

Jaś stoi ostatni w kolejce do apteki. Ponieważ Jasiowi bardzo się śpieszy, to postanowił, że spróbuje się pozamieniać miejscami z niektórymi osobami, nawet jeśli musiałby za to zapłacić.

Każda osoba jest chętna do zamiany, ale i -tej osobie za przesunięcie o każde jedno miejsce dalej w kolejce trzeba zapłacić c_i . Dokładniej, jeśli Jaś jest k miejsc ($k > 0$) dalej od kasy niż pewna osoba i jeśli chce się z nią zamienić miejscami, to musi jej zapłacić kwotę $k \cdot c_i$.

Jaś chciałby być pierwszy w kolejce i zastanawia się, jak dokonywać zamian, aby wydać jak najmniej.

Wejście

Pierwszy wiersz standardowego wejścia zawiera jedną liczbę całkowitą n ($1 \leq n \leq 10^6$), oznaczającą liczbę osób, które stoją przed Jasiem w kolejce do apteki.

Następny wiersz wejścia zawiera n liczb całkowitych $c_1, c_2, \dots, c_n$ ($1 \leq c_i \leq 10^9$), gdzie c_i oznacza kwotę, jaką Jaś musi zapłacić i -tej osobie za przesunięcie o każde miejsce dalej w kolejce. Kolejność osób liczona jest od osoby, za którą bezpośrednio stoi Jaś, a więc od końca kolejki do jej początku.

Możesz założyć, że w przynajmniej 50% przypadków testowych zachodzi warunek: $n \leq 10\,000$.

Wyjście

Pierwszy i jedyny wiersz standardowego wyjścia powinien zawierać jedną liczbę całkowitą, równą minimalnej kwocie, jaką Jaś musi zapłacić, aby być pierwszym w kolejce.

Przykład

Dla danych wejściowych:

4
5 2 4 3

poprawnym wynikiem jest:

10

Wyjaśnienie do przykładu: Jaś zamieni się najpierw z 3 osobą w kolejce za kwotę $2 \cdot 2$, a następnie z pierwszą osobą w kolejce za kwotę $3 \cdot 2$.

Dwa słowa

VI OIG, etap I. Dostępna pamięć: 64 MB.

12.12.2011 - 23.01.2012

Marcin osiągnął już wiek, w którym zaczyna uczyć się słów. Dziwnym trafem (no, bo przecież nie w wyniku złośliwości rodziców) wszystkie słowa, jakie do tej pory, poznał są tej samej długości. Gdyby znał słowa różnych długości mógłby bawić się w jedną z wielu zabaw, o których słyszał od starszych kolegów. W tej chwili jednak nie pozostaje mu nic innego, jak zająć się jedyną możliwą rozrywką: zamianą liter.

Zabawa polega na tym, że Marcin na początku zapisuje dwa słowa, a potem wybiera po jednej literce z obu słów i... zamienia je miejscami. Po dokonanej zamianie pyta mamę, które słowo jest późniejsze leksykograficznie (sam nawet nie wie, co to słowo oznacza). Mama zawsze bezbłędnie odpowiada na to pytanie, a Marcin kontynuuje swoją zabawę ze zmienionymi już słowami.

Niestety, mama Marcina jest ostatnio bardzo zajęta przygotowywaniem potraw świątecznych. Na szczęście Marcin zawsze może liczyć na swoje starsze rodzeństwo. Zapewne domyślasz się już, że Marcin jest Twoim bratem. Pomóż mu i odpowiedz na jego pytania! Możesz też napisać program, który zrobi to za Ciebie.

Wejście

Pierwszy wiersz standardowego wejścia zawiera jedną liczbę całkowitą n ($1 \leq n \leq 10^6$), oznaczającą długość słów.

Następne dwa wiersze zawierają po jednym słowie długości n składające się wyłącznie z małych liter alfabetu angielskiego. Pozycje liter w słowie ponumerowane są od 0 do $n - 1$.

