**Лабораторна робота №2**

**Варіант 1.**

Компанія SPAM займається розсилкою рекламних газет та буклетів у різні міста. За домовленістю з поштою кожна посилка повинна містити строго певну кількість (N, де 1 > N > 100) примірників рекламних видань у сумі. Але є ще проблема та принцип компанії:

1) Два буклети не можуть стикатися, через неякісне поліграфічне покриття вони склеюються і псуються. Газети виконують ще й роль прокладання між ними.

2) Кожна посилка має бути унікальною, тобто відрізнятися від інших порядком укладання або кількістю різних типів видань (буклет/газета). Це додатковий рекламний хід компанії, який полягає в тому, що кожен клієнт повинен отримувати індивідуальне укладання або зміст посилки.

Необхідно наперед визначити максимальну кількість варіантів правильної комплектації посилки в залежності від N.

Вхідні дані:

На початку файлу "SPAM.IN" записана кількість екземплярів видань (деяка N).

Вихідні дані:

У першому рядку файлу "SPAM.OUT" необхідно вказати кількість можливих варіантів комплектації.

приклад 1.

Файл "SPAM.IN"4

Файл "SPAM.OUT"

8

приклад 2.

Файл "SPAM.IN"13

Файл "SPAM.OUT"

610

**Варіант 2.**

Послідовність клітин занумерована числами від 1 до N. У кожній клітині стоїть або чорна або біла фішка. Групою назвемо набір фішок одного кольору, що поспіль стоять, обмежений з обох сторін фішками іншого кольору або кінцями послідовності. Слід перемістити фішки так, щоб вони утворили трохи більше двох груп.

Переміщення фішок описується за допомогою плану обміну, в якому використовуються поняття операція обміну та крок. Операція обміну змінює місцями дві сусідні групи фішок. Крок складається не більше ніж K одночасно виконуваних обмінів. Обміни можна здійснювати одночасно лише тоді, коли в них беруть участь різні групи. Після кожного кроку групи одного кольору, що опинилися поряд, об'єднуються. План обмінів містить описи кроків, що виконуються послідовно.



Напишіть програму, яка визначає план обмінів, за допомогою якого за найменшу кількість кроків виходить послідовність, що складається не більше ніж з двох груп.

Формат вхідних даних

У першому рядку вхідного файлу записані числа N і K (1<=N<=100000 та 1<=K<=10000). Вихідна розстановка фішок визначається у наступних рядках, що містять N чисел (0 або 1), розділених пробілами або перекладами рядків. При цьому 0 відповідає чорній фішці, 1 - білій.

Формат вихідних даних

Вихідний файл повинен містити опис кроків плану по одному кроці на рядку. Опис кроку починається з числа L – кількості обмінів на цьому кроці. Потім кожного обміну вказується мінімальний номер клітини, у якій стоїть фішка, що у цьому обміні. Останній рядок плану має містити одне число 0.

Приклади

fishes.in

9 3

1 0 0 1 1 0 1 1 0

fishes.out

2 1 6

1 1

0

fishes.in

3 1

1 1 0

fishes.out

0

Примітка

Потрібно знайти план, що містить найменшу кількість кроків, причому загальна кількість обмінів може бути не мінімальним.

**Варіант 3.**

У деякому місті є метро, ​​що складається з N (1 <= N <= 1000) станцій та M ліній, що з'єднують їх. Кожна лінія забезпечує проїзд між двома станціями в обидві сторони. Між будь-якою парою станцій проведено не більше однієї лінії. Мережа метро побудована таким чином, щоб із кожної станції можна було проїхати на кожну (можливо через проміжні станції). Назвемо цю властивість зв'язковістю метро.

У зв'язку з винаходом нового виду транспорту метро стало збитковим, і його роботу вирішили припинити. На засіданні мерії міста було ухвалено закривати щороку по одній станції, але так, щоб зв'язність метро щоразу зберігалася. При закритті будь-якої станції лінії, що ведуть з цієї станції в інші, природно, теж перестають функціонувати.

Завдання. За введеною інформацією про мережу метро розробити будь-який порядок закриття станцій, у якому метро завжди залишатиметься зв'язковим. Наприклад, нехай метро виглядає так, як показано на малюнку. Тоді станції можна закривати, наприклад, у порядку 1, 2, 4, 3, 5. А порядок 3, 1, 2, 4, 5 - не підходить, тому що після закриття 3-ї станції метро розпадеться на чотири не зв'язкові між собою частини.



