

Тестуються всі!

У школі Містера Бограча сталася нова хвиля епідемії COVID-19. Для запобігання подальшому поширенню захворювання, школа вирішила протестувати всіх учнів за допомогою антигенних тестів зі зразками слини учнів.

Оскільки всі вчителі вже давно забули, як їх використовувати, Містер Бограч записався добровольцем, щоб допомогти з тестуванням.

Містер Бограч отримав зразки слини від N учнів (через Політику конфіденційності він може знати лише ідентифікатори від 0 до $N - 1$) і його завданням є визначити, які зразки є позитивними.

На жаль, коли Містер Бограч зрозумів, що протестувати усіх учнів – це дуже довго і нудно, зупинятися було вже надто пізно. Однак він подумав, що може провести тестування більш ефективним способом, ніж послідовне тестування кожного зразка (один за одним). Якщо Містер Бограч змішає підмножину зразків і протестує цей змішаний зразок, він зможе визначити, чи всі зразки в цьому змішаному варіанті є негативними, інакше хоча б один з них є позитивним. Це дозволить йому зменшити кількість тестів, які потрібно проводити!

Оскільки в кожному зразку достатньо слини, Містер Бограч може тестувати кожен зразок стільки разів, скільки завгодно. Крім того, тести абсолютно точні, тому ніколи не відбувається так, що різні тести дають різні результати для однієї й тієї ж людини.

За таких умов він хоче оптимізувати процес так, щоб використовувати якомога менше тестів. Однак він зайнятий тестуванням учнів, тому оптимізація процесу залишається за вами.

Із досвіду, він знає, що ймовірність позитивного тесту на COVID дорівнює P , і що позитивність або негативність одного зразка не впливає на позитивність або негативність будь-якого іншого зразка. Можливо, ви можете використати це для оптимізації проведення тестів?

Формат взаємодії

Ця задача є інтерактивною.

Ваша програма буде виконуватись на кількох тестових випадках. У рамках одного тестового випадку, тобто під час одного виконання вашої програми, вам потрібно буде вирішити T

різних сценаріїв.

Значення N та P однакові для всіх сценаріїв, але те, які учні мають позитивний тест, ймовірно, відрізнятимуться у кожному сценарії.

Ви можете реалізувати необхідний протокол взаємодії самостійно або скористатися наданим шаблоном. Шаблон можна знайти в CMS як вкладення до задачі під назвою `template.cpp`.

Протокол взаємодії

Спочатку ваша програма повинна зчитати рядок із стандартного вводу, який містить ціле число N , дійсне число P і ціле число T , розділені пробілами — кількість людей, ймовірність позитивного результату і кількість сценаріїв.

Після цього програма може виводити запити у стандартний вивід. Кожен запит повинен бути одним рядком, що містить Q , пробіл і рядок s довжиною N , де s_i дорівнює 1, якщо Містер Бограч має додати i -ту людину до тесту, і 0 в іншому випадку. Після виведення цього рядка програма повинна очистити стандартний вивід і після цього зчитати один символ в одному рядку, який буде R , якщо принаймні одна людина з тестової групи позитивна, і N в іншому випадку.

Додатково програма може вивести відповідь одним рядком у стандартний вивід, що містить A , пробіл і рядок s довжиною N , де s_i дорівнює 1, якщо i -ий учень має позитивний результат тесту, і 0 в іншому випадку. Після виведення цього рядка програма повинна очистити стандартний вивід і після цього зчитати один символ в одному рядку.

Якщо рядок містить C , то ваша відповідь була правильною. У цьому випадку програма може почати виконувати запити щодо наступного сценарію, або, якщо це була остання відповідь (тобто T -та), вийти.

Якщо рядок містить W , це означає, що ваша відповідь не була правильною. У цьому випадку програма повинна негайно завершити свою роботу.

Зауважте, що при вихід після отримання відповіді W є важливим для коректного вердикту системи CMS. Якщо програма не завершиться, що вона може отримати непередбачуваний результат.

