**Освітлені будівлі**

|  |  |
| --- | --- |
| **Input file name:** | illuminated.in |
| **Output file name:** | illuminated.out |
| **Time limit:** | 100 ms |
| **Memory limit:** | 256 M |

Степан з метою додаткового заробітку вирішив встановити в котеджному містечку надсучасний ліхтар, промені якого паралельні поверхні землі, щоб освітлити вулицю. На вулиці розташовані будівлі, задані висотами стін в метрах. Будівля вважається освітленою, якщо для прямих променів ліхтаря відкритий хоча б один метр стіни.  
Степан вирішив підрахувати кількість освітлених будівель, щоб знати скільки він зможе заробити. Допоможіть Степану - напишіть програму, яка вирішить його проблему.

**Вхідні дані:**Перший рядок вхідного файлу містить ціле число ***N (1 ≤ N ≤ 100000)*** - кількість будівель в котеджному місте чку. У другому рядку записано від ***N*** чисел - висоти будівель ***Hi (1 ≤ Hi ≤ 100000)***.

Будемо вважати, що всі будівлі розташовуються в один ряд на вулиці, орієнтованої зі сходу на захід, ліхтар освітлює їх з боку першої будівлі.

**Вихідні дані:**Єдиний рядок вихідного файлу має містити одне ціле число - кількість освітлених будівель.

**Малюнки до прикладів:**



**Приклади вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **illuminated.in** | **illuminated.out** |
| 4  1 2 2 3 | 3 |
| 5  2 1 3 1 2 | 2 |

**Завдання з біології**

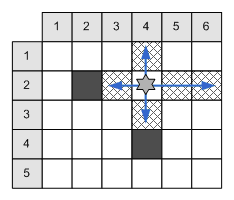
|  |  |
| --- | --- |
| **Input file name:** | biology.in |
| **Output file name:** | biology.out |
| **Time limit:** | 200 ms |
| **Memory limit:** | 256 M |

Cтепан, готуючись до олімпіади з програмування, пропустив багато уроків біології, і щоб виправити становище, взявся за непросте завдання. Це завдання полягає в аналізі зображень, отриманих за допомогою електронних мікроскопів.

Перед проведенням аналізу зображення проходять кілька стадій фільтрації. На першій стадії зображення переводиться в монохромне, тобто що має тільки два кольори: білий та чорний. Друга стадія є більш складною і полягає у видаленні шумів із зображення шляхом підбору фільтра. Але це виявилося непросто, оскільки для різних видів зображень потрібні різні фільтри. З метою підбору відповідного фільтра було вирішено перед другою стадією фільтрації проводити додаткові розрахунки.

Додаткові розрахунки полягають у знаходженні коефіцієнта зв'язності вихідного зображення. Коефіцієнт зв'язності - це число різних пар сусідніх пікселів.  
Зображення являє собою прямокутну таблицю пікселів розміру ***NxM***.  
Кожен піксель має або білий колір, або чорний. Два різних пікселя називаються сусідніми, якщо виконуються наступні умови:

1. Обидва пікселя мають білий колір.
2. Обидва пікселя знаходяться або в одному і тому ж стовпці або в одному і тому ж рядку.
3. Всі пікселі, що перебувають або в рядку, або в стовпці між зазначеними пікселями, є білими.

   
***Малюнок №1:*** N = 5 і M = 6, приклад сусідніх символів для пікселя (2, 4).

Для наведеного вище прикладу всі зафарбовані пікселі є сусідніми для пікселя, позначеного зірочкою. Дві пари сусідніх пікселів, які можуть бути отримані одна з іншої за допомогою перестановки елементів, не вважаються різними.Тобто пара сусідніх пікселів {(2; 4), (2; 6)} і пара {(2; 6), (2; 4)} не рахуються різними.  
Степану потрібно написати програму, яка по заданому опису зображення розраховує коефіцієнт зв'язності даного зображення. Допоможіть йому.

