

Задача А. Додайте їх усі!

Назва вхідного файлу: *standard input*
 Назва вихідного файлу: *standard output*
 Ліміт часу: 1 second
 Ліміт використання пам'яті: 1024 mebibytes

Вам дано дві послідовності додатних цілих чисел довжини n : $(a_1, a_2, \dots, a_n)$ та $(b_1, b_2, \dots, b_n)$.
 Для $k = 2, 3, \dots, 2n$, обчисліть значення

$$\sum_{i+j \leq k} (a_i + b_j),$$

тобто, суму $(a_i + b_j)$ для всіх пар індексів (i, j) таких, що $i + j \leq k$ та $1 \leq i, j \leq n$.

Формат вхідних даних

Перший рядок вхідних даних містить одне ціле число n ($1 \leq n \leq 2 \cdot 10^5$).

Другий рядок містить n цілих чисел: елементи послідовності a , у порядку ($1 \leq a_i \leq 10^6$).

Третій рядок містить n цілих чисел: елементи послідовності b , у порядку ($1 \leq b_i \leq 10^6$).

Формат вихідних даних

Виведіть $2n - 1$ рядків. У i -му рядку виведіть відповідь для випадку, коли $k = i + 1$.

Приклади

<i>standard input</i>	<i>standard output</i>
3 2 2 2 2 2 2	4 12 24 32 36
5 4 8 2 9 4 8 11 6 4 5	12 43 82 134 195 244 274 296 305
1 2 4	6

Задача В. Дужки та Реверсований рядок

Назва вхідного файлу: *standard input*
 Назва вихідного файлу: *standard output*
 Ліміт часу: 1 second
 Ліміт використання пам'яті: 1024 mebibytes

Правильна дужкова послідовність — це рядок, який задовольняє будь-яку з наступних умов:

- Це порожній рядок.
- Він утворений шляхом конкатенації $(, A,)$ у цьому порядку, де A — це правильна дужкова послідовність.
- Він утворений шляхом конкатенації A та B , де A та B — це правильні дужкові послідовності.

Вам дано рядок s довжини n , що складається з символів $($ та $)$.

Для кожного i , де $0 \leq i \leq n$, визначимо t_i як рядок, отриманий шляхом конкатенації суфікса s довжини $n - i$ та реверснутото префікса s довжини i , у цьому порядку. Тобто, якщо ми позначимо i -ий символ рядка s як s_i , рядок t_i утворюється шляхом розташування символів $s_{i+1}, s_{i+2}, \dots, s_n, s_i, \dots, s_2, s_1$ у послідовності.

Для кожного t_i , де $0 \leq i \leq n$, розв'яжіть наступну задачу. Розгляньте операцію, в якій ви замінюєте один символ у t_i на $($ або $)$. Знайдіть мінімальну кількість таких операцій, необхідних для того, щоб зробити t_i правильною дужковою послідовністю.

Формат вхідних даних

Перший рядок вхідних даних містить одне ціле число n ($2 \leq n \leq 2 \cdot 10^5$, n парне).

Другий рядок містить рядок s довжини n , що складається лише з $($ та $)$.

Формат вихідних даних

Виведіть $n + 1$ рядків. У рядку $i + 1$ виведіть відповідь для t_i .

Приклади

<i>standard input</i>	<i>standard output</i>
8))))((((	4 4 2 2 0 0 0 0 0
6))())(	3 1 1 1 1 1 1
4 (())	0 2 2 2 2

Задача С. Кастомні банкноти

Назва вхідного файлу: *standard input*
Назва вихідного файлу: *standard output*
Ліміт часу: 2 seconds
Ліміт використання пам'яті: 1024 mebibytes

У Королівстві Байтеланд до цього часу випущено лише банкноти номіналом 1 долар. Однак, через збільшення обігу банкнот, королівство вирішило повністю оновити свою систему банкнот. Нова система банкнот представлена послідовністю додатних цілих чисел $x = (x_1, x_2, \dots, x_k)$. Це означає, що нова система використовує k типів банкнот з номіналами $x_1, x_2, \dots, x_k$ доларів. Ви можете визначити кількість типів банкнот k та їх значення $x_1, x_2, \dots, x_k$ за такими обмеженнями:

- k — додатне ціле число.
- $1 = x_1 < x_2 < \dots < x_k$.
- x_{i+1} має бути кратним x_i ($1 \leq i \leq k-1$).