Введення. Перший рядок вхідного файлу міститиме числа N та M. У наступних M рядках знаходиться інформація про лінії. Кожен із цих рядків містить через пропуск числа Ai і Bi (Ai Bi) - дві станції, які з'єднує i-я лінія.

Висновок. Вихідний файл має складатися із N рядків. Кожен рядок має містити одне число – номер станції. Вивести станції потрібно у порядку їх закриття.

приклад

input.txt

5 4

3 1

3 2

3 4

3 5

output.txt

1

2

4

3

5

**Варіант 4.**

З метою підготовки до проведення олімпіади з інформатики мер вирішив забезпечити надійним електропостачанням усі школи міста. Для цього необхідно провести лінію електропередач від альтернативного джерела електроенергії "Майбуття" до однієї зі шкіл міста (до якої не має значення), а також з'єднати лініями електропередач деякі школи між собою.

Вважається, що школа має надійне електропостачання, якщо вона безпосередньо пов'язана з джерелом "Майбуття", або з однією з шкіл, які мають надійне електропостачання.

Відома вартість з'єднання між деякими парами шкіл. Мер міста вирішив обрати одну з двох найбільш економічних схем електропостачання (вартість схеми дорівнює сумі вартості пар шкіл).

Завдання

Напишіть програму SCHOOLS, яка обчислює вартість двох найбільш економних схем альтернативного електропостачання шкіл.

Вхідні дані

У першому рядку вхідного файлу SCHOOLS.DAT знаходяться два натуральні числа, розділені пробілом: N (3 <= N <= 100), кількість шкіл у місті, і M - кількість можливих з'єднань між ними. У кожному з наступних M рядків знаходяться по три числа: Ai, Bi, Ci, розділених пробілами, де Ci - вартість прокладки лінії електропостачання (1 <= Ci <= 300) від школи Ai до школи Bi (i = 1, 2, . .., N).

Приклад вхідного файлу

5 8

1 3 75

3 4 51

2 4 19

3 2 95

2 5 42

5 4 31

1 2 9

3 5 66

Вихідні дані

У єдиному рядку вихідного файлу SCHOOLS.SOL повинні міститися два натуральні числа S1 і S2, розділених пробілом - дві найменші вартості схем (S1 <= S2). S1=S2 і тоді, коли є кілька схем надійного електропостачання найменшої вартості.

Приклад вихідного файлу

110 121

**Варіант 5.**

Є гирі з масами: 1 г, 2 г, ..., N г (N <= 500 000). Написати програму, що розподіляє ці гирі на максимально можливу кількість пар так, щоб сумарна вага гир у кожній парі виражалася простим числом.

Вхідні дані:

Вхідний файл input.txt містить N.

Вихідні дані

У вихідному файлі output.txt відображається список пар. Усі числа у вихідному файлі поділяються пробілами та (або) символами перекладу рядка.

input.txt

7

output.txt

1 6

7 4

5 2

**Варіант 6**

Дивні часи настали у Лощині Бурштинової Роси. Усі кудись біжать, щось купують-продають, постійно намагаючись обдурити один одного. Немає колишнього спокою. Смутні часи не обминули і Монастир Світлого Місяця: Ніколи ще не було такого, щоб звичайний торговець намагався обдурити ченців, боязнь гніву Будди зупиняла його. Але й цей страх померк перед пристрастю наживи.

Мудрий Настоятель підозрює, що один із постачальників Монастиря нечистий на руку. Відомо, що під час розрахунку вартості товару він використовує Калькулятор. Цей Калькулятор вміє не так вже й багато... Все що він це вміє:

1. ввести число 1

2. подвоїти поточне число

3. поміняти в поточному числі першу та останню цифри.

Калькулятор може працювати тільки з цілими числами від 1 до 10000.

Зазвичай Торговець привозить до Монастиря товар, потім, користуючись Калькулятором, підраховує вартість товару, називає суму Настоятелю, і Настоятель оплачує товар. Настоятель хоче дізнатися, чи не обманює його Торговець, називаючи суму, яку не можна отримати за допомогою Калькулятора. Допоможіть йому у цьому.

