A – Akronim

Limit pamięci: 1024 MB
Limit czasu: 2 s

AMPPZ 2024
2024-11-17

Już jutro imieniny Agnieszki, która uwielbia sekrety i zabawy językowe. Jej przyjaciółka Ania ozdabia właśnie kartkę z życzeniami, ale chciałaby jeszcze zawrzeć zdanie (ciąg słów), którego pierwsze litery tworzą ukryte hasło. Taką konkatenację (sklejenie) pierwszych liter nazywamy akronimem.

Ania wymyśliła już następujące przykłady zdań na tę okazję:

- `SUKCESU TOTALNEGO ORAZ` o akronimie `STO`.
- `LICZNYCH ATRAKCJI TOBIE` o akronimie `LAT`.
- `AKRONIM NA IMIENINY AGI` o akronimie `ANIA`.
- `TORT ORAZ RABARBAROWY TORT` o akronimie `TORT`.

Ostatni przykład pokazuje, że słowa w zdaniu mogą się powtarzać; ponadto dopuszczamy użycie akronimu w zdaniu.

Ania musi zdecydować się na jakiś akronim, więc poprosiła Cię o pomoc. Dany jest słownik czyli zbiór n różnych dozwolonych słów. Sprawdź, czy da się z nich zbudować zdanie, którego pierwsze litery również tworzą słowo należące do słownika. Podaj przykład takiego zdania lub wypisz `-1` jeśli jest to niemożliwe.

Wejście

Pierwszy wiersz zawiera liczbę całkowitą n ($1 \le n \le 200$), oznaczającą liczbę słów w słowniku.

Każdy z kolejnych n wierszy zawiera jedno słowo s_i złożone z co najmniej 2 i co najwyżej 8 wielkich liter alfabetu angielskiego `A-Z`. Słowa na wejściu są parami różne.

Wyjście

Jeśli szukane zdanie nie istnieje, w jedynym wierszu wyjścia wypisz liczbę `-1`.

W przeciwnym przypadku w pierwszym wierszu wypisz liczbę słów w znalezionym zdaniu, a w drugim wierszu wypisz to zdanie w postaci ciągu słów rozdzielonych spacjami.

Jeśli istnieje wiele możliwych zdań spełniających warunki zadania, wypisz jedno dowolne z nich.

Przykład

Dla danych wejściowych:

```
5
ANIA
IMIENINY
AGI
AKRONIM
NA
```

jednym z poprawnych wyników jest:

```
4
AKRONIM NA IMIENINY AGI
```

Natomiast dla danych wejściowych:

```
1
XX
```

poprawnym wynikiem jest:

```
2
XX XX
```

Wyjaśnienie przykładów:

W pierwszym teście przykładowym można wypisać wiele innych zdań, na przykład `NA ANIA` albo `ANIA NA IMIENINY ANIA`. Natomiast niepoprawne są zdania `AKRONIM NA IMIENINY ANI` oraz `AKRONIM NA IMIENINY`, gdyż w danym słowniku nie ma słowa `ANI`.

W drugim teście przykładowym zdanie `XX XX` jest jedyną poprawną odpowiedzią.

ORGANIZATORZY
UNIWERSYTET WARSZAWSKI
FUNDACJA ROZWOJU INFORMATYKI
PARTNER
IDEAS NCBR
DIAMENTOWY SPONSOR
HUAWEI
ZŁOCI SPONSORZY
DIGITAL TECHNOLOGY POLAND
ElevenLabs
Jane Street
neptune.ai
SPONSORZY
ATENDE
Google
intel

B – Bagaże

Limit pamięci: 1024 MB
Limit czasu: 3 s

AMPPZ 2024
2024-11-17

Bartek jest zapalonym podróżnikiem i co roku odwiedza nowe kraje i kontynenty. Nie byłoby to możliwe, gdyby nie optymalizował kosztów podróży, szczególnie że podróżuje z dwiema walizkami. W ramach oszczędności jest gotów przechowywać bagaż u swoich znajomych oraz nocować w hotelach opłacanych przez organizatorów lokalnych konkursów programistycznych.

Na świecie jest n miast (ponumerowanych od 1 do n) oraz m połączeń lotniczych. Każde połączenie lotnicze jest opisane czwórką liczb (a, b, c, d), oznaczającą jednokierunkowy lot z miasta a do miasta b, o koszcie c i z limitem d walizek. Bartek może użyć tego samego lotu wiele razy, za każdym razem przewożąc co najwyżej d walizek. Cena nie zależy od ilości bagażu. Walizka nie może sama lecieć samolotem, ale w każdym mieście Bartek ma znajomego, który może przechować walizkę lub walizki dowolnie długo i dowolnie wiele razy.

Bartek ma dwie walizki i chciałby przedostać się między dwoma miastami. Dla każdej z n^2 par miast (x, y) znajdź najmniejszy łączny koszt biletów, jeśli Bartek zaczyna w mieście x z dwiema walizkami i chce się z nimi znaleźć w mieście y. Wypisz `-1` jeśli dla danej pary miast nie jest to możliwe.

Wejście

Pierwszy wiersz zawiera dwie liczby całkowite n i m ($1 \le n \le 400$; $0 \le m \le 500\,000$), oznaczające liczbę miast i liczbę połączeń lotniczych.

Każdy z kolejnych m wierszy zawiera cztery liczby całkowite a_i, b_i, c_i, d_i ($1 \le a_i, b_i \le n$; $a_i \ne b_i$; $1 \le c_i \le 10^9$; $0 \le d_i \le 2$), opisujące jedno połączenie lotnicze. Z jednego miasta do drugiego może istnieć wiele połączeń. Może się zdarzyć, że z jakiegoś miasta lub do jakiegoś miasta nie ma żadnych połączeń.

