

Testy na COVID

V Adamově škole se začal znovu šířit COVID, a proto se rozhodli, že všechny studenty otestují pomocí testů ze slin. Jelikož učitelé už zapomněli, jak se testy používají, požádali Adama, aby se o to postaral. Adam dostal vzorky slin od N studentů. Kvůli ochraně soukromí jsou vzorky označené pouze identifikačními čísly studentů od 0 do $N - 1$. Jeho úlohou je určit, které ze vzorků jsou pozitivní. Bohužel až pozdě si Adam všiml, že testovat všechny vzorky bude velmi pomalé a nudné. Rád by si proto práci usnadnil.

Adam si dále uvědomil, že může být trochu chytřejší a nemusí vzorky testovat jeden po druhém: Pokud pečlivě smíchá i jen malou část z více vzorků dohromady a otestuje výslednou směs, zjistí, zda všechny ze smíchaných vzorků byly negativní, nebo alespoň jeden z nich je pozitivní. Díky tomu by mohl snížit počet testů, které musí udělat! Naštěstí má všech vzorků slin dostatečně velké množství, takže je může testovat, kolikrát chce. Testy jsou navíc zcela přesné, nestane se mu tedy, že by dostal chybnou odpověď.

Adam by za těchto podmínek samozřejmě chtěl určit, kteří studenti jsou COVID-pozitivní, s použitím co nejméně testů. Jelikož ho ale plně vytěžuje provádění testů, je na vás, abyste mu s optimalizací pomohli.

Z informací od místních statistiků víme, že každý student má pravděpodobnost přesně P , že je COVID-pozitivní. Dále také víme, že to, zda někteří studenti jsou COVID-pozitivní či negativní, nijak neovlivňuje pravděpodobnosti, zda jiní studenti jsou COVID-pozitivní či negativní. Tj. jevy "student 0 je COVID-pozitivní", "student 1 je COVID-pozitivní", ..., "student $N - 1$ je COVID-pozitivní" jsou navzájem nezávislé. Možná těchto faktů můžete využít k optimalizaci počtu testů?

Komunikace s vyhodnocovačem

Tato úloha je interaktivní.

Váš program bude vyhodnocen na několika testovacích vstupech. V rámci každého testovacího vstupu (tj. během jednoho běhu vašeho programu) musí vyřešit T různých scénářů. Všechny tyto scénáře mají stejné hodnoty N a P , ale COVID-pozitivní studenti mohou být v každém z nich jiní.

Níže popsany protokol můžete implementovat sami, nebo k tomu můžete použít poskytnutý vzor. Ten naleznete v CMS jako přílohu `template.cpp`.

Protokol

Váš program musí nejprve ze standardního vstupu načíst jednu řádku obsahující kladné celé číslo N , reálné číslo P ($0 \leq P \leq 1$) a kladné celé číslo T . Tato čísla jsou oddělená mezerami a označují počet studentů, pravděpodobnost, že student je COVID-pozitivní, a počet scénářů.

Poté váš program může vypisovat dotazy na standardní výstup. Každý dotaz je jedna řádka obsahující písmeno `Q`, mezeru a řetězec s délky N takový, že $s[i]$ je `1`, jestliže Adam má vzorek studenta číslo i přidat do testované směsi, a `0` jinak. Po vypsání této řádky musí váš program vyprázdnit buffer standardního výstupu (flush) a přečíst jednu řádku ze standardního vstupu. Na této řádce bude jediný znak, `P` jestliže je výsledek testu pozitivní a `N` jinak.

Odpověď pro daný scénář váš program oznámí vypsáním řádky obsahující písmeno `A`, mezeru a řetězec s délky N na standardní výstup. Zde $s[i]$ má být `1`, jestliže student číslo i je COVID-pozitivní, a `0` jinak. Po vypsání této řádky program musí vyprázdnit buffer standardního výstupu a přečíst ze standardního vstupu jednu řádku obsahující jediné písmeno. Je-li toto písmeno `C`, vaše odpověď byla správná. Program může začít pokládat dotazy k dalšímu scénáři, či skončit v případě, že jste právě vypsali odpověď na T -tý scénář.

