Задача A. Золоте поле

Назва вхідного файлу:	`a.in` або стандартний ввід
Назва вихідного файлу:	`a.out` або стандартний вивід
Обмеження використання часу:	1 секунда
Обмеження використання пам'яті:	256 мегабайтів

Козак Вус випадково знайшов прямокутне поле розміром $n \times m$ квадратних метрів. Поле має n рядків та m стовпчиків. Рядки нумеруються від 1 до n зверху вниз. Стовпці нумеруються від 1 до m зліва направо.

Козак помітив, що у деяких квадратах є золоті монетки, а саме: у квадраті, який знаходиться у i-му рядку та j-му стовпчику, є рівно a_{ij} золотих монеток.

Просто забрати усі монетки — це занадто легко для Вуса, тому він вирішив, що забере усі монетки з квадратів, у яких кількість монеток парна.

Проте і така задача виявилася занадто легкою для нього, тому Козак Вус вирішив, що буде зсувати монетки: він може взяти **усі** монетки у якомусь квадраті та перекласти їх у будь-який сусідній квадрат. Квадрати вважаються сусідніми, якщо у них є спільна сторона. Описану операцію зсуву він може виконувати будь-яку кількість разів.

Тепер Козаку цікаво, яку максимальну кількість монет він може забрати. Допоможіть йому знайти цю кількість, а також допоможіть йому зрозуміти, як йому потрібно зсувати монетки, щоб забрати таку кількість.

Зверніть увагу, що Козаку не потрібно мінімізувати кількість операцій зсуву, йому потрібно лише максимізувати кількість монеток, які він забере.

Формат вхідних даних

Перший рядок містить два цілі числа t та g ($1 \leqslant t \leqslant 10$, $0 \leqslant g \leqslant 6$) — кількість тестів та номер блока.

Перший рядок кожного тесту містить два цілі числа n та m ($1 \leqslant n, m \leqslant 50$) — розміри поля.

Кожен з наступних n рядків містить m цілих чисел $a_{i1}, a_{i2}, \ldots, a_{im}$ ($0 \leqslant a_{ij} \leqslant 100$) — кількості золотих монеток у квадратах.

Не гарантується те, що є принаймні одна монетка на полі.

Формат вихідних даних

Для кожного тесту виконайте наступне:

У першому рядку виведіть одне ціле число — максимальну кількість монет, яку забере Вус.

У другому рядку виведіть одне ціле число p ($0 \leqslant p \leqslant 10\,000$) — кількість операцій зсуву, які необхідно виконати. Зверніть увагу, що не потрібно мінімізувати значення p.

У кожному з наступних p рядків виведіть по чотири цілі числа x_1, y_1, x_2, y_2 ($1 \leqslant x_1, x_2 \leqslant n$, $1 \leqslant y_1, y_2 \leqslant m$), які означають, що монетки, які знаходяться у квадраті (x_1, y_1), потрібно зсунути у квадрат (x_2, y_2).

Якщо існує кілька правильних відповідей, дозволяється вивести будь-яку з них. Гарантується, що існує оптимальна відповідь, де кількість операцій зсуву не перевищує 10 000.

Приклад

a.in або стандартний ввід	a.out або стандартний вивід
2 0 2 3 4 5 1 9 2 0 1 4 1 4 5 4	20 2 1 2 2 2 2 2 2 1 14 2 1 1 1 2 1 2 1 3

Примітка

У першому прикладі Козак може спочатку зсунути усі монетки з квадрата $(1, 2)$ у $(2, 2)$, після чого поле виглядатиме так:

```
4 0 1
9 7 0
```

Після зсуву монеток з $(2, 2)$ у $(2, 1)$ поле виглядатиме так:

```
4  0 1
16 0 0
```

Тому відповідь $4 + 16 = 20$.

У другому прикладі Козак може спочатку зсунути усі монетки з квадрата $(1, 1)$ у $(1, 2)$, після чого поле виглядатиме так:

```
0 5 5  4
```

Після зсуву монеток з $(1, 2)$ у $(1, 3)$ поле виглядатиме так:

```
0 0 10 4
```

Тому відповідь $10 + 4 = 14$.

Оцінювання

1. (14 балів) $n = 1$, усі a_{ij} парні;
2. (16 балів) $n = 1$, усі a_{ij} непарні;
3. (19 балів) $n = 1$;
4. (14 балів) $n, m > 1$, усі a_{ij} парні;
5. (17 балів) $n, m > 1$, усі a_{ij} непарні;
6. (20 балів) $n, m > 1$.

