COVID Tests

In Adam's Schule gibt es einen neuen Ausbruch der COVID Epidemie. Um Ansteckungen zu verhindern, hat die Schule beschlossen alle Schulkinder unter der Verwendung von Antigen Tests mit dem Speichel der Schulkinderzu testen.

Da alle Lehrpersonen vor langer Zeit vergessen haben, wie man diese benutzt, hat Adam sich als Freiwilliger gemeldet, um mit den Tests zu helfen.

Er hat Speichelproben von N Schulkindern bekommen (zum Schutze der Privatspähre, weiß er nur die Bezeichnungen von 0 bis $N-1$) und seine Aufgabe ist es zu ermitteln, welche Proben positiv sind. Leider war es zu spät, als er realisiert hat, dass alle Schulkinder zu testen eine lange und mühsame Arbeit ist. Jedoch hat er realisiert, dass er den Testprozess klüger durchführen kann, als alle Proben nacheinander zu testen. Wenn er eine Teilmenge von Proben mischt und diese testet, kann er herausfinden, ob alle Proben in der Mischung negativ sind oder ob mindestens eine davon positiv ist. Er könnte das benutzen, um die Anzahl der Tests die benötigt sind, zu reduzieren!

Da genügend Speichel in jeder Probe vorhanden ist, kann er jede Probe beliebig viele Male testen. Außerdem sind die Tests 100 Prozent präzise, es kommt also nicht vor, dass verschiedene Ausführungen der Tests für dieselbe Probe zu unterschiedlichen Ergebnissen führen.

Unter diesen Verhältnissen würde er gerne so wenige Tests wie möglich durchführen. Leider ist er aber zu beschäftigt mit dem Testen. Deswegen bist du für die Prozessoptimierung zuständig.

Aus örtlichen Statistiken konnte Adam die Wahrscheinlichkeit P herausfinden, dass eine beliebige Speichelprobe positiv ist. Außerdem ist die Wahrscheinlichkeit für jede Probe positiv zu sein genau P, unabhängig von den anderen Proben. Vielleicht hilft dir das beim Optimieren der Teststrategie.

Kommunikation

Die Aufgabe ist interaktiv.

Dein Program wird auf mehreren Testfällen ausgeführt. Innerhalb eines Testfalls muss dein Program T verschiedene Szenarien lösen. Der Wert von N und P wird zwischen allen T Szenarien in einem Testfall gleich bleiben, jedoch wird sich (sehr wahrscheinlich) ändern, welche Schulkinder positiv sind.

Du kannst entweder das folgende Protokoll selbst implementieren oder das bereitgestellte Template benutzen. Das Template der Aufgabe im CMS trägt den Namen `template.cpp`.

Protokoll

Als erstes soll dein Program eine Zeile von der Standardeingabe lesen, welches die ganze Zahl N, die reelle Zahl P und die ganze Zahl T beinhaltet, abgetrennt mit Leerzeichen — die Anzahl der Schulkinder, die Wahrscheinlichkeit einer positiven Probe und die Anzahl der Szenarien.

Nachher kann dein Program auf der Standardausgabe Abfragen stellen. Eine Abfrage soll aus einer einzelne Zeile bestehen: `Q`, ein Leerzeichen, und eine Zeichenkette s mit der Länge N, sodass s_i `1` ist, falls Adam den i-ten Lernenden zum Testpool hinzufügen soll, ansonsten `0`. Nach dieser Zeile soll dein Program die Standardausgabe flushen, und nacher einen Buchstaben von einer einzelnen Zeile lesen. Dieser Buchstabe ist `P`, falls mindestens ein Schulkind aus dem Testpool positiv ist, und `N` wenn alle negativ sind.

Schlussendlich kann dein Program eine Antwort einreichen - eine einzelne Zeile, mit: `A`, ein Leerzeichen, und eine Zeichenkette s mit der Länge N, so das s_i `1` ist, falls das `i`-te Schulkind positiv ist, und `0` wenn nicht. Nach dieser Zeile soll dein Program die Standardausgabe flushen und dann einen Buchstaben von einer einzelnen Zeile einlesen.

Falls dieser Buchstabe `C` ist, war deine Antwort korrekt. In diesem Fall kann dein Program anfangen Abfragen zum nächsten Szenario zu schicken, oder - wenn das der T-te Szenario war - das Program beenden.

