

Problem A: Четыре магазина

Prob Id: 1

Input file name: input.txt or standard input

Output file name: output.txt or standard output

Real time limit: 1 s

Memory limit: 64M

Тролль Зумбагеш живёт в большом городе Троллеграде. Город разбит на квадратные кварталы, и если посмотреть на карту города, то любая его улица будет проходить либо по горизонтали либо по вертикали. Будем для упрощения называть эти улицы горизонтальными и вертикальными. Для перемещения по Троллеграду можно использовать только эти улицы, движение внутри кварталов запрещено. Любые две перпендикулярные улицы в своем пересечении образуют перекресток, на котором есть магазин.

Для удобства, жители Троллеграда ввели систему координат, где центральная площадь города принята за точку отсчёта. Тогда любой перекресток и магазин на нём будет иметь две координаты x и y .

Сегодня Зумбагеш должен посетить четыре магазина, координаты которых заданы на входе. Он выберет один из них и подъедет туда на авто, после чего по-очереди посетит каждый магазин по одному разу, вернётся к месту стоянки авто и уедет. Ваша задача --- найти длину его пешего пути. Так как тролли не любят ходить пешком, нужно найти минимальный по длине такой маршрут.

Input format

На вход подаются четыре пары чисел --- координаты магазинов, которые должен посетить Зумбагеш. Каждая пара координат содержится в своей строке. Внутри строки два числа разделены пробелом. Все координаты по абсолютной величине не превосходят 10^9 .

Output format

Вывести одно целое число --- минимальную длину маршрута, который начинается и заканчивается возле одного и того же магазина, посещает три остальных и всегда проходит только по горизонтальным и вертикальным улицам. Магазины можно посещать в любом порядке.

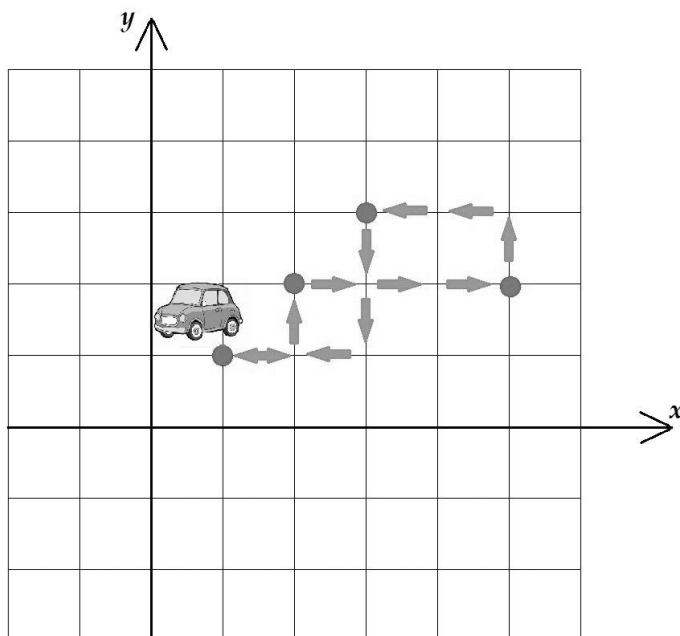
Examples

Input

```
1 1
2 2
5 2
3 3
```

Output

```
12
```



Notes

На рисунке можно видеть один из вариантов кратчайшего маршрута для примера из условия, который начинается в точке (1, 1), проходит через три другие точки и возвращается обратно в точку (1, 1). Его длина 12 кварталов.

Задача проверяется на 20 тестах. Каждый пройденный тест оценивается независимо от других в 5 баллов. Первый тест не совпадает с тестом из условия.

Problem B: Подвиг Магистра

Prob Id: 2

Input file name: input.txt or standard input

Output file name: output.txt or standard output

Real time limit: 2 s

Memory limit: 64M

"Едва мы оправились от страха, как раздалась душераздирающие вопли. Я сразу догадался, что то был воинственный клич какого-то дикого племени. И не ошибся.

Только мы успели, уцепившись за лиану, взобраться на дерево, как под нами появилась огромная толпа дикарей. Одни размахивали копьями, другие потрясали бумерангами.

Несмотря на неудобное положение, я всё же успел сосчитать, сколько воинов окружало нас. Математика прежде всего! Оказалось, что копьеносцев было больше, чем бумерангистов. При этом больше ВО столько раз, НА сколько тех же копьеносцев было больше, чем бумерангистов.