Введення. У файлі знаходиться однина k - сума, названа Торговцем (1 <= k <= 10000)

Висновок. Виведіть "YES", якщо сума може бути отримана за допомогою Калькулятора, та "NO" в іншому випадку.

Приклад input.txt #1

8042

Приклад output.txt #1

YES

Приклад input.txt #2

3

Приклад output.txt #2

NO

**Варіант 7.**

Багатокутник (не обов'язково опуклий) на площині заданий координатами своїх вершин. Потрібно підрахувати кількість точок з цілими координатами, що лежать усередині нього (але не на його кордоні).

Формат вхідних даних

У першому рядку міститься N (3<=N<=1000) - число вершин багатокутника. У наступних N рядках йдуть координати (Xi, Yi) вершин багатокутника в порядку обходу за годинниковою стрілкою. Xi та Yi - цілі числа, по модулю не перевищують 1000000.

Формат вихідних даних

У вихідний файл вивести одне число - кількість точок, що шукається.

Приклади

e.in

4

-1 -1

-1 1

1 1

1 -1

e.out

1

e.in

3

0 0

0 2

2 0

e.out

0

**Варіант 8.**

Ви є одним із розробників нової комп'ютерної гри. Гра відбувається на прямокутній дошці, що складається з клітин W\*H. Кожна клітина може або містити або не містити фішку (див. малюнок).

Важливою частиною гри є перевірка того, чи з'єднані дві фішки шляхом, що задовольняє наступні властивості:

1. Шлях повинен складатися з відрізків вертикальних та горизонтальних прямих.
2. 2. Шлях не повинен перетинати інші фішки.
3. При цьому частина шляху може опинитися поза дошкою. Наприклад:



Фішки з координатами (1,3) та (4,4) можуть бути з'єднані. Фішки з координатами (2,3) та (5,3) теж можуть бути з'єднані. А ось фішки з координатами (2,3) і (3,4) з'єднати не можна - будь-який шлях, що їх з'єднує, перетинає інші фішки.

Вам необхідно написати програму, яка перевіряє, чи можна з'єднати дві фішки шляхом, що має вищезгадані властивості, і, у разі позитивної відповіді, визначальну мінімальну довжину такого шляху (вважається, що шлях має злами, початок і кінець тільки в центрах клітин (або "уявних клітин") ", розташованих поза дошкою), а відрізок, що з'єднує центри двох сусідніх клітин, має довжину 1).

Формат вхідних даних

Перший рядок вхідного файлу містить два натуральні числа: W - ширина дошки, H - висота дошки (1 <= W, H <= 75). Наступні H рядків містять опис дошки: кожен рядок складається з W символів: символ "X" (заголовна англійська літера "ікс") позначає фішку, символ "." (Точка) позначає порожнє місце. Всі інші рядки містять описи запитів: кожен запит складається з чотирьох натуральних чисел, розділених пробілами - X1, Y1, X2, Y2, причому 1 <= X1, X2 <= W, 1 <= Y1, Y2 <= H. Тут (X1 , Y1) та (X2, Y2) - координати фішок, які потрібно з'єднати (ліва верхня клітина має координати (1,1)). Гарантується, що ці координати не збігатимуться (крім останнього запиту; див. далі). Останній рядок містить запит, що складається із чотирьох чисел 0; цей запит обробляти не треба.

Формат вихідних даних

Для кожного запиту необхідно вивести одне ціле число на окремому рядку - довжину найкоротшого шляху або 0, якщо такого шляху не існує.

Приклади

g.in

5 4

XXXXX

X...X

XXX.X

.XXX.

2 3 5 3

1 3 4 4

2 3 3 4

0 0 0 0

g.out

5

6

0

g.in

4 4

XXXX

.X..

..X.

X...

1 1 3 1

0 0 0 0

g.out

4

**Варіант 9.**

Дано N квадратів на площині (1 < N < 100), причому сторони квадратів паралельні осям координат. Визначити площу області, яку визначають ці квадрати в сукупності (точка належить цій області, якщо вона є хоча б в одному з квадратів).

Формат вхідних даних

У першому рядку вхідного файлу є N.

У наступних N рядках знаходяться трійки чисел (x, y, a), кожна з яких описує один квадрат таким чином: (x, y) – координати центру квадрата, a – довжина його сторони. Усі числа – цілі. -30000 < x, y < 30000. 0 < a < 1000.