Następny wiersz zawiera jedną liczbę całkowitą t ($1 \leq t \leq 10^5$), oznaczającą liczbę pytań Marcina.

W kolejnych t wierszach znajdują się po dwie liczby całkowite a_i i b_i , mówiące, że przed i -tym pytaniem Marcin zamienił miejscami literę znajdującą się na pozycji a_i w pierwszym słowie i literę znajdującą się na pozycji b_i w drugim słowie.

Możesz założyć, że w przynajmniej 50% przypadków testowych zachodzi warunek: $n \leq 1\,000$.

Wyjście

Standardowe wyjście powinno zawierać dokładnie t wierszy. W każdym z nich powinna znajdować się odpowiedź na kolejne pytanie Marcina, zadane po zamianie liter: 0, jeśli po zamianie słowa są równe, 1, jeśli późniejsze leksykograficznie jest słowo pierwsze lub 2, jeśli późniejsze leksykograficznie jest słowo drugie.

Przykład

Dla danych wejściowych:

4
aaab
aaba
2
2 2
3 2

poprawnym wynikiem jest:

1
0

Wyjaśnienie do przykładu: Po pierwszej zamianie pierwsze słowo ma postać **aabb**, a drugie **aaaa**, więc późniejsze leksykograficznie jest słowo pierwsze. Po drugiej zamianie oba słowa mają postać **aaba**.

Podróżnik Bajtonson wrócił właśnie z kolejnej podróży w nieznane. Z oczywistych względów nie mógł wspomagać się mapą, więc notował jedynie, w którą stronę podróżuje. Dlatego zawsze po pokonaniu kilometra notował kierunek, w jakim się poruszał – N, S, E lub W (odpowiednio północ, południe, wschód lub zachód).

Jako że w notesie było niewiele miejsca, podróżnik korzystał ze skrótów np. 10 NSSW oznaczało, że Bajtonson 10 razy powtórzył sekwencję „1 km na północ, 2 km na południe, 1 km na zachód”. W miejscu, w którym kończył dany odcinek trasy (opisany skrótem), zaczynał następny.

Po powrocie do domu, podróżnik zapragnął narysować mapę swoich podróży. Zdecydował, że każdemu kilometrowi jego podróży będzie odpowiadał jeden centymetr na mapie. Teraz potrzebuje kupić odpowiedni arkusz papieru, nie jest jednak w stanie ocenić, jak dużego arkusza będzie potrzebował. Dlatego poprosił Ciebie, swojego asystenta, o napisanie programu, który przetworzy zapiski z podróży i wyznaczy wymiary najmniejszego (pod względem pola powierzchni) arkusza, na którym zmieści się trasa Bajtonsona.

Oczywiście — jak każda mapa — mapa Bajtonsona musi być prostokątem o bokach równoległych do osi północ-południe i wschód-zachód.

Wejście

Pierwszy wiersz standardowego wejścia zawiera jedną liczbę całkowitą n ($1 \leq n \leq 1000$). W n następnych wierszach opisane są kolejne etapy podróży Bajtonsona. W i -tym z tych wierszy znajduje się liczba całkowita k_i ($1 \leq k_i \leq 20\,000$) a po odstępnie niepusty ciąg znaków złożony z liter N, S, E i W. Taki zapis oznacza, że w ramach i -tego etapu Bajtonson k_i razy powtórzył podaną sekwencję kierunków. Liczba znaków N, S, E i W na wejściu nie przekroczy 1 000 000.

Możesz założyć, że w przynajmniej 50% przypadków testowych łączna trasa, jaką pokonał Bajtonson, jest nie dłuższa niż 50 000 000 km.

Wyjście

Twój program powinien wypisać na standardowe wyjście dwie liczby całkowite oddzielone pojedynczym odstępem: wysokość i szerokość najmniejszego arkusza, na którym zmieści się trasa podróżnika, wyrażone w centymetrach. Możesz założyć, że obie liczby będą nie większe niż 1 000 000 000.