Wyjście

Wypisz n wierszy, każdy z n liczbami całkowitymi oddzielonymi spacjami. W i-tym wierszu j-ta liczba to najmniejszy łączny koszt przedostania się z dwiema walizkami z miasta i do miasta j, lub liczba `-1` jeśli nie jest to możliwe.

Przykład

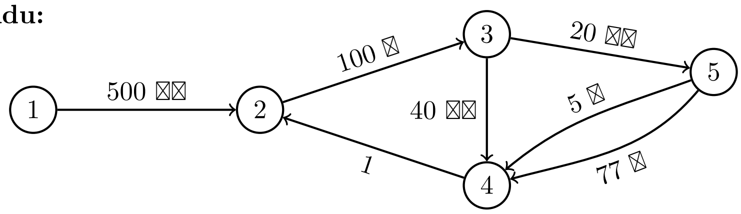
Dla danych wejściowych:

```
5 7
1 2 500 2
2 3 100 1
3 5 20 2
5 4 5 1
4 2 1 0
3 4 40 2
5 4 77 1
```

poprawnym wynikiem jest:

```
0 500 726 751 746
-1 0 226 251 246
-1 -1 0 40 20
-1 -1 -1 0 -1
-1 -1 -1 131 0
```

Wyjaśnienie przykładu:



Przykładowo, z miasta 1 do miasta 3 Bartek może przedostać się w następujący sposób:

$1 \to 2$ (zostaw walizkę) $\to 3$ (zostaw walizkę) $\to 5 \to 4 \to 2$ (odbierz walizkę) $\to 3$ (odbierz walizkę)

Łączny koszt to $500 + 100 + 20 + 5 + 1 + 100 = 726$.

ORGANIZATORZY
UNIWERSYTET WARSZAWSKI
FUNDACJA ROZWOJU INFORMATYKI
PARTNER
IDEAS NCBR
DIAMENTOWY SPONSOR
HUAWEI
ZŁOCI SPONSORZY
DIGITAL TECHNOLOGY POLAND
IIElevenLabs
Jane Street
neptune.ai
SPONSORZY
ATENDE
Google
intel

C – Cierpliwe krowy

Limit pamięci: 1024 MB
Limit czasu: 4 s

AMPPZ 2024
2024-11-17

Podłużne pastwisko podzielone jest na n segmentów, ponumerowanych kolejno od 1 do n. Początkowa wysokość trawy w i-tym segmencie wynosi h_i. W każdym segmencie pasie się jedna krowa. Segmenty są od siebie odgrodzone. Krowy nie mogą między nimi przechodzić, ale są w stanie wystawić łeb przez ogrodzenie i jeść trawę z sąsiedniego segmentu (i sąsiaduje z $i+1$ dla każdego i od 1 do $n-1$).

W każdej minucie każda krowa je trawę z jednego wybranego segmentu:

- Jeśli w jej segmencie jest jeszcze trawa ($h_i > 0$), to krowa zawsze wybiera swój segment.
- W przeciwnym przypadku krowa wybiera jeden z sąsiednich segmentów z trawą.
- Krowa nic nie robi, jeśli nie ma trawy w jej segmencie ani w sąsiednich segmentach.

Jeśli x krów je trawę z jednego segmentu, to w ciągu minuty skracają ją o x, przy czym wysokość nie spada poniżej zera. Czyli po minucie mamy $h_i := \max(0, h_i - x)$.

Krowy współpracują, by jak najszybciej zjeść całą trawę z pastwiska. Po której minucie może im się to udać?

Wejście

Pierwszy wiersz zawiera liczbę całkowitą n ($1 \le n \le 200\,000$), oznaczającą liczbę segmentów pastwiska.

Drugi wiersz zawiera n liczb całkowitych $h_1, h_2, \ldots, h_n$ ($0 \le h_i \le 10^9$), oznaczających początkowe wysokości trawy w kolejnych segmentach. Przynajmniej jedna z liczb h_i jest dodatnia.

Wyjście

Wypisz jedną liczbę całkowitą – minimalny czas zjedzenia całej trawy, w minutach.

Przykład

Dla danych wejściowych:

```
5
5 4 0 4 6
```

poprawnym wynikiem jest:

```
4
```

Natomiast dla danych wejściowych:

```
3
1 4 6
```

poprawnym wynikiem jest:

```
5
```

Wyjaśnienie przykładów:

W pierwszym teście przykładowym optymalna jest następująca strategia:

- $[5, 4, 0, 4, 6]$ – Krowa 3 wybiera sąsiedni segment 4. Pozostałe krowy jedzą w swoich segmentach.
- $[4, 3, 0, 2, 5]$ – Krowa 3 wybiera segment 4.
- $[3, 2, 0, 0, 4]$ – Krowa 3 wybiera segment 2, a krowa 4 segment 5.
- $[2, 0, 0, 0, 2]$ – Krowa 3 nic nie robi; krowa 2 wybiera segment 1; krowa 4 wybiera segment 5.
- $[0, 0, 0, 0, 0]$ – Krowy zjadły całą trawę w 4 minuty.

