

Testy COVID

Przez szkołę Adama przechodzi fala zachorowań na COVID. Aby zapobiec rozprzestrzenianiu się choroby, szkoła przetestuje uczniów nowymi, niezawodnymi testami. Wystarczą do tego próbki śliny uczniów.

Nauczyciele są na bakier z nowymi technologiami i nie umieją przeprowadzić testów. Adam zaoferował swoją pomoc.

Adam otrzymał próbki śliny N uczniów. Z uwagi na RODO, zna tylko ich zanonimizowane numery (od 0 do $N - 1$). Jego zadaniem jest określić, które z próbek zawierają wirusa. Z lokalnych statystyk pandemicznych wie, że prawdopodobieństwo zachorowania każdego z uczniów to P oraz że prawdopodobieństwa te są niezależne (w sekcji Dane pojawi się formalny opis). Testowanie każdego z uczniów pojedynczo to powolny i nudny proces. Z drugiej strony, można przecież mieszać próbki śliny! Wykonując taki mieszany test, Adam dowie się, że albo wszystkie próbki są zdrowe, albo choć jedna z nich zawiera wirusa. Może wykorzystać te fakty, aby oszczędzić sobie pracy!

Każda próbka jest obfita w ślinę – można ją wykorzystać do testów dowolną liczbę razy. Testy są niezawodne i zawsze poprawnie określają występowanie nawet najmniejszego stężenia wirusa.

Adam chce zoptymalizować proces testowania próbek tak, aby wykonać możliwie najmniej testów. Niestety jest bardzo zapracowany, więc to Ty musisz opracować jego strategię.

Komunikacja

Jest to zadanie interaktywne.

Każdy z testów wymaga rozwiązania T osobnych przypadków. Wartości N i P są stałe w obrębie jednego testu, ale numery zarażonych uczniów będą (prawdopodobnie) inne.

Możesz zaimplementować protokół samodzielnie lub użyć dostarczonego szkicu. Szkic znajduje się w systemie jako załącznik `template.cpp`.

Protokół

Najpierw Twój program powinien wczytać linię standardowego wejścia, zawierającą liczbę naturalną N , liczbę rzeczywistą P oraz liczbę naturalną T , oddzielone spacjami – liczba uczniów,

prawdopodobieństwo zarażenia wirusem oraz liczba przypadków testowych.

Następnie program może zadawać pytania poprzez standardowe wyjście. Każde pytanie powinno być pojedynczą linią, zawierającą: znak Q , spację oraz ciąg znaków s długości N , gdzie s_i określa, czy ślina i -tej osoby ma się znaleźć w teście. Znak 1 oznacza, że tak; znak 0, że nie. Po wypisaniu takiej linijki, program powinien zadbać o odświeżenie (flush) standardowego wyjścia, a następnie odczytać odpowiedź w formie pojedynczej linii złożonej ze znaku P , jeśli przynajmniej jedna z testowanych próbek śliny zawiera wirusa; w przeciwnym wypadku linia złożona będzie ze znaku N .

Program może również w podobny sposób przekazać rozwiązanie przypadku testowego. Rozwiązanie powinno być pojedynczą linią, zawierającą: znak A , spację oraz ciąg znaków s długości N , gdzie s_i określa, czy próbka nr i zawiera COVID. Znak 1 oznacza, że tak; znak 0, że nie. Po wypisaniu takiej linijki program powinien zadbać o odświeżenie (flush) standardowego wyjścia, a następnie odczytać ocenę rozwiązania w formie pojedynczej linii złożonej ze znaku C , jeśli rozwiązanie jest poprawne, lub znaku W , jeśli jest błędne. Jeśli rozwiązanie było poprawne, można natychmiast rozpocząć rozwiązywanie kolejnego przypadku testowego, lub zakończyć program, jeśli był to ostatni przypadek testowy. W przypadku przekazania błędnego rozwiązania i otrzymania odpowiedzi W należy zakończyć program, aby system pokazał właściwą informację zwrotną. W przeciwnym wypadku program sprawdzający może pokazać inny kod błędu.