Удивительное совпадение! И ВО сколько раз, и НА сколько — одно и то же число! А число было такое огромное (к сожалению, от страха я забыл его начисто!), что пришлось нам с Единичкой сдаваться в плен. Нас связали и повели к вождю..."

—В.А. Лёвшин, Магистр рассеянных наук

Зумбагеш решил повторить незабываемый подвиг великого Магистра. Убегая от дикарей, он залез на дерево и стал считать копьеносцев и бумерангистов. Как и следовало ожидать, копьеносцев оказалось строго больше чем бумерангистов. Однако, когда Зумбагеш подсчитал НА сколько их больше, то получил число в k раз большее, чем когда он подсчитал ВО сколько раз их было больше. Более точно: пусть копьеносцев больше чем бумерангистов НА x человек и больше В y раз. Тогда $x = ky$.

Вам дано целое число k и гарантируется, что отношение y --- тоже целое. Нужно найти все возможные варианты числа копьеносцев и числа бумерангистов, которые могли быть под деревом.

Input format

В единственной строке входа содержится одно целое число k . $1 \leq k \leq 10^9$.

Output format

В первой строке вывести количество n вариантов ответа, найденных вашим решением. В следующих n строках вывести по два числа a и b через пробел --- возможные количества копьеносцев и бумерангистов соответственно, таких, что a делится нацело на b , и если мы подсчитаем разность $x = a - b$ и частное $y = a / b$, то будет верно, что $x / y = k$. Первое число в строке должно быть числом копьеносцев a , второе числом бумерангистов b . Варианты пар чисел a и b можно выводить в любом порядке.

Examples

Input

5

Output

2

20 10

36 6

Input

1

Output

1

4 2

Notes

Для первого теста из условия можно показать, что есть два способа получить $k = 5$.

Если копьеносцев будет 20, а бумерангистов 10. Тогда разность $20 - 10 = 10$, а частное $20 / 10 = 2$. Видно, что разность в пять раз больше частного.

Для чисел 36 и 6 это свойство так же верно: $36 - 6 = 30$, $36 / 6 = 6$. И снова разность в пять раз больше частного.

Система оценки: За каждый правильный вариант ответа внутри одного теста, ваша попытка получит 1 балл. Таким образом, например, за тест, в котором имеется три правильных варианта ответа, ваша попытка получит три балла. Тесты сгенерированы таким образом, что за полное решение всех тестов суммарно можно набрать ровно 100 баллов. Каждый тест оценивается независимо от других. Примеры из условия не встречается среди основных тестов для этой задачи.

Если ваша попытка выводит количество ответов, большее чем должно быть в полном наборе ответов на этот тест, она получит 0 баллов. Повторяющиеся варианты ответов (при условии, что их суммарное количество не превосходит полного ответа) учитываются по одному разу.

Участник может видеть по данной задаче только полученные баллы. Протокол проверки не предоставляется.

Problem C: Цветомузыка

Prob Id: 3

Input file name: `input.txt` or standard input

Output file name: `output.txt` or standard output

Real time limit: 1 s

Memory limit: 64M

Зумбагеш --- профессионал цветомузыки и его услуги пользуются большим спросом. Сегодня у него важное мероприятие в цирке. Он должен обеспечить красивое световое шоу. Более конкретно --- по кругу арены будет расположено k разноцветных ламп, которые должны будут по-очереди загораться и тут же гаснуть во время представления. Каждая лампа имеет свой цвет, цвета ламп могут повторяться, все доступные цвета обозначены малыми латинскими буквами от **a** до **z**.

Для управления лампами нужно подготовить программу, которая имеет вид строки из n букв. Каждая буква --- это команда зажечься определенной лампе. Более точно: первая буква зажигает первую лампу, вторая буква --- вторую лампу, ... k -я буква зажигает k -ю лампу, $k + 1$ -я буква снова зажигает первую лампу, которая к этому времени уже погасла, $k + 2$ -я буква снова зажигает вторую лампу и так далее по кругу.

Сам Зумбагеш не умеет писать программы управления лампами, поэтому где-то по знакомству добыл какую-то случайную программу длины n . Проблема в том, что если цвет лампы не совпадает с пришедшим сигналом, она взрывается. Из этого следует, что в программе все буквы, зажигающие i -ю лампу должны быть одинаковыми и буква на этих позициях должна совпадать с цветом этой лампы.