У Королівстві Байтеланд товари часто продаються за цінами a , b або c доларів. Тому незручність нової системи банкнот визначається як:

(Мінімальна кількість банкнот, необхідних для представлення a доларів) +
(Мінімальна кількість банкнот, необхідних для представлення b доларів) +
(Мінімальна кількість банкнот, необхідних для представлення c доларів).

Ваше завдання — знайти мінімально можливе значення цієї незручності.

Формат вхідних даних

Перший рядок вхідних даних містить три цілі числа: a , b та c ($1 \leq a < b < c \leq 10^8$).

Формат вихідних даних

Виведіть одне ціле число: мінімально можливе значення незручності для нової системи банкнот.

Приклади

<i>standard input</i>	<i>standard output</i>
3 12 17	6
99999979 99999981 99999998	10

Задача D. Робота з послідовністю

Назва вхідного файлу: *standard input*
 Назва вихідного файлу: *standard output*
 Ліміт часу: 2 second
 Ліміт використання пам'яті: 1024 mebibytes

Розглянемо послідовність цілих чисел $b = (b_1, b_2, \dots, b_{2^n-1})$ довжини $2^n - 1$.

Нехай $f(b)$ — це мінімальна кількість операцій, необхідних для того, щоб виконати наступну умову:

- Операція: Вибрати одне ціле число i таке, що $1 \leq i \leq 2^n - 1$, і або збільшити, або зменшити b_i на 1.
- Умова: Для всіх i , де $1 \leq i \leq 2^{n-1} - 1$, повинна виконуватись рівність $b_i = b_{2i} + b_{2i+1}$.

Вам дана послідовність $a = (a_1, a_2, \dots, a_{2^n-1})$ довжини $2^n - 1$. Обробіть q запитів. Для кожного запиту i (де $1 \leq i \leq q$): дано цілі числа x_i та v_i , зробіть так, щоб a_{x_i} дорівнювало v_i , а потім виведіть $f(a)$.

Зміни в послідовності зберігаються між запитами. Наприклад, другий запит стосується послідовності, зміненої $a_{x_1} \leftarrow v_1$, а потім $a_{x_2} \leftarrow v_2$.

Формат вхідних даних

Перший рядок вхідних даних містить одне ціле число n ($2 \leq n \leq 18$).

Другий рядок містить $2^n - 1$ цілих чисел: послідовність a ($-10^9 \leq a_i \leq 10^9$).

Третій рядок містить одне ціле число q : кількість запитів ($1 \leq q \leq 10^5$).

Кожен з наступних q рядків містить два цілі числа x_i та v_i : параметри i -го запиту ($1 \leq x_i \leq 2^n - 1$, $-10^9 \leq v_i \leq 10^9$).

Формат вихідних даних

Виведіть q рядків. У i -му рядку виведіть відповідь для i -го запиту.

Приклад

<i>standard input</i>	<i>standard output</i>
3	5
2 0 3 1 -6 3 1	5
5	6
5 2	9
3 3	9
6 -1	
5 -7	
2 0	

Задача Е. Генерація виразів

Назва вхідного файлу: *standard input*
 Назва вихідного файлу: *standard output*
 Ліміт часу: 1 second
 Ліміт використання пам'яті: 1024 mebibytes

Вам дано рядок s . Кожен символ у s є одним з “0123456789+()?” (без лапок).

Нехай t буде рядком, сформованим шляхом заміни кожного ‘?’ у s на один з “0123456789+()”.

Визначимо $eval(t)$ наступним чином:

- Якщо t є правильним математичним виразом, то це значення, отримане шляхом обчислення t як виразу.
- Якщо t не є правильним математичним виразом, то це 0.

Обчисліть суму $eval(t)$ для всіх можливих способів замінити кожен ‘?’ у s на один з “0123456789+()”, і виведіть результат за модулем 998 244 353.