Формат вихідних даних У вихідний файл слід вивести площу області, покритої даними квадратами. Округлити значення до сотих і вивести рівно два знаки після десяткової точки.

Приклад введення

3

0 0 2

1 1 2

-3 1 5

Правильний висновок для прикладу

32.00

**Варіант 10.**

В одній державі є N міст. Деякі міста з'єднані дорогами, причому для будь-яких двох міст A і B виконується така властивість: існує рівно один спосіб потрапити з міста A в місто B якщо можна переміщатися тільки дорогами і не дозволяється проїжджати однією і тією ж дорогою більше одного разу.

Нещодавно президента цієї країни зацікавило питання: які три міста є найвіддаленішими одне від одного. А саме, назвемо взаємною віддаленістю один від одного трьох міст A, B і C мінімальну кількість доріг, яку необхідно використовувати, щоб доїхати від A до B, потім від B до C і потім від C до A (при цьому дозволяється використовувати одну й ту ж дорогу у різних подорожах).

Потрібно знайти три міста, для яких взаємна віддаленість одна від одної буде максимальною.

Наприклад, для п'яти міст, з'єднаних дорогами так, як це показано на малюнку 1, три найбільш віддалені один від одного міста - це міста 1, 2 і 5 (взаємна віддаленість дорівнює 2 + 3 + 3 = 8), а для міст на малюнку 2 - це будь-які три міста, вибрані з множини {1, 2, 4, 5} (віддаленість 2 + 2 + 2 = 6).



Формат вхідних даних

Перший рядок вхідного файлу містить число N - кількість міст (3<=N<=1000). Наступні N рядків містять опис міст. Опис i-го міста спочатку містить Ki - кількість міст, з якими він з'єднаний дорогами (1 <= Ki < N), а потім Ki чисел - номери міст, з якими він з'єднаний.

Гарантується, що вхідні дані коректні - тобто якщо є дорога з міста A до міста B, тобто дорога з міста B до міста A, причому для всіх пар міст виконується властивість, зазначена в умові завдання.

Формат вихідних даних

Виведіть у вихідний файл три різні числа - номери трьох найбільш віддалених один від одного міст у довільному порядку. Якщо рішень кілька, виведіть будь-яке з них.

Приклади

three.in

5

1 3

1 3

3 1 2 4

2 3 5

1 4

three.out

1 2 5

three.in

5

1 3

1 3

4 1 2 4 5

1 3

1 3

three.out

1 2 4

**Варіант 11.**

Держава Флатландія є прямокутником розміром M \* N, що складається з одиничних квадратиків. Флатландія розділена на K провінцій (2<=K<=100). Кожна провінція є зв'язкове безліч квадратиків, тобто. з кожної точки провінції можна дійти до будь-якої іншої точки, при цьому дозволяється переходити з квадратика на квадратик, тільки якщо вони мають спільну сторону (загальної вершини недостатньо). У Флатландії немає точки, яка межувала б з більш ніж трьома провінціями (тобто чотири квадратики, що мають спільну вершину, не можуть належати чотирма різним провінціям).

Кожна провінція має власний знак. Столиця Флатландії знаходиться в провінції, якій належить символ A (велика латинська літера A). Провінція називається прикордонною, якщо вона містить граничні квадратики. Провінція, де знаходиться столиця Флатландії, не є прикордонною.

Король сусіднього з Флатландією королівства Ректиланія вирішив завоювати Флатландію. І тому він хоче захопити столицю Флатландії. Однак він знає, що сил його армії недостатньо, щоб зробити це одразу. Тому спочатку він хоче оточити столичну провінцію, щоб послабити сили супротивника довгою блокадою, а потім захопити столицю.

Щоб оточити провінцію, потрібно захопити всі провінції, з якими вона межує. Дві провінції межують, якщо існує два квадратики, що мають спільну сторону, один з яких належить першій з них, а другий - другий. Щоб захопити провінцію, треба щоб виконувалася одна з двох умов: або вона прикордонна, або межує з якоюсь уже захопленою провінцією.

Щоб зберегти сили своєї армії, король Ректиланії хоче встановити блокаду столичної провінції, захопивши якнайменше провінцій. Допоможіть йому дізнатися, скільки провінцій потрібно захопити. Захоплювати столичну провінцію не можна, оскільки для цього сил армії Ректиланії поки що недостатньо.