**Вхідні дані:**Перший рядок вхідного файлу містить два цілих числа ***N*** і ***M (1 ≤ N, M ≤ 200)***. Далі йдуть ***N*** рядків по ***M*** символів в кожній, що описують задане монохромне зображення. Символ '.' (ASCII 46) - позначає білий символ, а '#' (ASCII 35) - чорний.

**Вихідні дані:**Одне ціле число - коефіцієнт зв'язності вихідного монохромного зображення.

**Приклади вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **biology.in** | **biology.out** |
| 2 3  ...  .#. | 5 |
| 3 4  ...#  .#..  ...# | 13 |

**Гра з рядком**

|  |  |
| --- | --- |
| **Input file name:** | string.in |
| **Output file name:** | string.out |
| **Time limit:** | 300 ms |
| **Memory limit:** | 256 M |

Степан дуже полюбляє паліндроми. Паліндромом називається рядок, який однаково читається як зліва направо, так і справа на ліво.

Підпаліндромом даного рядка називається послідовність символів з даного рядка, які не обов'язково йдуть поспіль, і є паліндромом.

Наприклад, ***ABA*** є підпаліндромом рядка ***ACBFADEF***.

Степан вирішив пограти з рядком. Він придумує рядок і бере з даного рядка підрядок (послідовність символів з ***l*** по ***r***, що йдуть поспіль) та визначає паліндром максимальної довжини. Потім бере новий підрядок і вже з нього визначає паліндром максимальної довжини, і так ***k*** разів.

Напишіть програму, яка допоможе Степану.

**Вхідні дані:**Перший рядок вхідного файлу містить рядок довжиною не більше 100 символів з великих букв латинського алфавіту.

Другий рядок містить одне число ***k (1 ≤ k ≤ 10)*** - кількість разів, яку Степан зіграв з даним рядком.

Наступні ***k*** рядків містять по два числа ***l*** та ***r*** - номери першого та номер останнього символа рідрядка.

**Вихідні дані:** Вихідний файл має містити ***k*** рядків, у кожному з яких має бути записана довжина максимального підпаліндрома та через пробіл сам максимальний підпаліндром для відповідних ***l*** та ***r*** у вхідних даних. Якщо таких підпаліндромів кілька, то виведіть любий з них.

**Приклади вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **string.in** | **string.out** |
| ACBFADEF  3  1 5  2 6  2 8 | 3 ABA  1 C  3 FAF |

**Адміністративна реформа**

|  |  |
| --- | --- |
| **Input file name:** | reform.in |
| **Output file name:** | reform.out |
| **Time limit:** | 300 ms |
| **Memory limit:** | 256 M |

В Ужляндії вирішили провести адміністративно-територіальну реформу. Як відомо, в Ужляндії є ***N*** міст, розташованих уздовж довгої прямої дороги, ***і***-е місто розташоване на відстані ***Xi (0 ≤ X1 < X2 < ... < XN ≤ 109)*** кілометрів від початку дороги.  
В рамках реформи було прийгято рішення розділити Ужляндію на ***К*** регіонів. Кожне місто має увійти рівно в один регіон.

У кожен регіон увійде від ***A*** до ***B*** міст, до того ж, ці міста повинні мати номера, що йдуть поспіль. Таким чином, кожен регіон характеризується числами ***i*** та ***l***, для яких ***1 ≤ i, i + l - 1 ≤ N, A ≤ l ≤ B*** і в регіон входять міста з номерами ***і, i + 1, ... , i + l - 1***.  
Щоб мінімізувати витрати на обслуговування регіонів, адміністрація хоче, щоб максимальна відстань між містами, що входять в одну провінцію, було якомога менше.   
Допоможіть виконати поділ Ужляндії.

