие нельзя совершать, если $k = x$, то есть если тарелка с первого стола при таком перемещении упадет в раковину.
- Если вы стоите рядом с тарелкой, то можно отнести ее к раковине, после чего вы будете стоять около раковины в ячейке номер 0. Это действие занимает ровно x секунд.

Так как уборка стола — дело очень утомительное, вы хотите отнести все тарелки в раковину за минимально возможное количество секунд. Определите минимальное время, необходимое для того, чтобы переместить все тарелки в раковину.

Формат входных данных

Первая строка содержит единственное целое число n — количество тарелок на столе $(1 \leqslant n \leqslant 10^6)$.

Следующая строка содержит n целых чисел $x_1, x_2, \ldots, x_n$ — номера ячеек, в которых исходно расположены тарелки $(1 \leqslant x_1 < x_2 < \ldots < x_n \leqslant 10^9)$.

Формат выходных данных

Выведите единственное целое число — минимальное время, за которое можно убрать все тарелки со стола в раковину.

Пример

стандартный ввод	стандартный вывод
5 1 4 5 7 11	34

Замечание

Рассмотрим возможную последовательность действий для первого примера:

1. дойти до тарелки в ячейке 1;
2. отнести тарелку из ячейки 1;
3. дойти до тарелки в ячейке 11;

4. сдвинуть тарелку из ячейки 11 семь раз (до ячейки 4); после этого неубранные тарелки будут занимать ячейки от первой до четвертой включительно;
5. отнести тарелку из ячейки 4 в раковину;
6. дойти до тарелки в ячейке 1 и отнести ее в раковину;
7. дойти до тарелки в ячейке 3;
8. сдвинуть тарелку из ячейки 3 в ячейку 2 (тарелка из ячейки 2 сдвинется в ячейку 1);
9. отнести тарелку из ячейки 2 в раковину;
10. дойти до последней тарелки в ячейке 1 и отнести ее в раковину.

Суммарное затраченное время равно $(1+1)+(11+7+4)+(1+1)+(3+1+2)+(1+1) = 2+22+2+6+2 = 34$.

Задача J. River Land

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Есть страна с большой речной системой. Города в этой стране пронумерованы от 1 до N; город № 1 является столицей. Речная система образует корневое дерево, так что столица является корнем. Каждая река соединяет два города и течет к корню. Для каждого некоренного города i есть река, которая течет из этого города в город p_i. В каждом городе также есть порт.

В стране есть только два способа путешествовать между городами: на лодке и на пароме. Лодка может двигаться только вниз по течению (то есть к столице), в то время как паром может двигаться в обоих направлениях. Есть прямые паромные маршруты из каждого порта в любой другой порт.

Изначально все порты открыты. Иногда некоторые порты могут закрываться или открываться снова (возможно, несколько раз). Порт в столице всегда открыт. Не разрешается путешествовать на пароме из закрытого порта, но допускается путешествие на пароме из открытого порта в закрытый порт. Путешествие на лодке всегда возможно, даже из закрытого порта.

Иногда вы хотите отправиться в путешествие. Вы уже выбрали город, в котором начнете путешествие. Однако у вас достаточно денег только для одной поездки на пароме. Поэтому вы хотите совершить путешествие следующим образом:

- Выбрать город b и отправиться в этот город на лодке. Должна быть возможность путешествовать от v до b на лодке (в частности, $v = b$ разрешено).
- Решить либо продолжить движение из города b на пароме, либо завершить поездку в этом городе.
- Если вы решаете продолжить путь на пароме, вы выбираете город f и отправляетесь на пароме от b до f. Также должна быть возможность путешествовать от b до f на пароме: не разрешается путешествовать вдоль любой реки дважды (независимо от направления), т.е., когда вы решаете продолжить путь на пароме, кратчайшие пути между (v, b) и (b, f) не должны иметь общих рек.
- Вы любите путешествовать на пароме, поэтому чем больше рек вы посещаете во время путешествия на пароме, тем счастливее вы становитесь. Также в каждом городе есть фиксированная красота; давайте обозначим красоту города i как a_i. Затем, для поездки со значением T (для каждой поездки задано своё значение) определим, что вы получаете счастье $a_b + D \cdot T$, где D — длина маршрута парома (количество рек, посещаемых во время путешествия на пароме; возможно, ноль, если продолжить путешествие на пароме из города b невозможно).

Вы должны обработать Q запросов. Есть три типа запросов:

1. - v: порт в городе v закрывается. Гарантируется, что порт в городе v был открыт до этого запроса.
2. + v: порт в городе v вновь открывается. Гарантируется, что порт в городе v был закрыт до этого запроса.
3. ? v T: вы хотите совершить поездку со значением T из города v.

Для каждого запроса третьего типа найдите максимальное счастье, которое вы можете получить от такой поездки.

Формат входных данных

Первая строка содержит два целых числа N и Q ($2 \leqslant N \leqslant 3 \cdot 10^5$, $1 \leqslant Q \leqslant 3 \cdot 10^5$).

Вторая строка содержит $N - 1$ целое число $p_2, p_3, \ldots, p_N$ ($1 \leqslant p_i < i$).

Третья строка содержит N целых чисел $a_1, a_2, \ldots, a_N$ ($1 \leqslant a_i \leqslant 10^9$).

Следующие Q строк описывают запросы в формате, описанном в условии. $1 \leqslant v \leqslant N$, $1 \leqslant T \leqslant 10^9$.

Формат выходных данных

Для каждого запроса третьего типа выведите ответ на него в отдельной строке.

Система оценки

Подзадача	Доп. ограничения	Баллы
1	$N, Q \leqslant 1000$	30
2	$N, Q \leqslant 10^5$	35
3	—	35

Пример

стандартный ввод	стандартный вывод
10 9 1 2 3 2 2 6 3 8 8 30 20 6 13 8 40 7 9 13 1 ? 4 11 - 4 ? 4 11 - 7 ? 10 6 + 7 - 6 - 2 ? 7 4	57 42 33 30

Замечание

В первом запросе вы путешествуеет на пароме из города 4 в город 7.

Во втором запросе вы не можете путешествовать на пароме из города 4, поэтому вы должны использовать второй по выгодности маршрут, который идет от города 2 к городу 7.

В третьем запросе вы едете в город 8 на лодке, а затем в город 7 на пароме.

В четвертом запросе вы едете в столицу (город № 1) на лодке и не путешествуете на пароме.