Поэтому, в первую очередь, ему нужно доработать эту случайную программу. Единственное что он может сделать для этого --- несколько раз потребовать замену. Каждое требование имеет вид **буква 1** -> **буква 2** и заменяет **все** вхождения буквы 1 в программе на букву 2. Каждое такое требование будет стоить Зумбагешу большого

штрафа, при этом величина штрафа не зависит от количества вхождений и, соответственно замен буквы 1 на букву 2 в программе. Таким образом, при исправлении программы, в первую очередь, он хочет минимизировать число таких требований.

Лампы еще не установлены, поэтому, после исполнения требований на замену и окончательного формирования программы, Зумбагеш пойдет в магазин и закупит нужное количество нужных по цвету ламп. Известно, что можно немного сэкономить и здесь, так как лампы цвета **a** стоят дешевле ламп цвета **b**, лампы цвета **b** стоят дешевле ламп цвета **c**, и так далее, самые дорогие --- лампы цвета **z**. Таким образом, если есть выбор, Зумбагеш изменит программу так, чтобы купить как можно более дешевый набор ламп. Но, повторимся, в первую очередь он хочет добиться минимального числа требований на замену.

Вам нужно помочь Зумбагешу в этой непростой ситуации. По числу ламп **k** и исходному набору команд, представляющему собой строку **s** длины **n** нужно найти такую исправленную строку **p**, что она получена из **s** минимальным числом требований на исправление, при этом ни одна лампа не взорвется, и, если есть возможность, потребует наименьших трат на закупку ламп.

Input format

В первой строке содержится число **k** --- количество ламп на арене цирка. $1 \leq k \leq 10^5$.

В следующей строке содержится непустая строка **s**, состоящая из малых букв латиницы. Длина этой строки не превосходит $2 * 10^5$. Эта строка описывает исходную программу, добытую Зумбагешем и которую нужно доработать.

Output format

Вывести исправленную строку **p**, такую, что она получена из **s** минимальным числом требований на исправление, при этом при её исполнении ни одна лампа не взорвется, и, если есть возможность, потребует наименьших трат на закупку ламп.

Examples

Input

```
4
zqxbzzzybzxhgh
```

Output

```
bbxbbbxbbbxbbb
```

Notes

Всего по кругу будет четыре лампы. Для программы из примера нужно сделать следующие требования на исправление:

z -> b

q -> b

h -> b

y -> x, после чего программа примет вид **bbxbbbxbbbxbbb**. На позиции 1, 2, 4 нужно закупить лампы цвета **b**, на позицию 3 --- лампу цвета **x**.

Подзадачи

Подзадача	Номера тестов	Баллы	Ограничения	Необходимые подзадачи
0 (тесты из условия)	1	0		
1	2-17	30	длина строки не превосходит 100	0
2	18-31	70	без дополнительных ограничений	0, 1

Проверка этой задачи проводится по группам тестов. Это значит, что решение должно выдать правильный ответ для каждого теста группы — только в этом случае все баллы за эту подзадачу будут засчитаны проверяемому решению. Если хотя бы на один тест из группы будет дан неверный ответ, баллы, начисляемые проверяемой программе за эту подзадачу, равны 0. Помимо этого, баллы за подзадачу начисляются только в том случае, когда пройдены тесты всех необходимых для нее подзадач.

Problem D: Троллинг-бизнес

Prob Id: 4

Input file name: input.txt or standard input

Output file name: output.txt or standard output

Real time limit: 1 s

Memory limit: 64M

Зумбагеш давно хотел заняться каким-нибудь бизнесом, приносящим прибыль. И вот ему повезло: он стал владельцем картинной галереи. Галерея состоит из n комнат, соединенных между собой. Более точно, есть главный вход, ведущий в первую комнату, из первой комнаты есть дверь, ведущая во вторую, из второй комнаты есть дверь в третью комнату..., из $n-1$ -й комнаты есть дверь, ведущая в n -ю комнату, из n -й комнаты есть выход, ведущий наружу. В i -й комнате может разместиться a_i посетителей.