Правильний математичний вираз визначається наступною нотацією Бекуса - Наура:

```
<expression> ::= <expression> + <primary> | <primary>
<primary> ::= ( <expression> ) | <number>
<number> ::= <nonzero-digit> <number-sub> | <digit>
<number-sub> ::= <number-sub> <digit> | <digit>
<digit> ::= 0 | <nonzero-digit>
<nonzero-digit> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Формат вхідних даних

Вхідні дані містять єдиний рядок s ($1 \leq |s| \leq 3000$, кожен символ s є одним з “0123456789+()?”)

Формат вихідних даних

Виведіть єдине ціле число: відповідь на задачу.

Приклади

<i>standard input</i>	<i>standard output</i>
?3?	48108
40???0+3??	726820366

Задача F. Інвертування

Назва вхідного файлу:	<i>standard input</i>
Назва вихідного файлу:	<i>standard output</i>
Ліміт часу:	1 second
Ліміт використання пам'яті:	1024 mebibytes

Вам дано кореневе дерево з n вершин, де вершина 1 є коренем. Батьком вершини i ($2 \leq i \leq n$) є вершина p_i . Кожна вершина має значення 0 або 1, написане на ній, і спочатку вершина i ($1 \leq i \leq n$) має значення a_i .

Визначимо операцію *інвертування вершини* x_i : якщо значення, написане на вершині x_i , дорівнює 0, змініть його на 1; якщо воно 1, змініть його на 0.

Визначимо операцію *інвертування шляху* для вершини i : виконайте інвертування вершини для кожної вершини на шляху від кореня (вершина 1) до вершини i (включно).

Вам потрібно обробити q запитів. i -й запит ($1 \leq i \leq q$) має наступний вигляд:

- Спочатку виконайте інвертування вершини x_i .
- Після цього знайдіть і виведіть мінімальну кількість операцій інвертування **шляху**, необхідних для того, щоб усі вершини мали значення 0.

Можна довести, що всі нулі завжди можна досягти за скінченну кількість операцій інвертування шляху.

Зміни зберігаються між запитами. Наприклад, другий запит стосується дерева після виконання інвертування вершини x_1 і потім виконання інвертування вершини x_2 .

Формат вхідних даних

Перший рядок вхідних даних містить одне ціле число n ($2 \leq n \leq 2 \cdot 10^5$).

Другий рядок містить $n - 1$ цілих чисел p_i . i -те ціле число є батьківською вершиною для вершини $i + 1$ ($1 \leq p_i \leq i$).

Третій рядок містить n цілих чисел a_i : початкові значення, написані на вершинах ($0 \leq a_i \leq 1$).

Четвертий рядок містить одне ціле число q ($1 \leq q \leq 2 \cdot 10^5$).

Кожен з наступних q рядків містить одне ціле число x_i : параметр i -го запиту ($1 \leq x_i \leq n$).

Формат вихідних даних

Виведіть q рядків. У i -му рядку виведіть відповідь на i -й запит.

Приклад

<i>standard input</i>	<i>standard output</i>
4	2
1 2 1	3
0 1 1 0	3
3	
4	
1	
2	

Задача G. Генерація трикутників

Назва вхідного файлу: *standard input*
Назва вихідного файлу: *standard output*
Ліміт часу: 1 second
Ліміт використання пам'яті: 1024 mebibytes

У вас є a_1 паличок довжини 1, a_2 паличок довжини 2 та a_3 паличок довжини 3. Ви можете виконувати наступну операцію будь-яку кількість разів: вибрати 3 палички так, щоб вони могли утворити трикутник з ненульовою площею, і використати ці 3 палички для створення трикутника. Після використання палички не можуть бути використані для утворення інших трикутників.

Визначте максимальну кількість трикутників, які можна створити. Дано t тестових випадків, обчисліть відповідь для кожного.

Формат вхідних даних

Перший рядок вхідних даних містить одне ціле число t , кількість тестових випадків ($1 \leq t \leq 10^4$).

Кожен з наступних t рядків позначає один тестовий випадок і містить три цілі числа: a_1 , a_2 та a_3 ($0 \leq a_i \leq 10^8$).