Szkic

Jeśli korzystasz z implementacji protokołu z pliku `template.cpp`, musisz zaimplementować funkcję `std::vector<bool> find_positive()`. Funkcja ta wywołana jest raz dla każdego przypadku testowego. Funkcja powinna zwracać wektor wartości logicznych (Boolean) długości N , gdzie wartość na indeksie i określa, czy program uznał próbkę śliny nr i za zawierającą wirusa.

Możesz używać funkcji `bool test_students(std::vector<bool> mask)`, aby przetestować miks próbek śliny. Jej jedynym argumentem jest wektor wartości logicznych o długości N , gdzie wartość na indeksie i określa, czy próbka śliny nr i powinna znaleźć się w testowanej mieszance. Funkcja zwróci `true` wtedy i tylko wtedy, gdy przynajmniej jedna z próbek śliny w mieszance zawiera wirusa.

Dozwolone jest też używanie zmiennych globalnych N i P , zawierających wartości N i P opisane w treści zadania. Inicjalizację dodatkowych zmiennych należy wykonać w funkcji `main`, po pierwszym wywołaniu `scanf`.

Dane

Program sprawdzający nie jest adaptacyjny – zawartość wirusa w próbkach jest ustalona przed uruchomieniem Twojego programu. Decyzja dla każdej próbki została podjęta niezależnie, z prawdopodobieństwem P , przy pomocy uczciwego generatora liczb pseudolosowych.

Podzadania i ocenianie

Są dwa podzadania.

Podzadanie pierwsze (10 punktów)

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

Aby otrzymać pełne 10 punktów, należy użyć co najwyżej $2 \cdot N$ pytań na każdym przypadku testowym.

Podzadanie drugie (90 punktów)

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

W tym podzadaniu można otrzymać punkty częściowe.

Jeśli rozwiązanie jest błędne, otrzymasz zero punktów. W przeciwnym wypadku liczba punktów za test jest zależna od średniej liczby pytań per przypadek testowy. Prostymi słowami: im mniej pytań, tym więcej punktów. Niech Q to średnia liczba pytań zadanych przez program na danym teście, zaokrąglona do jednego miejsca po przecinku. Dla każdego testu autorzy zadania określili wartość F (tabela poniżej). Wynik na teście będzie obliczony wg następujących zasad:

- Jeśli $Q > 10 \cdot F$, otrzymasz 0 punktów (system zwróci WA / błędna odpowiedź).
- Jeśli $F < Q \leq 10 \cdot F$, liczba punktów będzie obliczona następującym wzorem:

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

- Jeśli $Q \leq F$, otrzymasz pełne 90 punktów.

Rozwiązanie będzie oceniane na testach o różnych wartościach P . Ostateczna punktacja będzie zależna od najmniejszego wyniku na indywidualnym teście.

Oto wartości P i F dla poszczególnych testów:

| 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 |

System sprawdzający udzieli informacji zwrotnej dla każdego testu. Informacja zwrotna będzie zawierała wartość Q dla każdego testu, na którym Twój program otrzyma niezerową liczbę punktów.

Przykładowa interakcja

Oto przykładowa interakcja programu ze sprawdzaczką. Podane w przykładzie wartości N i T nie wystąpią w prawdziwych podzadaniach. Nie zapomnij o odświeżaniu (flush) standardowego wyjścia po każdej linii.

| Wejście | Twoje wyjście |
|----------|---------------|
| 10 0.4 2 | |
| | Q 1000000000 |
| P | |
| | Q 0000001000 |
| P | |
| | Q 0000000001 |
| P | |
| | Q 0111110110 |
| N | |
| | A 1000001001 |
| C | |
| | A 0000000000 |
| W | |

Program rozwiązał poprawnie pierwszy przypadek, ale nie drugi. Poprawnym rozwiązaniem było 1100010010, (czego program nie mógł wiedzieć, bo nie zadał ani jednego pytania). Nawet gdyby w teście występowały kolejne przypadki testowe, program powinien natychmiast zakończyć działanie.