Тролли очень культурный народ, поэтому к главному входу сразу выстроилась большая очередь. Зумбагеш только и успевает продавать билеты. Его бизнес устроен следующим образом: каждый час открывается ровно одна из дверей. Сначала это входная дверь, через неё в первую комнату входят столько троллей, сколько свободных мест есть на данный момент в первой комнате. При входе каждый тролль платит Зумбагешу одну монету. Через час открывается дверь между первой комнатой и второй, и несколько посетителей через неё переходят во вторую комнату. При этом выполняется условие: посетители переходят либо до тех пор, пока есть свободные места во второй комнате, либо есть посетители в первой комнате. Еще через час открывается дверь в третью комнату и из второй в третью, посетители снова переходят либо до заполнения третьей комнаты, либо пока не окончатся посетители во второй. И так происходит до того момента, пока не откроется выходная дверь из последней n -й комнаты наружу. В этот момент все тролли из n -й комнаты выходят на улицу и требуют вернуть им их деньги. Зумбагеш конечно возвращает. Разница между числом полученных на входе монет и числом монет, которые ему пришлось вернуть на выходе и составляет его прибыль. Далее опять открывается входная дверь и процесс повторяется заново.

Тролли идут в картинную галерею Зумбагеша непрерывным потоком, но через некоторое время он заметил, что число монет, которые он собирает на входе стало совпадать с числом монет, которые ему приходится отдавать на выходе. Из этого он сделал вывод, что его бизнес достиг потолка. Осталось понять, а какое количество монет он все-таки смог привлечь.

Input format

В первой строке входного потока содержится одно целое число n --- число комнат в картинной галерее. $1 \leq n \leq 10^5$.

Во второй строке содержатся n целых чисел a_i через пробел --- количество посетителей, которые могут поместиться в i -й комнате. $1 \leq a_i \leq 10^9$.

Output format

Вывести в ответ одно целое число --- максимальную прибыль Зумбагеша после того, как его бизнес достигнет потолка.

Examples

Input

```
4
4 6 3 2
```

Output

```
7
```

Notes

Рассмотрим как будут развиваться события в примере из условия.

Во время первого цикла войдут 4 посетителя и отдадут 4 монеты. Далее из первой комнаты во вторую перейдет 4 тролля, из второй в третью 3 тролля, из третьей в четвертую 2 тролля, из четвертой наружу выйдут 2 тролля и потребуют вернуть 2 монеты, итого прибыль Зумбагеша составила $4 - 2 = 2$ монеты.

Во время второго цикла войдут 4 посетителя и отдадут 4 монеты. Далее из первой комнаты во вторую перейдет 4 тролля, из второй в третью 2 тролля, из третьей в четвертую 2 тролля, из четвертой наружу выйдут 2 тролля и потребуют вернуть 2 монеты, итого прибыль Зумбагеша составила $4 - 2 = 2$ монеты.

Во время третьего цикла войдут 4 посетителя и отдадут 4 монеты. Далее из первой комнаты во вторую перейдет 3 тролля, из второй в третью 2 тролля, из третьей в четвертую 2 тролля, из четвертой наружу выйдут 2 тролля и потребуют вернуть 2 монеты, итого прибыль Зумбагеша составила $4 - 2 = 2$ монеты.

Во время четвертого цикла войдут 3 посетителя и отдадут 3 монеты. Далее из первой комнаты во вторую перейдет 2 тролля, из второй в третью 2 тролля, из третьей в четвертую 2 тролля, из четвертой наружу выйдут 2 тролля и потребуют вернуть 2 монеты, итого прибыль Зумбагеша составила $3 - 2 = 1$ монета.

Во время пятого и последующих циклов будут входить 2 посетителя и отдавать 2 монеты. Далее из первой комнаты во вторую перейдет 2 тролля, из второй в третью 2 тролля, из третьей в четвертую 2 тролля, из четвертой наружу выйдут 2 тролля и потребуют вернуть 2 монеты, итого прибыль Зумбагеша составит $2 - 2 = 0$ монет.

В итоге, суммарно его прибыль составит $2 + 2 + 2 + 1 = 7$ монет.