Формат вихідних даних

Для кожного тестового випадку виведіть одне ціле число: максимальна кількість трикутників, які можна створити.

Приклад

<i>standard input</i>	<i>standard output</i>
4	2
4 1 2	0
0 0 0	2
3 2 2	476190
142857 428571 857142	

Задача Н. Розфарбування півплощин

Назва вхідного файлу:	<i>standard input</i>
Назва вихідного файлу:	<i>standard output</i>
Ліміт часу:	1 second
Ліміт використання пам'яті:	1024 mebibytes

У вас є 2D площина, яка спочатку повністю біла. Ви можете виконувати наступну операцію будь-яку кількість разів: вибрати пряму та півплощину, обмежену цією прямою. Потім виконайте одну з двох наступних дій:

- Розфарбуйте півплощину в чорний колір, **виключаючи** межу.
- Розфарбуйте півплощину в білий колір, **включаючи** межу.

Вам дано багатокутник P з n вершинами, який не обов'язково є опуклим. Вершини P дано в порядку проти годинникової стрілки як $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, а i -ий край P з'єднує вершину (x_i, y_i) з вершиною $(x_{(i \bmod n)+1}, y_{(i \bmod n)+1})$.

Визначте, чи можливо використати вищезгадані операції, щоб розфарбувати лише внутрішню частину багатокутника P в чорний колір, залишаючи все інше білим.

Формат вхідних даних

Перший рядок вхідних даних містить одне ціле число n ($3 \leq n \leq 4000$).

Далі йдуть n рядків, кожен з яких містить два цілі числа x_i та y_i : координати i -ї вершини багатокутника ($-10^7 \leq x_i, y_i \leq 10^7$).

Ви можете припустити, що всі вершини різні, краї багатокутника P не мають спільних точок, окрім вершин, і кожен внутрішній кут багатокутника P не дорівнює 180 градусів.

Формат вихідних даних

Якщо можливо досягти бажаного стану за допомогою операцій, виведіть "Yes"; в іншому випадку виведіть "No".

Приклади

<i>standard input</i>	<i>standard output</i>
4 11 -4 3 -4 -6 7 -6 -7	Yes
12 16 0 18 2 11 9 18 16 16 18 9 11 2 18 0 16 7 9 0 2 2 0 9 7	No

Задача І. Цілі числа на краях

Назва вхідного файлу:	<i>standard input</i>
Назва вихідного файлу:	<i>standard output</i>
Ліміт часу:	2 second
Ліміт використання пам'яті:	1024 mebibytes

Вам дано ціле число $n \geq 2$. Побудуйте набір точок (i, j) на 2D площині такий, що $1 \leq i \leq n$ та $1 \leq j \leq i$. Розглянемо неорієнтований зважений граф, де ці точки є вершинами, та є наступні ребра:

- Між (i, j) та $(i + 1, j)$ існує ребро з вагою $a_{i,j}$ (для $1 \leq i \leq n - 1$ та $1 \leq j \leq i$).
- Між (i, j) та $(i + 1, j + 1)$ існує ребро з вагою $b_{i,j}$ (для $1 \leq i \leq n - 1$ та $1 \leq j \leq i$).
- Між (i, j) та $(i, j + 1)$ існує ребро з вагою $c_{i+1,j}$ (для $1 \leq i \leq n - 1$ та $1 \leq j \leq i - 1$).

Для довідки, граф має $n(n + 1)/2$ вершин та $3n(n - 1)/2$ ребер.

Для простого шляху між вершинами в цьому графі вага шляху визначається як **добуток** ваг ребер, через які проходить шлях.

Визначте кількість неупорядкованих пар різних вершин $\{s, t\}$, такі що кожен простий шлях між s та t має вагу, яка є повним квадратом.

Формат вхідних даних

Перший рядок вхідних даних містить одне ціле число n ($2 \leq n \leq 1000$).

Далі йдуть $n - 1$ рядків. i -ий з цих рядків містить ваги $a_{i,j}$ для $1 \leq j \leq i$ ($1 \leq a_{i,j} \leq 10^6$).

Далі йдуть ще $n - 1$ рядків. i -ий з цих рядків містить ваги $b_{i,j}$ для $1 \leq j \leq i$ ($1 \leq b_{i,j} \leq 10^6$).

Далі йдуть ще $n - 1$ рядків. i -ий з них містить ваги $c_{i+1,j}$ для $1 \leq j \leq i$ ($1 \leq c_{i+1,j} \leq 10^6$).

Формат вихідних даних

Виведіть єдине ціле число: відповідь на задачу.

Приклади