Задача B. Пари камінців

Назва вхідного файлу:	`b.in` або стандартний ввід
Назва вихідного файлу:	`b.out` або стандартний вивід
Обмеження використання часу:	1 секунда
Обмеження використання пам'яті:	256 мегабайтів

У Аліси та Боба є n купок з камінцями, пронумерованих від 1 до n. У i-ій купці a_i камінців. Вони виконують наступний процес:

1. Аліса вибирає дві непусті купки. Нехай у них x та y камінців відповідно.
2. Аліса забирає по одному камінцю з цих купок.
3. Далі Боб теж вибирає дві непусті купки так, щоб у них також було x та y камінців відповідно. Тобто Боб вибирає дві купки так, щоб у них була така ж кількість камінців, як було початково у купках Аліси. Бобу дозволяється вибирати купки, які вибрала Аліса. Якщо він це зробити не може, то процес закінчується.
4. Боб також забирає по одному камінцю з вибраних ним купок.
5. Якщо кількість непорожніх купок менша за 2, то описаний процес закінчується, інакше знову повторюється спочатку.

Зверніть увагу, що Аліса не зобов'язана вибирати кожного разу одні й ті ж самі значення x та y.

Аліса і Боб хочуть зробити так, щоб у купках залишилось якнайменше камінців. Зауважте, що в них спільна мета. Допоможіть їм знайти мінімальну кількість камінців, які можуть залишитися.

Формат вхідних даних

Перший рядок містить два цілі числа n та g ($2 \leqslant n \leqslant 2 \cdot 10^5$, $0 \leqslant g \leqslant 5$) — кількість купок та номер блока.

Другий рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$) — кількість камінців у кожній купці.

Формат вихідних даних

Виведіть одне число — відповідь на задачу.

Приклади

b.in або стандартний ввід	b.out або стандартний вивід
3 0 4 4 3	5
7 0 4 7 7 4 2 2 3	3

Примітка

У першому прикладі Аліса може спочатку вибрати першу та третю купки: $x = 4$ та $y = 3$. Після того, як вона забере камінці, у купках буде $3, 4, 2$ камінців відповідно.

Боб повинен вибрати купки, у яких $x = 4$ та $y = 3$ камінці, тому він вибере другу та першу купки відповідно. Після того, як він забере камінці, у купках буде $2, 3, 2$ камінців відповідно.

Наступного кроку Аліса може вибрати, наприклад, другу та третю купки: $x = 3$ та $y = 2$. Після того, як вона забере камінці, у купках буде $2, 2, 1$ камінців відповідно.

Боб не зможе вибрати такі купки, щоб в них було $x = 3$ та $y = 2$ камінців, тому процес закінчується. Залишається $2 + 2 + 1 = 5$ камінців.

Оцінювання

1. (17 балів) $n \leqslant 8$; $a_i \leqslant 8$;
2. (23 балів) n — парне; $a_1 = a_2,\ a_3 = a_4,\ \ldots,\ a_{n-1} = a_n$;
3. (22 балів) всі a_i — парні;
4. (18 балів) $a_i \leqslant 2 \cdot 10^5$
5. (20 балів) без додаткових обмежень.

Задача C. Топологічні сортування дерева

Назва вхідного файлу:	`c.in` або стандартний ввід
Назва вихідного файлу:	`c.out` або стандартний вивід
Обмеження використання часу:	1 секунда
Обмеження використання пам'яті:	256 мегабайтів

Вам дано дерево з n вершин, пронумерованих від 1 до n. Корнем дерева є вершина з номером 1. Для кожного v від 2 до n визначимо p_v — номер суміжної з v вершини, що лежить на простому шляху між вершиною v та корнем. На кожному з ребер (p_v, v) записаний символ «<» або «>».

Знайдіть кількість способів записати числа від 1 до n у вершинах дерева так, щоб кожне число було використано рівно один раз і при цьому виконувались усі відношення, вказані на ребрах. Тобто для усіх ребер з символом «<» повинно виконуватись $a[p_v] < a[v]$, а для усіх ребер з символом «>» повинно виконуватись $a[p_v] > a[v]$.

Через те, що відповідь може бути дуже великою, виведіть її за модулем $10^9 + 7$.

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leqslant n \leqslant 3\,000$) — кількість вершин дерева.

Кожен з наступних $n - 1$ рядків містить одне ціле число p_i ($1 \leqslant p_i < i$) та символ c_i ($c_i \in \{$«<», «>»$\}$), які позначають, що на ребрі між вершинами з індексами p_i та i записаний символ c_i. Зверніть увагу, що тут нумерація за i починається з 2.