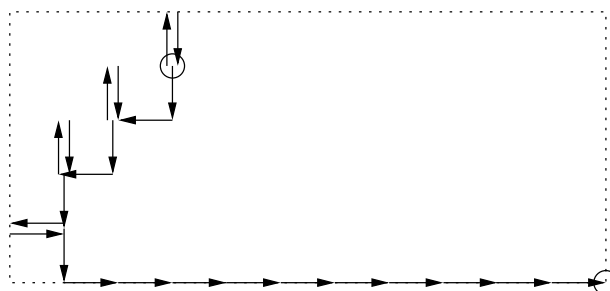
Przykład

Dla danych wejściowych:

```
3
3 NSSW
1 ES
10 E
```

poprawnym wynikiem jest:

```
5 11
```



Dzieci spędzają wakacje nad morzem. Obecnie kąpią się w wodzie. Nad ich bezpieczeństwem czuwa ratownik, którego zasięg wzroku wynosi k metrów. Wszystkie osoby znajdujące się dalej niż k metrów od ratownika są dla niego niewidoczne. Znając położenie ratownika i dzieci, oblicz ile z nich w razie niebezpieczeństwa nie otrzyma pomocy.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się liczby całkowite: n, k, x, y ($1 \leq n, k \leq 100\,000$, $0 \leq x, y \leq 10\,000$) oznaczające kolejno: liczbę dzieci, zasięg wzroku ratownika oraz jego położenie. W następnych n wierszach znajdują się wartości: x_i i y_i ($0 \leq x_i, y_i \leq 10\,000$) oznaczające współrzędne i -tego dziecka (kąpielisko zostało opisane na układzie współrzędnych).

Wyjście

W pierwszym wierszu standardowego wyjścia należy wypisać liczbę dzieci, które nie mogą czuć się bezpiecznie.

Przykład

Dla danych wejściowych:

5 3 2 2
2 4
2 6
3 3
4 2
5 6

poprawnym wynikiem jest:

2

Na lekcji śpiewu uczniowie Bajtazara siedzą w jednym, długim rzędzie. Nie wszystkie krzesła są w nim zajęte i pomiędzy poszczególnymi uczniami mogą występować wolne miejsca.

Na lekcji uczniowie potrzebują śpiewników, ale nie każdy musi trzymać śpiewnik. Nauczyciel musi się zatroszczyć tylko o to, aby każdy uczeń bez śpiewnika siedział bezpośrednio obok ucznia ze śpiewnikiem. Ponieważ uczniowie na każdej lekcji siadają w inny sposób, a śpiewników jest dosyć mało, Bajtazar poprosił Ciebie, swojego przyjaciela, o napisanie programu, który dla danego rozmieszczenia uczniów wyznaczy minimalną liczbę potrzebnych im śpiewników, aby ułatwić Bajtazarowi rozdawanie śpiewników.

Wejście

W pierwszym wierszu standardowego wejścia znajduje się jedna liczba naturalna n ($1 \leq n \leq 1\,000\,000$) oznaczająca liczbę miejsc w rzędzie. W drugim wierszu znajduje się ciąg n znaków opisujących kolejne miejsca:

- znak „W” oznacza miejsce wolne,
- znak „Z” oznacza miejsce zajęte przez ucznia.

Możesz założyć, że w przynajmniej 50% przypadków testowych zachodzi warunek: $n \leq 2\,000$.

Wyjście

Twój program powinien wypisać na wyjściu jedną liczbę całkowitą oznaczającą minimalną liczbę śpiewników, które można rozdać uczniom tak, aby każdy miał śpiewnik lub siedział obok kogoś ze śpiewnikiem.

Przykład

Dla danych wejściowych:

9

ZWZZZWZZW

poprawnym wynikiem jest:

3