W drugim teście przykładowym krowy nie mają nigdy wyboru. Proces musi przebiegać następująco:

$$[1, 4, 6] \to [0, 3, 5] \to [0, 1, 4] \to [0, 0, 3] \to [0, 0, 1] \to [0, 0, 0]$$

ORGANIZATORZY
UNIWERSYTET WARSZAWSKI
FUNDACJA ROZWOJU INFORMATYKI
PARTNER
IDEAS NCBR
DIAMENTOWY SPONSOR
HUAWEI
ZŁOCI SPONSORZY
DIGITAL TECHNOLOGY POLAND
IIElevenLabs
Jane Street
neptune.ai
SPONSORZY
ATENDE
Google
intel

D – Dodawanie ułamków

Limit pamięci: 1024 MB
Limit czasu: 1 s

AMPPZ 2024
2024-11-17

Daria napisała program do arytmetyki liczb wymiernych. Program oblicza wynik wyrażenia składającego się z dodawania i odejmowania co najwyżej n ułamków, których liczniki i mianowniki to dodatnie liczby całkowite nie większe niż n.

Daria martwi się o wydajność programu, gdy wynikiem jest ogromna liczba lub dodatnia liczba bardzo bliska zeru. Tę pierwszą możliwość łatwo już przetestowała, wprowadzając oczywiście wyrażenie $\frac{n}{1} + \frac{n}{1} + \ldots + \frac{n}{1}$. Ale co z uzyskaniem bardzo małej liczby?

Dane jest t przypadków testowych, każdy z limitem n rozważanym przez Darię. Dla każdego danego n znajdź wyrażenie, którego wynik jest najmniejszą możliwą liczbą dodatnią.

Wejście

Pierwszy wiersz zawiera liczbę całkowitą t ($1 \leq t \leq 1000$), oznaczającą liczbę przypadków testowych.

Każdy z t kolejnych wierszy zawiera liczbę całkowitą n ($1 \leq n \leq 50\,000$).

Suma wartości n nie przekracza $4 \cdot 10^6 = 4\,000\,000$.

Wyjście

Dla każdego przypadku testowego wypisz jeden wiersz ze znalezionym wyrażeniem. Jeśli istnieje wiele rozwiązań, wypisz jedno dowolne z nich.

Wyrażenie powinno zawierać od 1 do n ułamków postaci a/b ($1 \leq a, b \leq n$). Ułamki oddzielone są znakami „`+`” lub „`-`”. Przed pierwszym ułamkiem może, ale nie musi, wystąpić znak „`-`” (ale nie może wystąpić znak „`+`”).

W wyrażeniu nie powinny występować spacje ani inne dodatkowe znaki.

Przykład

Dla danych wejściowych:

```
2
3
6
```

jednym z poprawnych wyników jest:

```
1/2-1/3
-3/6+1/4+2/5-5/6+6/4-4/5
```

Wyjaśnienie przykładu:

W pierwszym przypadku testowym mamy $n = 3$ i wyrażenie `1/2-1/3` $=$ `1/6`. Nie da uzyskać liczby dodatniej mniejszej niż `1/6`. Możliwe są też inne rozwiązania, na przykład `-1/3+1/2` lub `2/3-1/2` lub `-3/2+1/1+2/3`.

W drugim przypadku testowym mamy `-3/6+1/4+2/5-5/6+6/4-4/5` $=$ `1/60` dla $n = 6$.

ORGANIZATORZY
PARTNER
DIAMENTOWY SPONSOR
ZŁOCI SPONSORZY
SPONSORZY

E – Ekspresowe rotacje

Limit pamięci: **1024 MB**
Limit czasu: **3 s**

AMPPZ 2024
2024-11-17

amppz

Dany jest ciąg złożony z n liczb $a_1, a_2, \ldots, a_n$. Twoim zadaniem jest minimalnym kosztem usunąć wszystkie elementy w kolejności od największych; równe elementy można usuwać w dowolnej kolejności. Możesz wykonywać operacje trzech typów:

- Przenieś pierwszy element na koniec ciągu.
- Przenieś ostatni element na początek ciągu.
- Usuń pierwszy element, o ile w ciągu nie ma większego elementu.

Koszt przeniesienia jest równy wartości przenoszonego elementu. Usuwanie nic nie kosztuje. Znajdź najmniejszy możliwy łączny koszt uzyskania pustego ciągu.

Wejście

Pierwszy wiersz zawiera liczbę całkowitą n ($1 \leq n \leq 500\,000$).

Drugi wiersz zawiera n liczb całkowitych $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^6$).

Wyjście

Wypisz jedną liczbę całkowitą – minimalny łączny koszt.

Przykład

Dla danych wejściowych:

```
6
6 10 6 5 4 5
```

poprawnym wynikiem jest:

```
16
```

Wyjaśnienie przykładu: Optymalnie jest użyć następujących operacji:

- $(6, 10, 6, 5, 4, 5)$, przenieś pierwszy element na koniec. Koszt 6.
- $(10, 6, 5, 4, 5, 6)$, usuń pierwszy element.
- $(6, 5, 4, 5, 6)$, usuń pierwszy element.
- $(5, 4, 5, 6)$, przenieś ostatni element na początek. Koszt 6.
- $(6, 5, 4, 5)$, usuń pierwszy element.
- $(5, 4, 5)$, usuń pierwszy element.
- $(4, 5)$, przenieś pierwszy element na koniec. Koszt 4.
- $(5, 4)$, usuń pierwszy element.
- (4), usuń pierwszy element.
- $()$, ciąg jest pusty.

Łączny koszt to $6 + 6 + 4 = 16$.

ORGANIZATORZY
UNIWERSYTET WARSZAWSKI
FUNDACJA ROZWOJU INFORMATYKI
PARTNER
IDEAS NCBR
DIAMENTOWY SPONSOR
HUAWEI
ZŁOCI SPONSORZY
DIGITAL TECHNOLOGY POLAND
IIElevenLabs
Jane Street
neptune.ai
SPONSORZY
ATENDE
Google
intel

F – Fikające żaby

Limit pamięci: 1024 MB
Limit czasu: 2 s

AMPPZ 2024
2024-11-17

W bardzo długim i wąskim stawie żabia mama złożyła n jaj, i-te na pozycji a_i. Z każdego jaja wykluwa się jedna żaba, w kolejności od 1 do n. Dana jest też długość żabiego skoku k.