Шаблон

Якщо ви використовуєте шаблон `template.cpp`, то вам потрібно реалізувати функцію `std::vector<bool> find_positive()`. Ця функція буде визвана один раз для кожного сценарію. Вона повинна повертати вектор Булевих значень довжини N , де i -ий елемент рівний 1 якщо i -ий учень отримав позитивний результат тесту.

Аби виконати це, використайте функцію `bool test_students(std::vector<bool> mask)`. Ця функція проводить тест на підмножині зразків слини учнів. Єдиним її аргументом є вектор Булевих значень довжини N , де i -ий елемент рівний 1 якщо ми повинні додати i -ий зразок слини в суміш. Функція поверне значення `true` тоді і тільки тоді, коли принаймні один з використаних зразків слини є зразком з позитивним тестом.

Ви можете використовувати глобальні змінні N і P які містять в собі значення N і P з умови. Ви можете виконувати будь яку ініціалізацію в функції `main` після першого виклику `scanf`.

Формат вхідних даних

Інтерактор до цієї задачі не є адаптивним, що означає що результат тесту кожного учня є визначений до початку запитів. Результат тесту *кожного* учня визначений незалежно за допомогою генератора випадкових чисел.

Підзадача і оцінювання

Всього є дві підзадачі

Перша підзадача (10 балів)

- $N = 1\,000$
- $T = 1$
- $0 \leq P \leq 1$

Рішення буде прийняте якщо воно використовує не більше ніж $2 \cdot N$ запитів для одного тесту.

Друга підзадача (90 балів)

- $N = 1\,000$
- $T = 300$
- $0.001 \leq P \leq 0.2$

Ця підзадача використовує часткове оцінювання.

Якщо ваша відповідь на будь-який сценарій неправильна, ви отримаєте 0 балів. В іншому випадку, кількість балів за кожний тест буде визначатися на основі середньої кількості запитів на сценарій. Загалом, менше запитів означає більше балів.

Позначимо через Q середню кількість запитів, використаних вашою програмою для всіх сценаріїв, округлену до однієї десяткової цифри. Для кожного тестувального випадку ми обчислили значення F (див. нижче). Оцінка за кожний тест буде розрахована за наступними правилами:

- Якщо $Q > 10 \cdot F$ ви отримаєте 0 балів.
- Якщо $F < Q \leq 10 \cdot F$, то кількість балів буде розрахована за допомогою наступної формули:

$$90 \cdot \frac{F}{F + 4 \cdot (Q - F)}$$

- Якщо $Q \leq F$, ви отримаєте всі 90 балів.

Ваше рішення буде оцінене на декількох тестових випадках з різними значеннями P . Загальна кількість балів, яку ви отримаєте, буде мінімальною кількістю балів серед усіх тестових випадків (тобто для всіх ймовірностей P).

Значення P та F на яких буде протестоване ваше рішення:

P	F
0.001	15.1
0.005256	51.1
0.011546	94.9
0.028545	191.5
0.039856	246.3
0.068648	366.2
0.104571	490.3
0.158765	639.1
0.2	731.4

Система оцінювання надасть інформацію для кожного тестового випадку. Ця інформація буде включати значення Q вашого рішення для кожного тестового випадку, де ви отримаєте ненульовий бал.

Приклад взаємодії

Ось як може виглядати зразок взаємодії з інтерактором. Не забувайте очищувати буфер виводу після кожного рядку. Зверніть увагу, що значення N і T не можуть з'являтися в жодній підзадачі.

Ваш ввід	Ваш вивід
10 0.4 2	
	Q 1000000000
P	
	Q 0000001000
P	
	Q 0000000001
P	
	Q 0111110110
N	
	A 1000001001
C	
	A 0000000000
W	

Програма правильно вирішила перший сценарій, але неправильно другий, оскільки правильне рішення було 1100010010 (яке програма не могла знати, оскільки не виконувала жодних запитів). Навіть якщо є ще один сценарій, програма повинна негайно завершити роботу.