Формат вихідних даних

Виведіть єдине ціле число — кількість способів розставити усі числа від 1 до n у вершинах дерева так, щоб виконувались усі відношення, вказані на ребрах. Зверніть увагу, що потрібно вивести не саму відповідь, а лише її остачу від ділення на $10^9 + 7$.

Приклади

c.in або стандартний ввід	c.out або стандартний вивід
4 1 < 2 < 3 >	3
4 1 < 1 < 1 <	6
5 1 < 1 < 3 > 3 >	18

Оцінювання

1. (8 балів) $n \leqslant 10$;
2. (6 балів) $n \leqslant 18$;
3. (10 балів) $c_i =$ '<';
4. (4 бали) $p_i = 1$;
5. (13 балів) $p_i = i - 1, 1 \leqslant n \leqslant 200$;
6. (19 балів) $p_i = i - 1$;
7. (24 бали) $n \leqslant 200$;
8. (16 балів) без додаткових обмежень.

Задача D. Додай ребра

Обмеження використання часу: N/A
Обмеження використання пам'яті: N/A

Дано дерево з n вершин та $n-1$ ребер. Потрібно додати рівно m нових ребер до цього дерева. Додавати кратні ребра або петлі заборонено. Якщо перед додаванням ребер у певної вершини була степінь t_i, то після додавання ребер її степінь не має перевищувати $t_i + k$. Нагадаємо, що степінь вершини — це кількість ребер, які її з'єднують з іншими вершинами.

Крім цього, також дано $m+n-1$ цілих чисел $a_1, a_2, \ldots, a_{m+n-1}$. Після додавання ребер потрібно зіставити кожному ребру один з елементів масиву a так, щоб кожен елемент масиву відповідав рівно одному ребру. Значення a_i зіставленому ребру позначатиме його вагу.

Потрібно додати нові ребра та зіставити числа до ребер так, щоб сума найкоротших відстаней між кожною парою вершин була **максимальною**. Тобто потрібно максимізувати функцію $\sum d_{ij}$, де d_{ij} — мінімальна відстань між вершинами, для всіх i та j $(1 \leqslant i < j \leqslant n)$. Мінімальною відстанню між вершинами вважається сумарна вага усіх ребер на простому шляху між ними.

Зверніть увагу, що у цій задачі потрібно відправляти не код, а самі відповіді. Вам також доступні усі тести, які можна завантажити за посиланням `https://c.oi.in.ua/uoi2020/mulBeIDcmQIo/`

Формат вхідних даних

Дано 5 тестів.

Перший рядок кожного тесту містить три цілі числа n, m та k $(1 \leqslant n \leqslant 5\,000$, $1 \leqslant m \leqslant 250\,000$, $1 \leqslant k \leqslant 500)$ — кількість вершин у дереві, кількість ребер, які потрібно додати, та максимальна кількість ребер, які можна додати до однієї вершини.

Другий рядок містить $m+n-1$ цілих чисел $a_1, a_2, \ldots, a_{m+n-1}$ $(1 \leqslant a_i \leqslant 10^6)$.

Наступні $n-1$ рядків містять по два цілі числа v_i та u_i $(1 \leqslant v_i, u_i \leqslant n)$ — номери вершин, між якими є ребро. Гарантується, що початково заданий граф — дерево.

Гарантується, що завжди можна додати ребра так, щоб всі описані в умові вимоги виконувались.

Формат вихідних даних

Для кожного тесту виведіть наступне:

Якщо у вас є відповідь на цей тест, виведіть 1, інакше 0.

Якщо у вас є відповідь, то у кожному з наступних $m+n-1$ рядків виведіть по три цілі числа v_i, u_i, a_i — номери вершин, між якими є ребро у кінцевому графі, та його вага.

Оцінювання

Нехай d_0 певна змінна у тесті, а d — сума відстаней у вашому графі. Якщо $d > d_0$, то ви отримаєте 20 балів за тест. Інакше ви отримаєте стільки балів за тест:

$$(100^{(d-d_0)/d_0})^5 \cdot 20$$

Якщо s — сума балів за всі тести, то за спробу ви отримаєте s, заокруглене до найближчого цілого числа.

Приклад

Вхідні дані	Вихідні дані
4 1 1 1 2 4 2 1 2 2 3 4 2	1 1 2 1 3 2 2 2 4 4 1 3 2