W którymś momencie pierwsze p wyklutych żab zacznie grać w skakanego berka. W berku każda żaba w nieskończoność goni swoją młodszą siostrę, a najmłodsza goni najstarszą (żaba i goni żabę $i+1$, p goni 1). Co sekundę każda żaba skacze w lewo lub prawo o k w kierunku gonionej siostry; w prawo w przypadku tej samej pozycji. Żaby od $p+1$ do n nie biorą udziału w zabawie. Formalnie, jednocześnie dla każdej z p żab: jeśli $a_{1+(i \bmod p)} \geq a_i$ to a_i zwiększa się o k, a w przeciwnym przypadku a_i zmniejsza się o k.

Żabia mama martwi się, że jej dzieci za bardzo się oddalą i wyskoczą poza staw. Niezależnie dla każdego p od 2 do n, sprawdź czy w skakanym berku żab $1, 2, \ldots, p$ którakolwiek z nich oddali się kiedykolwiek od swojej początkowej pozycji o co najmniej $99^{(99^{99})}$. Dla każdego p wypisz `1` jeśli tak się stanie, a `0` jeśli nie.

Wejście

Pierwszy wiersz zawiera dwie liczby całkowite n i k ($2 \leq n \leq 500\,000$; $1 \leq k \leq 10^9$), oznaczające liczbę jaj i długość skoku.

Drugi wiersz zawiera n liczb całkowitych $a_1, a_2, \ldots, a_n$ ($-10^9 \leq a_i \leq 10^9$), oznaczających pozycje jaj.

Wyjście

Wypisz bez spacji odpowiedzi dla każdego $p = 2, 3, \ldots, n$. Gdyby pierwsze p żab grało w nieskończoność w skakanego berka, to wypisz `1` jeśli któraś z nich oddali się od swojej początkowej pozycji o co najmniej $99^{(99^{99})}$, lub wypisz `0` w przeciwnym przypadku. Na wyjściu musi więc być słowo binarne o długości $n-1$.

Przykład

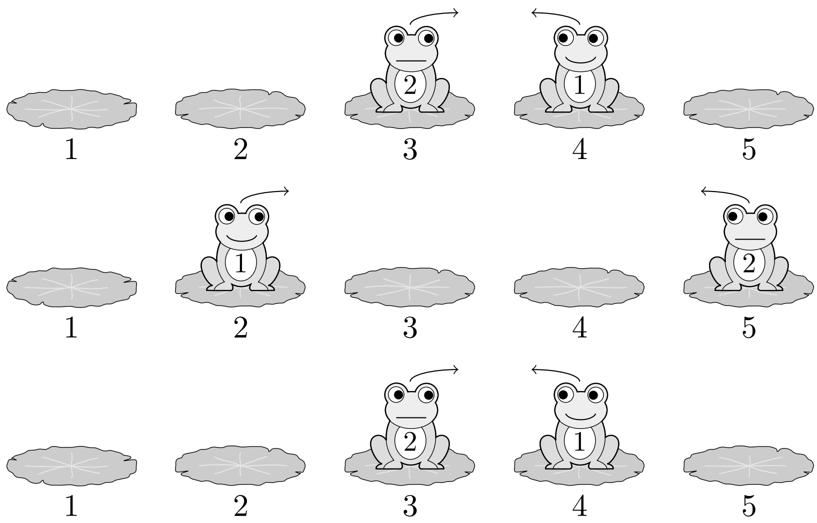
Dla danych wejściowych:

```
6 2
4 3 -3 5 100 100
```

poprawnym wynikiem jest:

```
01011
```

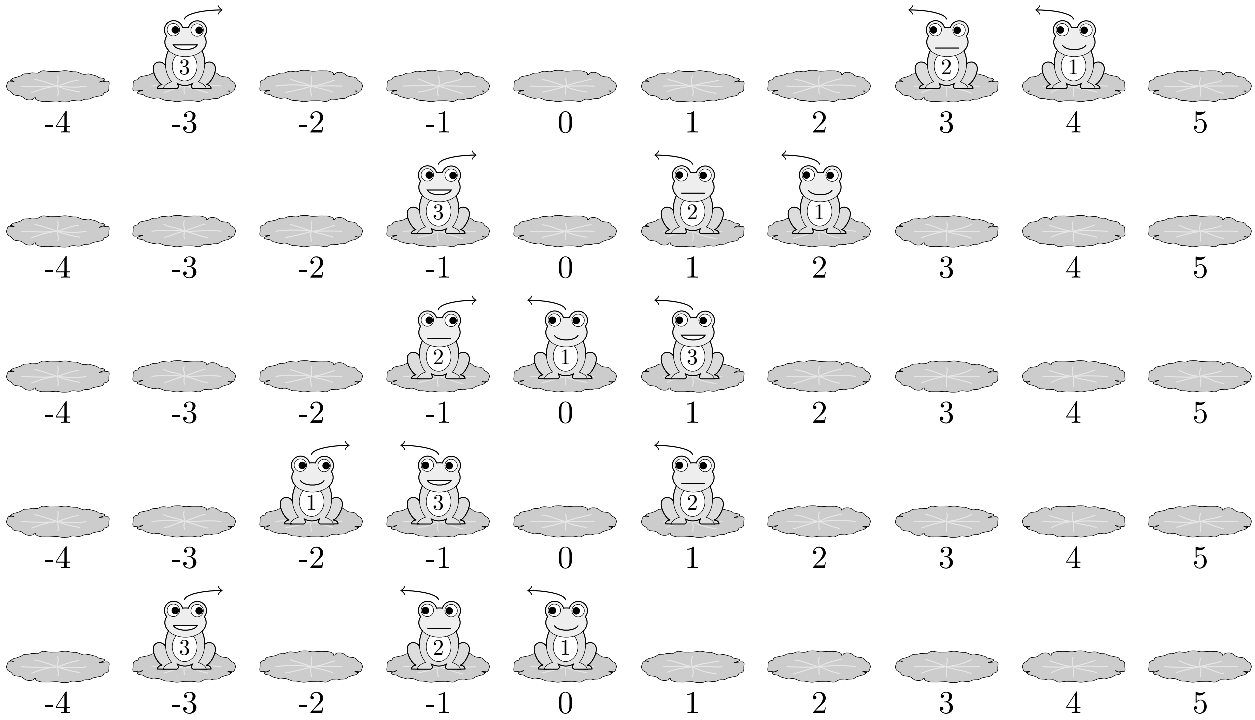
Wyjaśnienie przykładu: Poniższe rysunki (patrz też druga strona) pokazują pierwsze sekundy skakanego berka w scenariuszach $p = 2, 3, 4$. Dla $p = 2$ dwie żaby zaczynają na pozycjach 4 i 3, do których wracają co 2 sekundy. Żadna żaba nie oddali się bardzo od początkowej pozycji, więc odpowiedź to `0`.