<i>standard input</i>	<i>standard output</i>
2 1 3 3	1
3 3 1 4 1 5 9 2 6 5	0

Задача J. Джунглі та Піраміда

Назва вхідного файлу:	<i>standard input</i>
Назва вихідного файлу:	<i>standard output</i>
Ліміт часу:	4 seconds
Ліміт використання пам'яті:	1024 mebibytes

Аліса, яка досліджувала містичні будівлі в джунглях у пошуках скарбу, знайшла піраміду, побудовану майянськими священиками. Аліса увійшла в неї і опинилася в коридорі, де входи до n кімнат були розташовані в ряд. Після розслідування вона виявила, що кімнати пронумеровані унікально від 1 до n , але точний номер кожної кімнати Алісі невідомий, поки вона не увійшла. Нехай i -та кімната зліва має номер p_i . Вона також виявила, що скарб був захований у кімнаті k .

З огляду на її залишкову витривалість, Алісі було важко перевірити всі кімнати. Однак Аліса мала секретну стратегію, щоб подолати цю ситуацію: бінарний пошук. Аліса успішно застосовувала бінарний пошук до різних завдань раніше. З останньою краплею сил вона вирішила використати бінарний пошук, щоб знайти кімнату k .

Конкретно, Аліса діє наступним чином. Спочатку вона ініціалізує змінні ℓ та r з $\ell = 0$ та $r = n + 1$. Потім вона повторює кроки 1 до 3 наступним чином:

1. Якщо $\ell + 1 = r$, зупиняємо пошук, оскільки вона не знайшла кімнату k .
2. Встановимо $m = \lfloor (\ell + r) / 2 \rfloor$.
Увійдемо у кімнату, що розташована на m -му місці зліва та перевіримо її номер.
3. Якщо $p_m = k$, зупинимо пошук, оскільки вона знайшла кімнату k .
Якщо $p_m < k$, оновимо ℓ до m .
Якщо $p_m > k$, оновимо r до m .

Існує $n!$ можливих перестановок номерів кімнат $(p_1, p_2, \dots, p_n)$. Вам потрібно визначити кількість перестановок, для яких Аліса може успішно знайти кімнату k за допомогою наведеного вище процесу, за модулем 998 244 353.

Дано t тестових випадків, обчисліть відповідь для кожного.

Формат вхідних даних

Перший рядок вхідних даних містить єдине ціле число t , кількість тестових випадків ($1 \leq t \leq 10^5$). Потім йдуть t тестових випадків.

Кожен тестовий випадок складається з одного рядка, що містить два цілі числа n та k ($3 \leq n \leq 10^6$, $1 \leq k \leq n$).

Формат вихідних даних

Для кожного тестового випадку виведіть єдине ціле число: відповідь на задачу.

Приклад

<i>standard input</i>	<i>standard output</i>
5	12
4 1	4
3 2	348
6 4	12337920
11 5	918345168
1000000 500000	

Задача К. Король, Джокер та Подільність

Назва вхідного файлу: *standard input*
Назва вихідного файлу: *standard output*
Ліміт часу: 3 seconds
Ліміт використання пам'яті: 1024 mebibytes

Король просить Джокера відповісти на q запитів наступного вигляду:

Дано три цілі числа: ℓ_i , r_i та k_i . Знайдіть k_i -те найменше додатне ціле число x , таке що x не ділить жодне з цілих чисел від ℓ_i до r_i (включно).

Формат вхідних даних

Перший рядок вхідних даних містить одне ціле число q ($1 \leq q \leq 10^5$).

Кожен з наступних q рядків містить запит, що складається з трьох цілих чисел: ℓ_i , r_i та k_i ($1 \leq \ell_i \leq r_i \leq 2 \cdot 10^5$, $1 \leq k_i \leq 2 \cdot 10^5$).

Формат вихідних даних

Для кожного запиту виведіть одне ціле число: значення x для цього запиту.

Приклад

<i>standard input</i>	<i>standard output</i>
3	13
20 22 5	32204
12354 31415 789	200000
2 100000 100000	

Задача L. Давайте виміряємо перестановки

Назва вхідного файлу: *standard input*
Назва вихідного файлу: *standard output*
Ліміт часу: 1 second
Ліміт використання пам'яті: 1024 mebibytes

Розглянемо перестановку p цілих чисел $1, 2, \dots, n$. Наскільки вона близька до перестановки $(1, 2, \dots, n)$?