Jinak tato řádka obsahuje písmeno `W`. Vaše odpověď v tomto případě byla chybná a program by měl okamžitě skončit. To je důležité, abyste od CMS dostali správné vyhodnocení; jestliže program poběží dále, můžete od CMS dostat jinou než očekávanou chybu.

Vzorová implementace protokolu

Použijete-li předpřipravenou vzorovou implementaci protokolu v `template.cpp`, musíte implementovat funkci `std::vector<bool> find_positive()`. Tato funkce bude volána právě jednou pro každý scénář a musí vrátit pole pravdivostních hodnot délky N , jehož prvek na indexu i je `true`, právě když student číslo i je COVID-pozitivní.

V implementaci této funkce můžete používat funkci `bool test_students(std::vector<bool> mask)`. Tato funkce provede test na podmnožinu vzorků zadanou parametrem `mask`, což je pole pravdivostních hodnot délky N , jehož prvek na indexu i je `true`, právě když vzorek studenta číslo i má být přidán do testované směsi. Funkce vrátí `true`, právě když alespoň jeden ze vzorků v této směsi je od COVID-pozitivního studenta.

Můžete také používat globální proměnné `N` a `P`, jejichž význam je popsán v zadání. Do funkce `main` za první volání `scanf` můžete přidat libovolnou inicializaci, kterou má váš program provést pouze jednou.

Vstupy

Vyhodnocovač není adaptivní, tj. zda jsou studenti COVID-pozitivní se pevně určí před tím, než je váš program spuštěn. Pro každého studenta je určeno, zda je COVID-pozitivní, s pravděpodobností P nezávislým voláním náhodného generátoru.

Podúlohy a hodnocení

Jsou dvě podúlohy.

První podúloha (10 bodů)

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

Řešení je přijato, pokud je ve všech testovacích vstupech odpověď správná a vaše řešení v každém z nich použilo nejvýše $2 \cdot N$ dotazů.

Druhá podúloha (90 bodů)

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

V této podúloze můžete získat i jen část bodů.

Je-li vaše odpověď na libovolný scénář chybná, vaše řešení dostane 0 bodů. Jinak je počet bodů za testovací vstup určen na základě průměrného počtu dotazů v jeho scénářích; za méně dotazů můžete dostat více bodů. Necht' Q je průměrný počet dotazů ve všech scénářích, zaokrouhlený dolů na jednu číslici za desetinnou čárkou. Pro každý testovací vstup je určena hodnota F , specifikovaná níže. Počet bodů za testovací vstup je určen následovně:

- Jestliže $Q > 10 \cdot F$, obdržíte 0 bodů a odpověď je považována za chybnou.
- Jestliže $F < Q \leq 10 \cdot F$, počet bodů je určen vzorcem

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

- Jestliže $Q \leq F$, za tento testovací vstup obdrží vaše řešení všech 90 bodů.

Vaše řešení bude vyhodnoceno na více testovacích vstupech s různými hodnotami P a počet bodů, který celkově obdržíte, je minimum z počtu bodů za každý z nich (určený dle výše popsanych pravidel).

Testovací vstupy a jejich hodnoty P a F udává následující tabulka:

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

Vyhodnoč pro tuto úlohu poskytuje zvlášť informaci ke každému z těchto testovacích vstupů, dozvíte se tedy hodnotu Q pro každý testovací vstup, v němž vaše řešení obdrží nenulový počet bodů.

Příklad interakce

Zde je příklad možné interakce s vyhodnocovačem; nezapomeňte po vypsání každé řádky vyprázdnit buffer standardního výstupu. Poznámka: Vstup s těmito hodnotami N a T se nemůže vyskytnout v žádné podúloze.

Čtení vstupu	Výpis na výstup
10 0.4 2	
	Q 1000000000
P	
	Q 0000001000
P	
	Q 0000000001
P	
	Q 0111110110
N	
	A 1000001001
C	
	A 0000000000
W	

Program vyřešil správně první scénář, ale ne druhý. Správná odpověď na druhý scénář byla 1100010010, což program nemohl vědět, jelikož v něm nepoložil žádné dotazy. I kdyby v testovacím vstupu byl další scénář, váš program by měl okamžitě skončit.