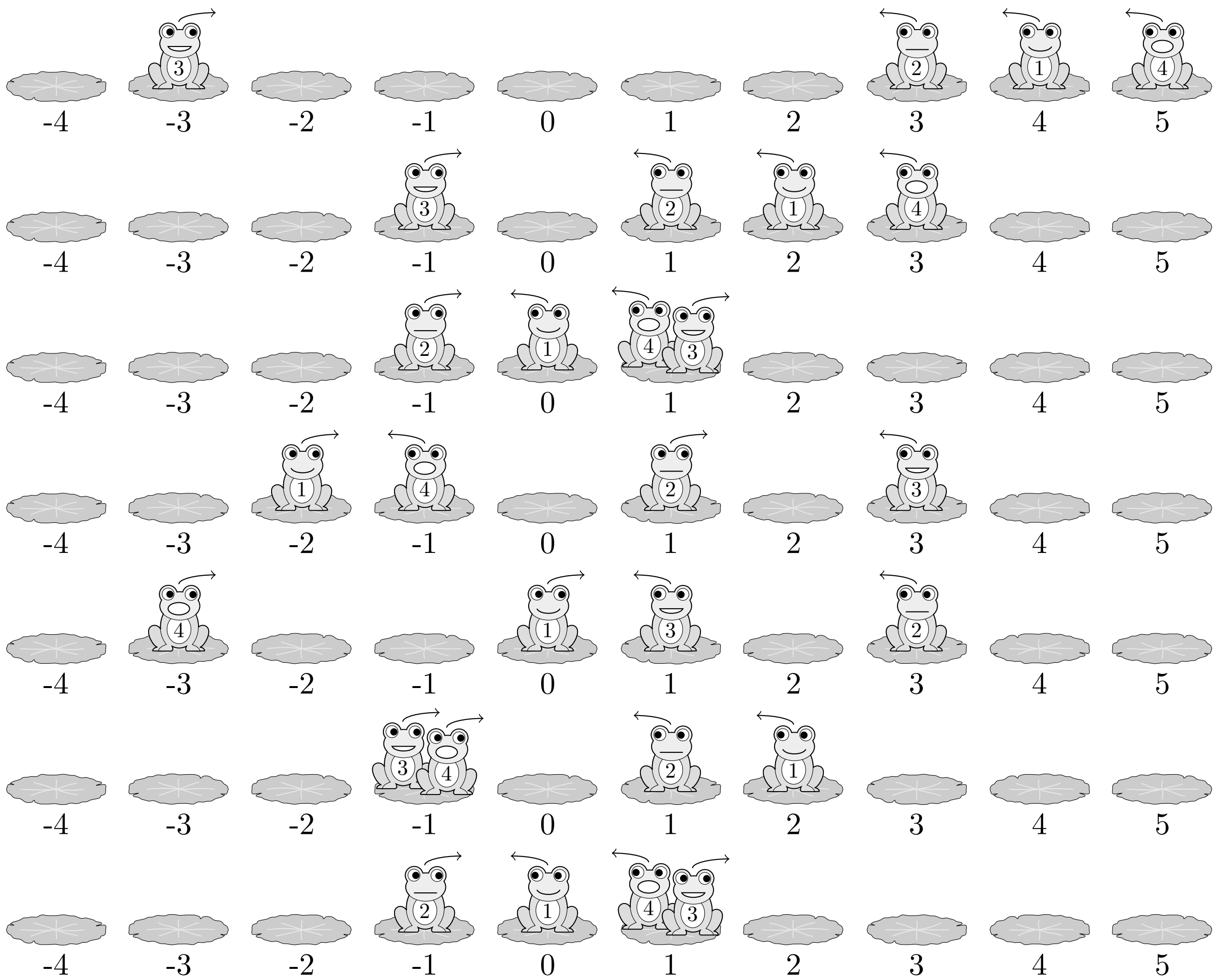
Rysunki dla $p = 3$ i $p = 4$ na następnej stronie!

ORGANIZATORZY
Uniwersytet Warszawski
Fundacja Rozwoju Informatyki
PARTNER
IDEAS NCBR
DIAMENTOWY SPONSOR
HUAWEI
ZŁOCI SPONSORZY
Digital Technology Poland
ElevenLabs
Jane Street
neptune.ai
SPONSORZY
ATENDE
Google
intel

Pierwsze kilka sekund dla $p = 3$ czyli $a = [4, 3, -3]$. Odpowiedź to `1`.