Формат вхідних даних

Перший рядок містить M та N (3 <= M, N <= 200). Наступні M рядків містять N символів кожен і задають карту Флатландії. Символ, що знаходиться в i+1-му рядку вхідного файлу на j-му місці, є символом провінції, якій належить квадратик (i, j). Всі символи мають ASCII-код більше 32 (пробілу).

Формат вихідних даних

Виведіть у вихідний файл однину - кількість провінцій, які потрібно захопити. Якщо блокаду встановити неможливо, виведіть "-1".

Приклади

input.txt

5 6

BBBBBZ

BCCCBZ

BCAbbZ

BDDDbZ

ЗЗЗЗЗZ

output.txt

4

**Варіант 12.**

Сподіваємось, що вам знайома наступна гра. Вибирається слово і з його літер складаються інші осмислені слова, при цьому кожна з літер вихідного слова може бути використана не більше за таку кількість разів, яку вона в ньому зустрічається.

Напишіть програму, яка допомагає грати в цю гру.

Вхідні дані

У першому рядку вхідного файлу записано вибране для гри слово. У наступних рядках задано "словник" - безліч слів, які ми вважаємо осмисленими. Їхня кількість не перевищує 1000. Слово - це послідовність не більше ніж 255 маленьких латинських літер. Кожне слово записано в окремому рядку. Закінчується словник словом "ZZZZZZ" (що складається з великих літер), яке ми вважатимемо осмисленим. Слова в словнику, хоч як це дивно, необов'язково розташовуються за абеткою.

Вихідні дані

У вихідний файл потрібно видати список слів, які можна отримати з вихідного слова. Слова мають бути видані в тому самому порядку, в якому вони зустрічаються у словнику. Кожне слово має бути записане в окремому рядку. Список слів повинен закінчуватися тим самим неосмисленим словом "ZZZZZZ".

Приклади вхідного та вихідного файлів

C.IN

Soundblaster

Sound

Sound

Blaster

Soundblaster

Master

Last

Task

Sos

Test

Bonus

Done

ZZZZZZ

C.OUT

Sound

Sound

Blaster

Soundblaster

Last

Sos

Bonus

Done

ZZZZZZ

C.IN

Windowsmustdie

Summer

Informatics

School

Rules

Olympiadisstarting

Goodbymylovegoodby

Exit

ZZZZZZ

C.OUT

ZZZZZZ

**Варіант 13.**

Користувач мережі Інтернет підписаний на кілька різних списків розсилки, які надсилають йому електронною поштою повідомлення на певні теми. Для зручності користувач створив набір папок, кожна з яких відповідає одній з тем. Перед тим, як читати повідомлення, він копіює їх у відповідну папку.

Поштова програма, встановлена ​​на комп'ютері користувача, дозволяє за одну "операцію" переносити з "списку нових повідомлень" у відповідну папку:

- Одне повідомлення з будь-якого місця списку

- Декілька повідомлень, що йдуть у списку поспіль, і які стосуються однієї теми

Переносити можна не обов'язково починаючи з початку "списку нових повідомлень". Користувачу необхідно перенести нові повідомлення у відповідні їм папки за найменшу кількість операцій.

приклад. Нехай користувач підписаний на розсилки анекдотів, веселих історій, спортивних новин та прогнозу погоди. Нехай "список нових повідомлень" при деякому входженні до поштової програми мав такий вигляд: (Анекдоти, Спортивні новини, Прогноз погоди, Спортивні новини, Веселі історії, Веселі історії, Спортивні новини)

Список папок Список нових повідомлень

1 Анекдоти 1 Анекдоти

2 Веселі історії 3 Спортивні новини

3 Спортивні новини 4 Прогноз погоди

4 Прогноз погоди 3 Спортивні новини

2 Веселі історії

2 Веселі історії

3 Спортивні новини

Перенести повідомлення в папки може наступним чином: спочатку два повідомлення на тему "Веселі історії". Тоді він отримає наступний список: (Анекдоти, Спортивні новини, Прогноз погоди, Спортивні новини, Спортивні новини). Потім перенести повідомлення про прогноз погоди, після цього "Анекдоти", а потім одночасно всі три повідомлення про спортивні новини. На це він витратить 4 "операції".

