

XXX Всеукраїнська олімпіада з інформатики

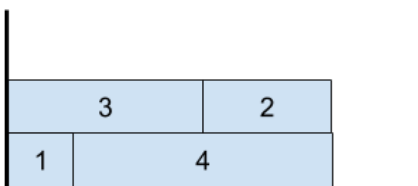
Другий тур

А. Пірамідка

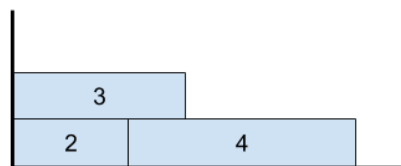
Петрик обожає конструктори. Цього разу він захотів погратися з набором прямокутних кубиків. Цей набір складається з n кубиків, причому містить рівно по одному кубу кожного з розмірів 1×1 , 1×2 , ..., $1 \times n$. Петрик хоче вибрати деякі кубики з набору та побудувати з них пірамідку.

Пірамідка складається з одного або більше рівнів висотою один; в кожному рівні кубики розташовані горизонтально, впритул один до одного меншою стороною, а найлівіший кубик у рівні розташований впритул до стіни; і кожен наступний рівень не більше за довжиною, ніж попередній (рахуючи знизу догори). При цьому Петрик вважає, що пірамідка буде більш стійкою, якщо кожен кубик у кожному рівні, окрім найнижчого, буде лежати не менше, ніж на двох різних кубиках з попереднього рівня.

Наприклад, ліва конструкція не буде піраміdkою, бо кубик 1×2 лежить лише на одному кубу, а права – буде.



Не пірамідка



Пірамідка

Завдання

Знайдіть, піраміdkу з якою найбільшою кількістю рівнів можна побудувати з деяких кубиків з набору з n кубиків, та виведіть конструкцію цієї піраміdkи. Якщо таких пірамідок декілька, то виведіть довільну.

Вхідні дані

В єдиному рядку вхідного файлу **pyramid.dat** знаходиться ціле число n ($1 \leq n \leq 10^5$).

Вихідні дані

В першому рядку вихідного файлу **pyramid.sol** виведіть ціле число M – кількість рівнів у піраміdkі з найбільшою можливою кількістю рівнів. Далі виведіть M рядків, які описують рівні знизу догори, кожен з цих рядків починається числом k_i – кількістю кубиків у відповідному рівні, далі k_i чисел – довжини кубиків у відповідному рівні в порядку зліва направо.

Оцінювання

Додатково гарантуються такі умови:

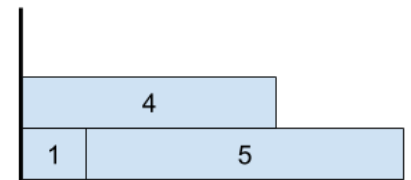
- 30 % тестів: $n \leq 7$
- 60 % тестів: $n \leq 100$

Приклад вхідних та вихідних даних

pyramid.dat	pyramid.sol
5	2 2 1 5 1 4

Пояснення до прикладу

Тестовому прикладу відповідає така пірамідка:



В. OrangeBook та наліпки

Нещодавно програмісту Васі подарували його перший OrangeBook. Для початку, він вирішив обклеїти кришку цього ноутбуку наліпками. Відомо, що кришка являє собою прямокутник розмірами $N \times M$, у центрі якої знаходиться прямокутний логотип, розмірами $K \times L$. Вася хоче розмістити на кришці якомога більше однакових прямокутних наліпок із логотипами Sney Inc., які будуть орієнтовані в одну сторону. Очевидно, що ніякі 2 логотипи не повинні перетинатися. Вася може замовити скільки завгодно однакових прямокутних наліпок розміром $A \times B$, де A і B – довільні натуральні числа. Причому, щоб наліпки не здавалися занадто малими чи занадто великими, їх периметр повинен дорівнювати $2P$.

Завдання

Напишіть програму, яка за даними натуральними числами N , M , K , L та P знайде максимальну кількість наліпок, які Вася зможе наклеїти.