Pierwsze kilka sekund dla $p = 4$ czyli $a = [4, 3, -3, 5]$. Odpowiedź to `0`.



G – Gra MPO

Limit pamięci: 1024 MB
Limit czasu: 2 s

AMPPZ 2024
2024-11-17

W grze planszowej MPO otrzymuje się punkty za budowanie struktur na polach kwadratowej planszy $n \times n$. Dana jest początkowa plansza, a każde pole jest opisane jednym z 7 znaków „`.mpoMPO`”:

- Wielkie litery `M`, `P`, `O` oznaczają pole ze zbudowaną strukturą: odpowiednio miastem, parkiem lub oceanem.
- Małe litery `m`, `p`, `o` oznaczają puste pole, na którym można zbudować odpowiednio miasto, park lub ocean.
- Kropka '`.`' oznacza puste pole, na którym nie można nic zbudować.

Otrzymujesz punkt za każdą strukturę oraz za każdą stykającą się (bokiem lub rogiem) parę miasto-park oraz ocean-ocean. *Ruchem* jest zbudowanie struktury na dozwolonym polu, czyli zmiana małej litery na wielką `m` $\to$ `M`, `p` $\to$ `P` albo `o` $\to$ `O`. **Zakazane jest wykonanie ruchu, który daje tylko jeden punkt** (tzn. zwiększa liczbę punktów o dokładnie 1).

Twoja strategia jest prosta. Zawsze wykonujesz jeden z ruchów, które dają najwięcej punktów. Kończysz rozgrywkę, gdy nie możesz już wykonać ruchu – czyli nie ma możliwego ruchu, który dałby co najmniej dwa punkty.

Twoim zadaniem jest wyznaczyć początkową i końcową liczbę punktów, a także końcową planszę. Da się udowodnić się, że końcowy stan nie zależy od sposobu rozstrzygania remisów między ruchami dającymi tyle samo punktów.

Wejście

Pierwszy wiersz zawiera liczbę całkowitą n ($2 \leq n \leq 10$), oznaczającą rozmiar planszy.

Kolejne n wierszy opisuje początkową planszę. Każdy wiersz zawiera słowo składające się z n znaków „`.mpoMPO`”.

Wyjście

W pierwszym wierszu wypisz dwie liczby całkowite – początkową i końcową liczbę punktów.

Kolejne n wierszy powinny opisywać końcową planszę w formacie jak na wejściu.

Przykład

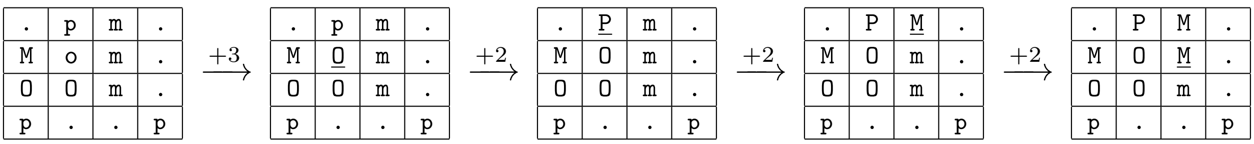
Dla danych wejściowych:

```
4
.pm.
Mom.
OOm.
p..p
```

poprawnym wynikiem jest:

```
4 13
.PM.
MOM.
OOm.
p..p
```

Wyjaśnienie przykładu: Początkowa plansza ma 3 struktury (miasto i dwa oceany) i 1 parę stykających się oceanów, co łącznie daje 4 punkty.



- W pierwszym ruchu budujesz ocean, co daje 3 punkty (jeden za nową strukturę, dwa za dwie nowe pary stykających się oceanów).
- W drugim ruchu budujesz park, co daje 2 punkty (jeden za nową strukturę, jeden za parę miasto-park).
- W trzecim i czwartym ruchu w dowolnej kolejności budujesz dwa nowe miasta, każde dające 2 punkty.

Pozostały trzy pola pozwalające na budowanie struktur, ale ruch na każdym z nich dałby tylko jeden punkt. Rozgrywka dobiega końca z wynikiem 13 punktów (7 struktur, 3 pary miasto-park, 3 pary ocean-ocean).

ORGANIZATORZY
UNIWERSYTET WARSZAWSKI
FUNDACJA ROZWOJU INFORMATYKI
PARTNER
IDEAS NCBR
DIAMENTOWY SPONSOR
HUAWEI
ZŁOCI SPONSORZY
DIGITAL TECHNOLOGY POLAND
ElevenLabs
Jane Street
neptune.ai
SPONSORZY
ATENDE
Google
intel

H – Hop

Limit pamięci: **1024 MB**
Limit czasu: **2 s**

AMPPZ 2024
2024-11-17

Henryk startuje w zawodach skoku wzwyż. Przed każdym skokiem może wybrać wysokość zawieszenia poprzeczki – liczbę całkowitą od 1 do n. Jako jego wieloletni trener znasz dobrze zdolności Henryka. Dla każdej możliwej wysokości h znasz prawdopodobieństwo p_h udanego przeskoczenia poprzeczki na tej wysokości. Im większa wysokość, tym oczywiście mniejsze prawdopodobieństwo sukcesu.

W dzisiejszych zawodach nie ma miejsca na błędy. Już jeden nieudany skok kończy występ zawodnika, a jego wynikiem jest największa zaliczona wcześniej wysokość (lub 0, jeśli już pierwszy skok się nie udał). Występ kończy się automatycznie z wynikiem n w przypadku zaliczenia wysokości n. Pomóż Henrykowi w optymalnym wyborze kolejnych wysokości poprzeczki. Jaka jest największa możliwa wartość oczekiwana* wyniku?