Завдання: Напишіть програму EMAIL яка обчислюватиме мінімальну кількість "операцій", за допомогою яких можна перенести нові повідомлення в папки. Для зручності кожній темі надається номер.

Єдиний рядок вхідного файлу EMAIL.DAT містить число N (0 < N < 200), що відповідає кількості нових повідомлень і N цілих чисел, які відповідають повідомленням, і є номерами тим, яким ці повідомлення належать.

Приклад вхідних даних

7 1 3 4 3 2 2 3

Вихідні дані У першому рядку вихідного файлу EMAIL.SOL має бути мінімальна кількість операцій для даних, наведених у вхідному файлі.

Приклад вихідних даних

4

**Варіант 14.**

Тривимірна фігура складається із одиничних кубиків. За фігурою можна побудувати її фронтальну та праву проекції. Очевидно, що за цими двома проекціями не завжди можна відновити фігуру.

Завдання

Напишіть програму CUBES, яка отримує на вхід фронтальну та праву проекції фігури та визначає мінімальну та максимальну кількість кубиків, яку можна було б використовувати для побудови фігури із заданими проекціями.

Вхідні дані

У першому рядку вхідного файлу CUBES.DAT знаходяться три числа N, M та К, які задають розміри проекцій (1 < N, M, K < 100). Далі задаються дві проекції: спочатку передня, а потім права. Проекція задається N рядками, кожна з яких складається з чисел 0 та 1, розділених пробілами. Для фронтальної проекції таких чисел буде M, а правої - K. 0 означає вільну клітину проекції, 1 - заповнену.

Приклад вхідних даних

2 2 3

1 0

1 1

0 0 1

1 1 1

Вихідні дані

В єдиному рядку вихідного файлу CUBES.SOL має бути два числа: мінімальна і максимальна кількість кубиків, які можна було б використовувати для побудови фігури із заданими проекціями.

Приклад вихідних даних

4 7

**Варіант 15.**

У МНС надійшло повідомлення про можливу лісову пожежу у заданому квадраті тайги. Для пошуку місця займання було надіслано N літаків. Проте жоден із екіпажів пожежі не виявив.

Відомо, що з літака видно смугу тайги, межі якої знаходяться на відстані 50 км праворуч і ліворуч від лінії на поверхні Землі, над якою пролітає літак (див. малюнок), причому точки, що знаходяться на відстані рівно 50 км від цієї лінії, все ще видно. Донесення з кожного літака містило інформацію про те, в яких двох різних точках (xb, yb) і (xe, ye) літак входив у заданий квадрат і залишав його відповідно. Між цими точками літак рухався строго прямою.



Рис. 5

Потрібно

Написати програму, яка визначить, чи справді весь заданий квадрат тайги було переглянуто з літаків. Якщо це не так, то програма повинна знаходити координати якоїсь точки, що лежить усередині або на межі квадрата і не потрапила в жодну з переглянутих смуг.

Вхідні дані

Вхідний файл з ім'ям FIRE.IN складається з N+2 рядків.

У першому рядку записано натуральне число L – розмір заданого квадрата тайги в кілометрах (0 < L <= 1000). У другому рядку – натуральне число N (1<=N<=100) – кількість літаків. У кожному з наступних N рядків записано донесення з літака - чотири речові координати xb, yb, xe, ye. Координати задані за кілометри. Сторони квадрата тайги паралельні осям координат, його нижній лівий кут знаходиться в точці з координатами (0, 0), а правий верхній - в точці (L, L).

Вихідні дані

Вихідний файл з ім'ям FIRE.OUT має містити один рядок. Якщо заданий квадрат був переглянутий повністю, цей рядок повинен складатися зі слова OK, написаного великими латинськими літерами. В іншому випадку в цьому рядку повинні бути записані через пропуск координати x і y будь-якої точки, яка не потрапила в жодну з переглянутих смуг. Координати потрібно виводити за кілометри з помилкою не більше одного метра.

Приклад вхідного файлу

245

1

26.1 0 193.568 245

Приклад вихідного файлу для прикладу прикладу вхідного файлу

155.123 100

Примітка. Окремо оцінюватимуться рішення, зроблені у припущенні, що кожен літак летів паралельно до однієї зі сторін квадрата.