Falls dieser Buchstaben `W` ist, war deine Antwort falsch. In diesem Fall soll dein Program sofort abbrechen.

Es ist wichtig nach `W` abzubrechen, um die richtige Rückmenldung vom CMS zu erhalten. Falls dein Program weiterläuft, könnte es abstürzen oder eine andere erfolglose Rückmeldung geben.

Das Template

Wenn du unsere Implementierung vom Protokoll im `template.cpp` benutzt, musst du die Funktion `std::vector<bool> find_positive()` implementieren. Diese Funktion wird einmal für jedes Szenario aufgerufen. Es muss einen Vektor mit boolschen Werten der Länge N ausgeben, in dem das i-te Element `true` ist, genau dann wenn das i-te Schulkind positiv ist.

Du kannst die Funktion `bool test_students(std::vector<bool> mask)` nutzen, um zu bestimmen, ob eine Mischung positiv ist. Diese Funktion führt einen Test auf einen Testpool aus. Es nimmt ein einziges Argument, ein Vektor mit Boolschen Werten der Länge N, in dem das i-te Element `true` ist, falls es zum Testpool hinzugefügt werden soll. Es gibt genau dann `true` zurück, falls mindestens ein Schulkind im Testpool positiv ist.

Du kannst auch die globalen Variabeln `N` und `P` benutzen, die N und P enthalten. Du darfst in der `main` Funktion nach dem ersten `scanf` Aufruf Initialisierungen vornehmen.

Eingaben

Der Grader der Aufgabe ist nicht adaptiv, das heißt, dass im Voraus bestimmt wird, welches Schulkind positiv ist und welches nicht. Außerdem wird für jedes Schulkind unabhängig von einander mit einem fairen Zufallsgenerator und einer Wahrscheinlichkeit von P bestimmt, ob es positiv ist.

Teilaufgaben und Bewertung

Es gibt zwei Teilaufgaben.

Die erste Teilaufgabe (10 Punkte)

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

Eine Lösung wird akzeptiert, falls es die korrekte Lösung ausgibt und höchstens $2 \cdot N$ Abfragen pro Szenario stellt.

Die zweite Teilaufgabe (90 Punkte)

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

Bei dieser Teilaufgabe gibt es Teilpunkte.

Falls deine Antwort zu einem Szenario falsch ist, erhältst du keine Punkte. Ansonsten wird die Anzahl der Punkte für einen beliebigen Testfall basierend auf der durchschnittlichen Anzahl an Abfragen pro Szenario bestimmt. Im Allgemeinen geben weniger Abfragen mehr Punkte. Bennenen wir die durchschnittliche Anzahl von Abfragen, abgerundet auf eine Dezimalstelle, mit Q. Für jeden Testfall wurde ein Wert F berechnet (siehe unten). Die Anzahl Punkte für einen gegebenen Testfall wird mit den folgenden Regeln bestimmt:

- Falls $Q > 10 \cdot F$ erhälst du keine Punkte (falsche Antwort).
- Falls $F < Q \leq 10 \cdot F$, wird die Anzahl Punkte mir der folgenden Formel bestimmt:

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

- Falls $Q \leq F$, erhälst du alle 90 Punkte.

Deine Lösung wird an mehreren Testfällen mit verschiedenen P Werten getestet. Deine schlussendliche Punktzahl ist die kleinste Punktzahl von allen Testfällen (also, von allen Wahrscheinlichkeiten P).

Es gibt die folgenden Testfälle:

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

Der Grader wird dir für jeden Testfall eine Rückmeldung geben. Diese Rückmeldung wird den Wert Q deiner Lösung zu jedem Testfall enthalten, welcher mehr als 0 Punkten gibt.

Beispielinteraktion

Hier ist eine Beispielinteraktion mit dem Grader. Vergiss nicht nach jeder Zeile zu flushen.

Eingabe	Ausgabe
`10 0.4 2`	
	`Q 1000000000`
`P`	
	`Q 0000001000`
`P`	
	`Q 0000000001`
`P`	
	`Q 0111110110`
`N`	
	`A 1000001001`
`C`	
	`A 0000000000`
`W`	

Das Program hat die erste Aufgabe korrekt gelöst, die Zweite aber nicht, da die richtige Lösung `1100010010` war. (was das Program nicht wissen konnte, da es keine Abfragen gestellt hat). Auch falls es eine weitere Aufgabe gäbe, muss das Program sofort abbrechen.