Wejście

Pierwszy wiersz zawiera liczbę całkowitą n ($1 \leq n \leq 500\,000$), oznaczającą limit na wysokość poprzeczki.

Drugi wiersz zawiera n liczb rzeczywistych $p_1, p_2, \ldots, p_n$ ($0 < p_i < 1$; $p_i > p_{i+1}$), każda z co najwyżej 9 cyframi po kropce. Liczba p_i to prawdopodobieństwo udanego skoku z poprzeczką na wysokości i.

Wyjście

Wypisz jedną liczbę rzeczywistą – maksymalną wartość oczekiwaną wyniku Henryka.

Dopuszczalny błąd względny lub bezwzględny to 10^{-6}. Czyli jeśli wypiszesz x, a poprawny dokładny wynik to y, to musi zachodzić $|x - y| \leq 10^{-6} \cdot \max(1, y)$. Możesz wypisać co najwyżej 20 cyfr po kropce.

Przykład

Dla danych wejściowych:

```
5
0.9 0.85 0.6 0.456000 0.000000017
```

poprawnym wynikiem jest:

```
2.475200006589
```

Wyjaśnienie przykładu: Optymalna jest następująca strategia:

- Ustawiamy poprzeczkę na wysokości 2. Henryk przeskakuje z prawdopodobieństwem 0.85 albo kończy zawody z wynikiem 0 (p-stwo 0.15).
- Jeśli udał się pierwszy skok, to ustawiamy poprzeczkę na wysokości 4. Henryk przeskakuje z p-stwem 0.456 albo kończy z wynikiem 2.
- Jeśli udał się drugi skok, to ustawiamy poprzeczkę na wysokości 5. Henryk albo przeskakuje z p-stwem 0.000000017 i kończy z wynikiem 5, albo strąca poprzeczkę i kończy z wynikiem 4.

Wartość oczekiwana wyniku takiej strategii to:

$$0 \cdot 0.15 + 2 \cdot 0.85 \cdot 0.544 + 4 \cdot 0.85 \cdot 0.456 \cdot 0.999999983 + 5 \cdot 0.85 \cdot 0.456 \cdot 0.000000017 = 2.4752000065892$$

* *Wartością oczekiwaną* nazywamy średnią, ważoną prawdopodobieństwem, wartość zmiennej losowej. Intuicyjnie, jest to spodziewany średni wynik doświadczenia losowego przy jego wielokrotnym powtarzaniu.

ORGANIZATORZY
UNIWERSYTET WARSZAWSKI
FUNDACJA ROZWOJU INFORMATYKI
PARTNER
IDEAS NCBR
DIAMENTOWY SPONSOR
HUAWEI
ZŁOCI SPONSORZY
DIGITAL TECHNOLOGY POLAND
ElevenLabs
Jane Street
neptune.ai
SPONSORZY
ATENDE
Google
intel

I – Intensywny trening

Limit pamięci: **1024 MB** *AMPPZ 2024*
Limit czasu: **4 s** *2024-11-17*

Na zajęcia siatkówki regularnie przychodzi n zawodników. Trener Igor zna dobrze ich zdolności: a_i to poziom i-tego zawodnika. Każdy trening składa się z ciągu meczów, a do każdego meczu trener wybiera dwie rozłączne drużyny po k zawodników w każdej. Poziomem drużyny nazywamy sumę poziomów jej zawodników.

Trener zauważył, że im większa różnica (tzn. wartość bezwzględna różnicy) między poziomami drużyn, tym szybciej kończy się mecz. A im szybciej kończą się mecze, tym więcej można ich przeprowadzić w trakcie jednego treningu! Postanowił więc wybierać drużyny w taki sposób, by maksymalizować różnicę ich poziomów.

Pomóż Igorowi zaplanować cały trening. Znajdź maksymalną możliwą różnicę między poziomami dwóch drużyn, a następnie zlicz modulo $10^9 + 7$ mecze osiągające to maksimum.

Mecze nie mogą się powtarzać, czyli dwie drużyny mogą grać przeciwko sobie co najwyżej raz. Na przykład dla $n = 4$ i $k = 2$ są tylko trzy możliwe mecze: zawodnicy 1 i 2 grają przeciwko zawodnikom 3 i 4; albo zawodnicy 1 i 3 przeciwko 2 i 4; albo zawodnicy 1 i 4 przeciwko 2 i 3.

Wejście

Pierwszy wiersz zawiera dwie liczby całkowite n i k ($2 \le n \le 2000$; $1 \le k \le \frac{n}{2}$), oznaczające odpowiednio liczbę dostępnych zawodników oraz rozmiar każdej z drużyn.

Drugi wiersz zawiera n liczb całkowitych $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$), poziomy poszczególnych zawodników.

Wyjście

Wypisz dwie liczby całkowite – maksymalną różnicę w poziomach drużyn oraz resztę z dzielenia przez $10^9 + 7$ liczby meczów, które osiągają to maksimum.