**Вхідні дані:** Перший рядок вхідного файлу містить чотири цілих числа: ***N, K, A, B (1 ≤ N ≤ 200, 1 ≤ K ≤ N, 1 ≤ A ≤ B ≤ N, AK ≤ N ≤ BK)***. Другий рядок вхідного файлу містить ***N*** цілих чисел: ***X1, X2, ..., XN (0 ≤ X1 < X2 < ... < XN ≤ 109)***.

**Вихідні дані:** Виведіть одне ціле число: мінімально можливе ***Z***, таке щоб можна було розбити міста на регіони описаним чином, і відстань між містами всередині одного регіону не перевищувала ***Z***.

**Пояснення до прикладу:**У прикладі оптимально перші 4 міста об'єднати в перший регіон, а п'яте і шосте - у другий. Максимальна відстань між двома містами в одному регіоні: 13 - 6 = 7.

**Приклади вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **reform.in** | **reform.out** |
| 6 2 2 4  1 2 3 4 6 13 | 7 |

**Степан i "Великi любителi Гаррi Потера"**

|  |  |
| --- | --- |
| **Input file name:** | spell.in |
| **Output file name:** | spell.out |
| **Time limit:** | 300 ms |
| **Memory limit:** | 128 M |

Степан дуже любить серiю фантастичних романiв "Гаррi Поттер". Вiн так сильно нею захоплюється, що входить в групу "Великi любителi Гаррi Потера" та навiть склав збiрку всiх заклинань, яким вчать в Гоґвартс. Також Степан вивчив всi правила та закономiрностi цих заклинань.

Самi по собi заклинання нiчого не вартi, їх потрiбно вимовляти в певнiй послiдовностi, яка називається комбiнацiєю заклинань.

Ця послiдовнiсть має починатись з базового заклинання, та закінчуватись на заклинання, яке називається основним.

Поряд в комбінації можуть йти лише, так звані, спарені заклинання, наприклад, пiсля базового можна промовити лише спаренi з ним, Степан вивчив які заклинання є спареними. Комбiнацiя записується як конкатенацiя заклинань (без промiжкiв мiж ними).  
Степан визначив яким має бути основне (останнє) заклинання в комбiнацiї, тепер вiн хоче представити свою комбінацію публiцi. Для того, щоб група любителiв Гаррi Поттера прийняла цю комбiнацiю, необхiдно, щоб її довжина була наймешною серед всiх можливих (адже так простiше буде вчити). Також Степан помiтив, що лексикографічно малі комбінації краще приймаються публікою, хто знає, можливо це важливо. Тому необхiдно серед всiх можливих найкороштих комбiнацiй вибрати лексографiчно найменшу.  
Допоможiть Степану знайти таку комбiнацiю.

**Вхідні дані:** В першому рядку знаходиться два цiлих числа ***N*** i ***M (2 ≤ N ≤ 1000, 1 ≤ M ≤ 250000)*** — кiлькiсть заклинань та кiлькiсть пар спарених заклинань вiдповiдно. Далi в ***N*** рядках записанi самi заклинання, до того ж, перше заклинання — базове, останнє заклинання — заклинання на яке комбiнацiя має закiнуватись. Всi заклинання складаються лише з маленьких лiтер латинського алфавiту, їх довжини не перевищують 1000 символiв. Далi в ***M*** рядках записано по два рiзних числа ***Ai*** i ***Bi*** — номера заклинань, якi є спареними. Пари заклинань не повторюються.

**Вихідні дані:** В єдиному рядку виведiть шукану комбiнацiю.

**Оцінювання:**

N ≤ 50, назви заклинань з номерами від 1 до N складаються з однієї букви – не менше 30 балів

N ≤ 50 – не менше 60 балів

N ≤ 250 – не менше 75 балів

**Приклади вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **spell.in** | **spell.out** |
| 2 1  avada  kedavra  1 2 | avadakedavra |
| 4 4  sectum  crucio  reparo  sempra  1 2  1 3  2 4  3 4 | sectumcruciosempra |