Вхідні дані

В єдиному рядку вхідного файлу **orangebook.dat** записано 5 натуральних чисел – N , M , K , L та P (N , M , $P \leq 10^6$; $K < N$; $L < M$), причому N , M , K , L – парні.

Вихідні дані

В вихідний файл **orangebook.sol** виведіть одне число – максимальну кількість наліпок, яку зможе наклеїти Вася.

Оцінювання

Додатково гарантуються такі умови:

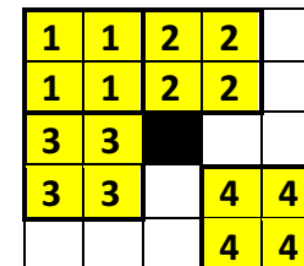
- 20 % тестів: $N, M \leq 7$
- 60 % тестів: $N, M \leq 100$

Приклад вхідних та вихідних даних

orangebook.dat	orangebook.sol
5 5 1 1 4	4

Пояснення до прикладу

Можна розмістити 4 наліпки розмірами 2×2 , як показано на малюнку:



C. Поля

У деяких клітинках прямокутної таблиці $N \times M$ (N рядків, M стовпців) розташовано центри полів олімпіадної енергії. Сила кожного такого поля виражається натуральним числом S . Кожне таке поле діє не лише на клітинку, в якій розташовано, а й на певний окіл цієї клітинки, а саме:

- на дану клітинку (центр поля) поле діє з силою S ;
- на сусідні з центром поля клітинки (по стороні) поле діє з силою $S - 1$ (якщо це число додатне);
- на сусідні з ними клітинки, тобто на всі клітинки на відстані 2 від центра поля воно діє з силою $S - 2$ (якщо це число додатне);
- ...
- на всі клітинки на відстані $S - 1$ від центра поля воно діє із силою 1;
- на решту клітинок поле не діє.

Завдання

За інформацією про поля в таблиці для кожної її клітинки відновіть сумарну силу полів, що на неї діють.

Вхідні дані

У першому рядку вхідного файлу **fields.dat** вказано натуральні числа N та M , що не перевищують 700, — розміри таблиці. У кожному з наступних N рядків вказано по M невід’ємних цілих чисел, не більших за 3000: силу поля з центром у даній клітинці або 0, якщо дана клітинка не є центром поля.

Вихідні дані

У N рядках вихідного файлу **fields.sol** виведіть по M чисел: для кожної клітинки таблиці сумарну силу полів, що на неї діють.

Оцінювання

- У 30% тестів $\text{Max}\{N, M\} < 40$;
- У 30% тестів $40 \leq \text{Max}\{N, M\} \leq 400$;
- У 40% тестів $\text{Max}\{N, M\} > 400$.

Приклад вхідних та вихідних даних

fields.dat	fields.sol
2 3	6 8 6
2 1 3	5 6 7
0 5 1	

D. День Народження

У Стаса незабаром день народження, і з цієї нагоди він вирішив влаштувати грандіозну вечірку. Звісно, Стас розіслав запрошення всім своїм знайомим. Очікується велика кількість гостей, тому планується грандіозне шоу.

Усі гості сповістили Стаса, в який час вони прийдуть на вечірку, та о котрій будуть змушені її залишити. Проаналізувавши цю інформацію, Стас склав розклад прибуття та відбуття гостей, адже він хоче, щоб все пройшло ідеально, і належним чином підготуватись.

Вечірка буде продовжуватись N хвилин. Стас знає, що рівно K гостей зможуть прийти до початку вечірки. Далі, протягом усієї вечірки кожної хвилини або прибуде новий гість, або ж хтось із присутніх її залишить.

Докладаючи великих зусиль, Стасу вдалося отримати згоду свого улюбленого Відомого Артиста на виконання однієї пісні. Для нього на вечірці буде побудована величезна сцена, перед якою стоятиме нескінченна кількість стільців, пронумерованих послідовними натуральними числами, починаючи з одиниці.

