

COVID tesztek

Ádám iskolájában a COVID-járvány újabb hulláma indult el. A további terjedés megakadályozására az iskola úgy döntött, hogy minden diákot tesztelnek antigén tesztekkel a tanulók nyálmintáit használva.

Mivel a tanárok már régen elfelejtették, hogyan kell ezeket használni, Ádám önként jelentkezett, hogy segít a tesztek elvégzésében.

Ádám N diák nyálmintáját kapta meg (adatvédelmi okokból csak a 0-tól $N - 1$ -ig terjedő azonosítókat ismerheti) és az ő feladata, hogy meghatározza, mely minták pozitívak. Sajnos már túl későn jött rá, hogy az összes nyálminta tesztelése rendkívül hosszú és unalmas feladat. Kitalálta azonban, hogy a tesztelési folyamatot a minták egyesével történő tesztelésénél okosabb módon is elvégezheti. Ha a minták egy részhalmazát összekeveri, és ezt a keveréket teszteli, megtudja, hogy ebben a keverékben az összes minta negatív-e, vagy legalább egy pozitív közülük. Ezt felhasználhatja arra, hogy csökkentse a szükséges tesztelések számát.

Minden egyes mintában elegendő nyál van, így annyiszor vizsgálhat egy mintát, ahányszor csak akar. Ráadásul a tesztek abszolút pontosak, így soha nem fordulhat elő, hogy a különböző tesztek különböző eredményeket adnak ugyanarra a mintára.

Ilyen körülmények között szeretné optimalizálni a folyamatot, hogy minél kevesebb tesztet használjon. A teszteléssel azonban el van foglalva, így a folyamat optimalizálása a te feladatod.

A helyi statisztikákból Ádám kiderítette már, hogy P annak a valószínűsége, hogy egy adott minta pozitív, és hogy az egyik minta pozitív vagy negatív voltát nem befolyásolja más minták pozitív vagy negatív volta. Talán ezt az információt fel tudod használni annak optimalizálásához, hogy milyen méréseket végezzen?

Kommunikáció

Ez a feladat interaktív.

A programodat számos tesztesetre fogjuk lefuttatni. Egy teszteset részeként, tehát a programod egy végrehajtása során, T különböző feladványt kell megoldanod. Az N és a P értéke minden feladványban ugyanaz, de az, hogy mely tanulók mintái pozitívak, (valószínűleg) minden feladványban más és más lesz.

A szükséges protokollt vagy te magad implementálhatod, vagy használhatod a megadott sablont. A sablont a CMS-ben találod meg a feladat mellékleteként `template.cpp` néven.

A protokoll

Először is a programodnak be kell olvasnia egy sort a standard bemenetről, amely tartalmazza az N egész számot, a P valós számot és a T egész számot, szóközzel elválasztva - a diákok számát, a pozitív minta valószínűségét és a feladványok számát.

Ezután a program kiírhatja a lekérdezéseket a standard kimenetre. Minden egyes lekérdezésnek egyetlen sornak kell lennie, amely tartalmaz egy `Q` karaktert, majd egy szóközt és egy N hosszúságú s karakterláncot, ahol s_i értéke 1 , ha Ádám az i -edik diák mintáját hozzáadja a teszthez, és 0 , ha nem. Miután kiírta ezt a sort, a programnak flush-olnia kell a kimenetet, majd be kell olvasnia egy sort egyetlen karakterrel, amely `P` lesz, ha a vizsgált csoportból legalább egy minta pozitív, egyébként pedig `N`.

A program a választ is egyetlen sorban írhatja ki a standard kimenetre, amely egy `A` karaktert, egy szóközt és egy N hosszúságú s karakterláncot tartalmaz, ahol az s_i értéke 1 , ha az i -edik diák mintája pozitív, és 0 , ha negatív. Miután kiírta ezt a sort, a programnak flush-olnia a standard kimenetet, majd olvasson be egy sort egyetlen karakterrel.

Ha a sorban `C` szerepel, akkor a válaszod helyes volt. Ebben az esetben a program elkezdheti a következő feladványra vonatkozó lekérdezések végrehajtását, vagy, ha ez volt a T -edik válasz, akkor le kell állnia.