Подзадачи

Подзадача	Номера тестов	Баллы	Ограничения	Необходимые подзадачи
0 (тесты из условия)	1	0		
1	2-11	30	$n * \max(a_i) \leq 5 * 10^6$	0
2	12-22	70	без дополнительных ограничений	0, 1

Проверка этой задачи проводится по группам тестов. Это значит, что решение должно выдать правильный ответ для каждого теста группы — только в этом случае все баллы за эту подзадачу будут засчитаны проверяемому решению. Если хотя бы на один тест из группы будет дан неверный ответ, баллы, начисляемые проверяемой программе за эту подзадачу, равны 0. Помимо этого, баллы за подзадачу начисляются только в том случае, когда пройдены тесты всех необходимых для нее подзадач.

Problem E: Выставка собачек

Prob Id: 5

Input file name: input.txt or standard input

Output file name: output.txt or standard output

Real time limit: 2 s

Memory limit: 512M

Зумбагеш помогает организовать выставку собачек, которая называется "Дог-шоу". Для участия в выставке были зарегистрированы n собачек самых разных пород. Организаторы решили разместить их в один ряд слева направо за большой стеклянной стеной на подсвеченной сцене, на которой уже подготовлены n красивых подиумов, по одному для каждой собачки. На сцену можно зайти с двух сторон: слева и справа. Владельцы собачек уже стоят в некоторой очереди в центральном проходе со своими питомцами в ожидании их размещения на сцене.

Про каждую собачку известно, что она любит лаять на других собачек. Более точно, i -я собачка тявкнет a_i раз на любую другую. Число a_i будем называть показателем лаятельных способностей собачки номер i . Организаторы приняли решение, что собачки будут расположены на сцене слева направо в порядке неубывания их лаятельных способностей, то есть, например, если собачка Белла тявкает 3 раза, а собачка Чарли тявкает 10 раз, то Белла будет находиться на сцене левее чем Чарли. Если лаятельные способности двух собачек равны, то они могут быть расположены друг относительно друга в любом порядке.

Теперь Зумбагеш должен организовать размещение этих собачек на сцене. Он будет брать очередную собачку у её владельца и по одному из входов на сцену (левому или правому) будет заносить её за стеклянную стену. Там он будет проносить собачку до её места. Если есть несколько мест, куда можно поместить эту собачку, то Зумбагеш сам может выбрать на какое из этих мест её поместить. При этом собачка в его руках будет лаять на всех собачек, которых Зумбагеш уже ранее разместил на их подиумах и мимо которых в данный момент её проносят, а те будут в ответ также лаять на неё. Само собой, Зумбагеш хочет минимизировать шум в зале, и вносить собачек так, чтобы они суммарно лаяли как можно меньше.

Порядок внесения собачек Зумбагеш изменить уже не может, никакой владелец с этим не согласится. Но Зумбагеш может выбрать с какой стороны заходить на сцену.

По заданным числам n и описанию очереди a_i нужно определить минимально возможное число лая, который придется выслушать Зумбагешу, прежде чем он сможет отчитаться о том, что все собачки размещены на подиумах и "Дог-шоу" можно начинать.

Input format

В первой строке входных данных содержится целое число n --- количество собачек. $1 \leq n \leq 2 \cdot 10^5$.

Во второй строке содержатся n целых чисел a_i через пробел --- лаятельные способности собачек в том порядке, в котором их будут вносить на сцену. $1 \leq a_i \leq 10^9$.

Output format

Вывести в ответ одно целое число --- минимально возможное число лая, который придется выслушать Зумбагешу при размещении всех собачек на их места.

Examples

Input

```
8
3 1 9 4 3 4 5 4
```

Output

```
62
```

Input

```
3
3 1 2
```

Output

```
3
```

Notes

Разберём первый пример из условия. Итоговая расстановка собачек по их лаятельным способностям должна иметь вид 1 3 3 4 4 4 5 9.

Сначала с любого входа заносим собачку номер 1, которая лает 3 раза. Её можно поместить на подиумы номер 2 или номер 3, предусмотрительно помещаем её на подиум номер 3 за 0.

Теперь с левого входа заносим собачку номер 2, которая лает 1 раз. Её помещаем на самый левый подиум номер 1 за 0.

Теперь с правого входа заносим собачку номер 3, которая лает 9 раз. Её помещаем на самый последний подиум номер 8 за 0.

Пришла очередь собачки номер 4, которая лает 4 раза. Её можно разместить на подиумах 4, 5 или 6. Сначала решим с какого входа её заносить. Если занесём справа, то наша собачка пролает 4 раза на одну собачку с подиума номер 8, а та в ответ пролает 9 раз, итого будет 13, а если занесём слева, то наша пролает два раза по 4, и получит в ответ 1 и 3 лая соответственно, итого 12, что лучше. Из дальнейших событий станет ясно, что нашу собачку нужно посадить на подиум номер 5 за 12.