Ліза вимірює це як кількість інверсій у p : кількість пар (i, j) таких, що $i < j$ і $p_i > p_j$.

Лео, з іншого боку, використовує метрику $\frac{1}{2} \sum_{i=1}^n |p_i - i|$.

Дано послідовність $a = (a_1, a_2, \dots, a_k)$ з k різних цілих чисел від 1 до n , існує $(n-k)!$ перестановок довжини n , які мають a як свій префікс.

Знайдіть кількість цих перестановок, для яких метрика Лізи та метрика Лео є рівними, і поверніть результат за модулем 998 244 353.

Формат вхідних даних

Перший рядок вхідних даних містить два цілі числа n та k ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq k \leq n$).

Якщо k додатне, другий рядок містить k парно різних цілих чисел a_i : префікс перестановки ($1 \leq a_i \leq n$, $a_i \neq a_j$ для $i \neq j$).

Формат вихідних даних

Виведіть одне ціле число: кількість таких перестановок за модулем 998 244 353.

Приклади

<i>standard input</i>	<i>standard output</i>
6 3 3 4 6	1
11 11 10 3 9 1 2 4 7 5 6 8 11	0
5 0	42

Задача М. Магентові та Цианові Цілі Числа

Назва вхідного файлу: *standard input*
 Назва вихідного файлу: *standard output*
 Ліміт часу: 2 seconds
 Ліміт використання пам'яті: 1024 mebibytes

Для послідовності додатних цілих чисел $x = (x_1, x_2, \dots, x_k)$, де $k \geq 2$, нехай $f(x)$ буде відповіддю на таку задачу:

Серед k додатних цілих чисел $x_1, x_2, \dots, x_k$ пофарбуйте принаймні одне і не більше ніж $k-1$ з них у магенту, а решту - у циановий. Нехай d_m буде найбільшим спільним дільником (НСД) цілих чисел, пофарбованих у магенту, а d_c - НСД цілих чисел, пофарбованих у циановий. Знайдіть максимальне можливе значення $d_m + d_c$.

Вам дана послідовність з n додатних цілих чисел $a = (a_1, a_2, \dots, a_n)$. Вам також буде дано q запитів. Для кожного запиту вам будуть дані два цілі числа ℓ_i та r_i , такі що $1 \leq \ell_i < r_i \leq n$. Для кожного запиту нехай $x = (a_{\ell_i}, a_{\ell_i+1}, \dots, a_{r_i})$, і знайдіть $f(x)$.

Формат вхідних даних

Перший рядок вхідних даних містить одне ціле число n ($2 \leq n \leq 2 \cdot 10^5$).

Другий рядок містить n цілих чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^{18}$).

Третій рядок містить одне ціле число q ($1 \leq q \leq 2 \cdot 10^5$).

Кожен з наступних q рядків описує один запит і містить два цілі числа: ℓ_i та r_i ($1 \leq \ell_i < r_i \leq n$).

Формат вихідних даних

Для кожного запиту виведіть одне ціле число: значення $f(x)$.

Приклад

<i>standard input</i>	<i>standard output</i>
6	6
2 5 1 4 3 3	5
3	6
1 4	
3 6	
1 6	

Задача N. Гарні підпоследовності

Назва вхідного файлу: *standard input*
 Назва вихідного файлу: *standard output*
 Ліміт часу: 1 second
 Ліміт використання пам'яті: 1024 mebibytes

Дано послідовність $a = (a_1, a_2, \dots, a_n)$, *гарна* підпоследовність a — це підпоследовність (не обов'язково неперервна), де сусідні елементи в підпоследовності не є взаємно простими.

Знайдіть максимальну довжину ℓ гарної підпоследовності a . Також визначте кількість гарних підпоследовностей довжини ℓ , за модулем 998 244 353.

Формат вхідних даних

Перший рядок вхідних даних містить одне ціле число n ($2 \leq n \leq 2 \cdot 10^5$).

Другий рядок містить послідовність a і складається з n цілих чисел a_i ($1 \leq a_i \leq 10^6$).

Формат вихідних даних

Виведіть два рядки. У першому рядку виведіть ℓ . У другому рядку виведіть кількість гарних підпоследовностей довжини ℓ з a , за модулем 998 244 353.

Приклади

<i>standard input</i>	
3	
3 5 15	
<i>standard output</i>	
2	
2	
<i>standard input</i>	
7	
1 1 1 1 1 1 1	
<i>standard output</i>	
1	
7	