Ha a sorban `W` szerepel, akkor a válaszod nem volt helyes. Ebben az esetben a programnak azonnal le kell állnia.

Vedd figyelembe, hogy a `W` után történő leállás fontos ahhoz, hogy helyes visszajelzést kapj a CMS-től. Ha a program tovább fut, akkor összeomolhat, vagy más sikertelen visszajelzést kaphat.

A sablon

Ha a `template.cpp`-ben található protokoll implementációt használod, akkor implementálnod kell az `std::vector<bool> find_positive()` függvényt. Ez a függvény minden egyes feladványra egyszer hívódik meg. Egy N hosszúságú logikai vektort kell visszaadnia, ahol az i -edik elem akkor és csak akkor `true` (igaz), ha az i -edik tanuló mintája pozitív.

Ehhez felhasználhatod a `bool test_students(std::vector<bool> mask)` függvényt. Ez a függvény a minták egy részhalmazán végez tesztet. Egyetlen argumentuma egy N hosszúságú logikai vektor, ahol az i -edik elem igaz, ha az i -edik mintát hozzá kell adni a keverékhez. Akkor (és csak akkor) ad vissza `true` (igaz) értéket, ha a keverékben lévő minták közül legalább egy pozitív.

A feladatléírásban említett N és P értékeket tartalmazó N és P globális változókat is használhatod. A `main` függvényben az első `scanf` hívás után bármilyen inicializálást elvégezhetsz.

Bemenetek

Az értékelőprogram nem adaptív, ami azt jelenti, hogy az egyes minták pozitivitását a program futtatása előtt határozzuk meg. Továbbá, minden mintára egymástól függetlenül P valószínűséggel döntöttünk úgy, hogy pozitív-e, egy egyenletes véletlenszám-generátor segítségével.

Részfeladatok és értékelés

Két részfeladat van.

Első részfeladat (10 pont)

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

Egy megoldás akkor fogadható el, ha helyesen válaszol, és minden tesztesetre legfeljebb $2 \cdot N$ lekérdezést használ.

Második részfeladat (90 pont)

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

Ez a részfeladat részleges pontozást használ.

Ha bármelyik feladványra rosszul válaszol a program, nulla pontot kap. Ellenkező esetben az adott tesztesetre kapott pontok számát a feladványonkénti lekérdezések átlagos száma alapján határozzuk meg. Általában a kisebb számú lekérdezés nagyobb pontszámot eredményez. Legyen Q a program által az összes feladványra vonatkozóan a használt lekérdezések átlagos száma, egy tizedesjegyre kerekítve. Minden egyes tesztesethez kiszámítottunk egy F értéket (lásd alább). Egy adott teszteset pontszámát a következő szabályok szerint számítjuk ki:

- Ha $Q > 10 \cdot F$, akkor 0 pontot kapsz (rossz válasz).
- Ha $F < Q \leq 10 \cdot F$, a pontok számát a következő képlet alapján határozzuk meg:

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

- Ha $Q \leq F$, akkor a teljes 90 pontot kapod.

A megoldásodat több, a P különböző értékeit tartalmazó teszteset alapján értékeljük. Az összesített pontszámot az összes teszteset (azaz az összes P valószínűség) minimális pontszáma adja.

A következő tesztesetek vannak:

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

Az értékelőrendszer minden tesztesethez visszajelzést ad. Ez a visszajelzés tartalmazza a megoldás Q értékét minden olyan tesztesetnél, ahol nem nulla pontszámot kapott.

Minta interakció

Íme egy minta interakció az értékelővel. Megjegyezzük, hogy ilyen N és T értékek nem fordulhatnak elő egyik részfeladatban sem. Ne felejtse el minden sor után flush-olni a kimenetet.

A bemeneted	A kimeneted
10 0.4 2	
	Q 1000000000
P	
	Q 0000001000
P	
	Q 0000000001
P	
	Q 0111110110
N	
	A 1000001001
C	
	A 0000000000
W	

A program az első feladványt helyesen oldotta meg, de a másodikat nem, mivel a helyes megoldás a 1100010010 volt (amit a program nem tudhatott, mivel nem végzett lekérdezést). Még ha lenne is egy másik feladvány, a programnak azonnal le kellene állnia.