Теперь собачка номер 5, которая лает 3 раза. Заносим слева, при этом она лает 3 раза на собачку с первого подиума, а та в ответ лает 1 раз, итого 4 раза. Садим эту пятую собачку на место номер 2 за 4.

Собачка номер 6, которая лает 4 раза. Если заносить её слева, то придется пронести мимо трёх собачек, что в итоге даст 19 лаев, а если заносить справа, то проносим мимо одной собачки, которая лает 9 раз, что суммарно с нашей даст 13. Таким образом шестую собачку садим на место номер 6 за 13.

Далее собачка номер 7, которая лает 5 раз. Заносим справа, она лает 5 раз на собачку с последнего подиума, а та в ответ лает 9 раз, итого 14 раз. Садим эту седьмую собачку на место номер 7 за 14.

И наконец, собачка номер 8. Заносим слева мимо трёх, которые суммарно лают 7 раз, и наша им в ответ 12, то есть садим её на подиум номер 4 за 19.

Всего пришлось выслушать $0 + 0 + 0 + 12 + 4 + 13 + 14 + 19 = 62$ лая.

Подзадачи

Подзадача	Номера тестов	Баллы	Ограничения	Необходимые подзадачи
0 (тесты из условия)	1-2	0		
1	3-9	15	$n \leq 1000, a_i \leq 1000$	0
2	10-16	15	$n \leq 1000, a_i \leq 10^9$	0, 1
3	17-26	50	$n \leq 2 \cdot 10^5, a_i \leq 2 \cdot 10^5$	0, 1, 2
4	27-36	20	$n \leq 2 \cdot 10^5, a_i \leq 10^9$	0, 1, 2, 3

Проверка этой задачи проводится по группам тестов. Это значит, что решение должно выдать правильный ответ для каждого теста группы — только в этом случае все баллы за эту подзадачу будут засчитаны проверяемому решению. Если хотя бы на один тест из группы будет дан неверный ответ,

баллы, начисляемые проверяемой программе за эту подзадачу, равны 0. Помимо этого, баллы за подзадачу начисляются только в том случае, когда пройдены тесты всех необходимых для нее подзадач.