На жаль, є одна проблема. Відомий Артист має дуже щільний графік і точно не знає, в який момент часу він зможе приїхати виступати. Тому Стас вирішив зайняти гостей іншими розвагами, а коли Артист приїде і буде готовий розпочати, швиденько розсадить усіх гостей на підготовлені стільчики перед сценою.

Розсаджування відбувається таким чином:

Спочатку Стас визначає порядок розсадження. Гостям пропонується розсаджуватися по одному саме в тій послідовності, яку визначив наш герой.

Однак, кожен друг Стаса має свої примхи. Наприклад, кожен має своє улюблене число. Тому, коли хтось обирає, на який стілець йому сісти, він намагається обрати стілець зі своїм улюбленим номером. Якщо він вже зайнятий, гість ображається, але вирішує сісти на стілець, який стоїть одразу за ним. Якщо зайнятим буде і той стілець, гість образиться ще сильніше, але буде шукати наступний вільний стілець. Можна вважати, що цей процес відбувається миттєво. У підсумку, загальну образу гостя на Стаса можна виразити різницею номерів зайнятого і бажаного місць. Стас – прекрасний товариш, тому йому відоме улюблене число кожного з його знайомих.

Виступ Артиста триватиме рівно одну мить, тому Стасу не потрібно турбуватися, що комусь під час виконання пісні потрібно буде піти, або прийде новий гість, якого потрібно буде посадити.

Звісно, Стас хоче мінімізувати сумарну образу гостей.

Стасу необхідно для кожної хвилини вечірки розрахувати мінімальну сумарну образу, якщо потрібно буде розсадити усіх наразі присутніх гостей. Для нього це виявилось заважким завданням, тому він просить Вас допомогти.

Вхідні дані

У першому рядку вхідного файлу **party.dat** містяться два натуральних числа N і K - тривалість вечірки в хвилинах та кількість гостей, що прийдуть до початку вечірки.

Наступний рядок містить K натуральних чисел, розділених пробілом – улюблені числа гостей, які присутні до початку вечірки.

У кожному з N наступних рядків йде розклад прибуття-відбуття гостей. Рядки містять по два числа, T та X . Якщо T дорівнює 1, це означає, що в відповідну хвилину прийде новий гість, улюблене число якого дорівнює X . Інакше, вечірку буду змушений залишити гість, який мав число X за улюблене; гарантується, що в такому разі хоча б один гість з улюбленим числом X був присутній на вечірці.

Усі числа у вхідному файлі натуральні та не перевищують $5 \cdot 10^4$. T приймає значення 0 або 1.

Вихідні дані

У першому рядку вихідного файлу **party.sol** виведіть через пробіл N чисел - мінімальну сумарну образу після розсадження всіх присутніх гостей для кожної хвилини.

Оцінювання

- У 15% тестів - усі числа натуральні, не перевищують 5
- У 35% тестів - гості не залишають вечірки (усі T рівні одиниці)

Приклад вхідних та вихідних даних

party.dat	party.sol
4 3	4 1 1 3
1 1 2	
1 2	
0 1	
0 1	
1 2	

Пояснення до прикладу

До початку вечірки прийдуть гості, з улюбленими числами 1, 1 і 2. Вечірка починається, і у першу хвилини приходить ще один гість з числом 2. Маємо чотири гостя. Оптимальною може бути, наприклад, така розсадка:

Спочатку ми пропонуємо зайняти свої місця друзям с числами 1 та 2. Вони без проблем по черзі займають їх, не ображаючись на Стаса. Далі запропонуємо зайняти місце гостю з номером 1. Оскільки стільчики 1 та 2 зайняті, він займає третій стілець, ображаючись на Стаса на величину $3 - 1 = 2$. Останнім сідає гість з числом 2, займає 4 місце та ображається на $4 - 2 = 2$. Сумарна образа усіх гостей дорівнює 4.

О другій хвилині вечірку залишить гість з улюбленим числом 1, і залишаться лише люди з числами 1, 2, 2.