Problem F: 3D-печать торта

Prob Id: 6

Input file name: `input.txt` or standard input

Output file name: `output.txt` or standard output

Real time limit: 1 s

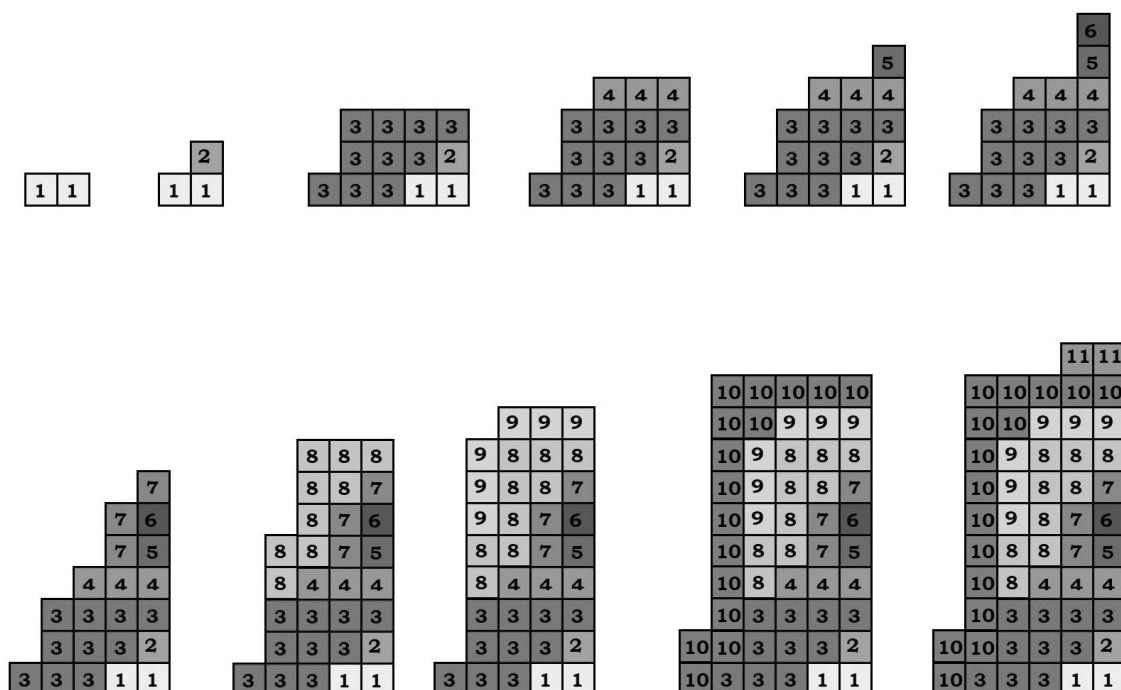
Memory limit: 64M

Зумбагеш умеет печатать торты на 3D-принтере. Это очень просто: заряжаешь в пищевой 3D-принтер n разноцветных слоев теста и нажимаешь кнопку.

Далее принтер работает следующим образом: он начинает печатать каждый слой справа налево поверх предыдущих слоев. Первый слой непосредственно на подложке, второй поверх первого справа налево, третий поверх второго справа налево, и т. д. Если бы для каждого слоя было заряжено одинаковое количество теста, то получился бы красивый ровный торт.

Однако Зумбагеш был слегка невнимателен при зарядке принтера, и зарядил разные по объему слои. Более точно, теста первого цвета он загрузил a_1 единиц, теста второго цвета он загрузил a_2 единиц, ... теста n -го цвета он загрузил a_n единиц.

В этих условиях, принтер работает следующим образом: очередной слой он начинает печатать поверх предыдущего и, если текущего теста больше чем ширина предыдущего слоя, он смещается на одну позицию левее и заполняет все в этой позиции до текущей высоты торта. Если и после этого текущее тесто еще не закончилось, происходит смещение еще на одну позицию левее и снова заполняется столбик до текущей высоты, и так далее. В какой-то момент текущее тесто заканчивается и принтер переходит к печати следующего слоя следующим тестом. Так он работает, пока в нем не закончится заряженное тесто.



Для лучшего понимания процесса, изучите рисунок. На нем показан процесс изготовления торта для примера из условия. Одинаковые слои обозначены одинаковым цветом и числом. Процесс показан пошагово, каждый отдельный фрагмент рисунка показывает вид торта после добавления очередного слоя.

Зумбагешу такой торт понравился даже больше обычного, но перед тем, как его подарить другу на день рождения, он решил запомнить для каждого слоя его ширину. По заданным числам $a_1, a_2, \dots, a_n$ нужно определить ширину каждого слоя полученного таким образом торта, слои перечисляются снизу вверх в порядке убывания.

Input format

В первой строке входных данных содержится одно целое число n --- количество видов теста, которое Зумбагеш зарядил в принтер. $1 \leq n \leq 10^5$.

Во второй строке содержатся через пробел n целых чисел a_i --- объемы слоев теста загруженных в принтер. Слои указаны в порядке их загрузки. $1 \leq a_i \leq 2 \cdot 10^9$.

Output format

Вывести n чисел в одну строку через пробел --- ширину каждого слоя полученного торта. Первое число указывает ширину первого слоя снизу, второе число --- ширину второго слоя снизу, и т. д. до n -го числа, указывающего ширину самого верхнего слоя.

Examples

Input

```
11
2 1 10 3 1 1 3 9 6 16 2
```

Output

```
6 6 5 5 5 5 5 5 5 2
```

Notes

Подзадачи

Подзадача	Номера тестов	Баллы	Ограничения	Необходимые подзадачи
0 (тесты из условия)	1	0		
1	2-8	10	$n \leq 50$	0

2	9-21	90	без дополнительных ограничений	0, 1
---	------	----	--------------------------------	------

Проверка этой задачи проводится по группам тестов. Это значит, что решение должно выдать правильный ответ для каждого теста группы — только в этом случае все баллы за эту подзадачу будут засчитаны проверяемому решению. Если хотя бы на один тест из группы будет дан неверный ответ, баллы, начисляемые проверяемой программе за эту подзадачу, равны 0. Помимо этого, баллы за подзадачу начисляются только в том случае, когда пройдены тесты всех необходимых для нее подзадач.