Przykład

Dla danych wejściowych:

```
6 2
2 5 7 2 5 2
```

poprawnym wynikiem jest:

```
8 6
```

Natomiast dla danych wejściowych:

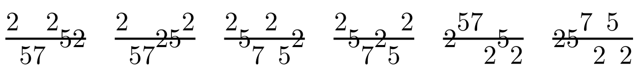
```
5 2
1 1 1 1 1
```

poprawnym wynikiem jest:

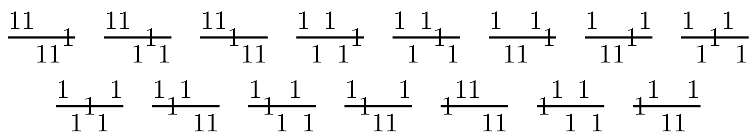
```
0 15
```

Wyjaśnienie przykładów:

W pierwszym teście przykładowym trener wybiera w każdym meczu dwie drużyny po $k = 2$ zawodników. Może przeprowadzić 6 meczów z różnicą poziomów 8. Mecze przedstawione są na poniższym rysunku, na przykład w pierwszym meczu zawodnicy 1 i 4 grają przeciwko zawodnikom 2 i 3, a różnica poziomów to $|(a_1 + a_4) - (a_2 + a_3)| = |(2 + 2) - (5 + 7)| = 8$.



W drugim teście przykładowym wszyscy zawodnicy mają ten sam poziom, więc różnica w poziomach drużyn zawsze wynosi 0. Zatem wszystkie 15 możliwych meczów osiąga maksymalną różnicę poziomów:



ORGANIZATORZY PARTNER DIAMENTOWY SPONSOR ZŁOCI SPONSORZY SPONSORZY

J – Jedynki i zera

Limit pamięci: **1024 MB**
Limit czasu: **1.5 s**

AMPPZ 2024
2024-11-17

Julia ma prostokątną planszę składającą się z $h \cdot w$ bitów ułożonych w h rzędów i w kolumn, gdzie liczba rzędów jest mała ($h \le 8$). Rzędy są ponumerowane od 1 do h od góry do dołu, a kolumny od 1 do w od lewej do prawej. Bit to wartość 0 lub 1, a *negowaniem* bitu nazywamy zmianę wartości na przeciwną: 0 na 1 lub 1 na 0.

Na planszy można wykonywać operacje trzech typów:

- `P` i j – zaneguj bit na przecięciu i-tego rzędu i j-tej kolumny,
- `R` i – zaneguj wszystkie bity w i-tym rzędzie,
- `K` j – zaneguj wszystkie bity w j-tej kolumnie.

Julia chciałaby wyzerować planszę, czyli zmienić wszystkie bity na 0. Minimalną liczbę operacji do tego potrzebnych nazwiemy *trudnością* planszy.

Psotny Romek lubi przeszkadzać Julii i łącznie q razy wykonuje operację jednego z trzech powyższych typów. Nie zdaje sobie jednak sprawy, że Julii pasuje takie wyzwanie. Obserwuje ona planszę i oblicza jej trudność w każdym z $q+1$ momentów (na samym początku oraz po każdej operacji Romka). Czy Ty też jesteś w stanie wyznaczyć te wartości?

Romek zmienia planszę na stałe. Julia nie wykonuje żadnych operacji, czyli nie zmienia planszy.

Wejście

Pierwszy wiersz zawiera trzy liczby całkowite h, w i q ($1 \le h \le 8$; $1 \le w, q \le 10^5$), oznaczające wymiary planszy i liczbę operacji wykonanych przez Romka.

Kolejne h wierszy opisuje początkową planszę: każdy z nich zawiera słowo binarne (znaki `0` i `1`) o długości w.

Ostatnie q wierszy opisuje operacje Romka, każdy wiersz w formacie „`P` i j”, lub „`R` i”, lub „`K` j” ($1 \le i \le h$; $1 \le j \le w$).

Wyjście

Wypisz $q+1$ liczb całkowitych (każdą w nowym wierszu) – trudność początkowej planszy oraz trudność planszy po każdej z q operacji Romka.

Przykład

Dla danych wejściowych:

```
3 4 6
1010
1101
0010
R 2
P 3 1
K 2
P 2 1
K 4
P 3 4
```

poprawnym wynikiem jest:

```
3
2
3
4
3
3
4
```

Wyjaśnienie przykładu: Rysunek pokazuje początkową planszę oraz kilka pierwszych operacji Romka. Trudność początkowej planszy to 3, bo Julia może użyć operacji: `P 1 1`, `R 2`, `K 3`.

```
          1010           1010            1010           1110            1110
          1101   R 2 ->  0010   P 3 1 -> 0010   K 2 ->  0110   P 2 1 -> 1110   K 4 ->   ...
          0010           0010            1010           1110            1110
trudność:  3              2               3              4               3
```


ORGANIZATORZY
UNIWERSYTET WARSZAWSKI
FUNDACJA ROZWOJU INFORMATYKI
PARTNER
IDEAS NCBR
DIAMENTOWY SPONSOR
HUAWEI
ZŁOCI SPONSORZY
DIGITAL TECHNOLOGY POLAND
ElevenLabs
Jane Street
neptune.ai
SPONSORZY
ATENDE
Google
intel

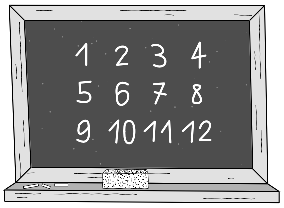
K – Kwadracik

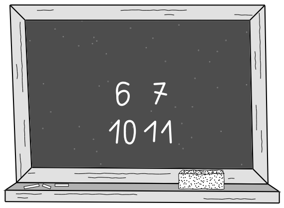
Limit pamięci: 1024 MB
Limit czasu: 3 s

AMPPZ 2024
2024-11-17

Dzieci w przedszkolu uczą się liczb naturalnych. Nauczyciel wypisał na tablicy kolejno liczby od 1 do $h \cdot w$, ułożone w h wierszy po w kolumn ($h, w \geq 2$). Pierwszy wiersz zawiera zatem liczby od 1 do w (od lewej do prawej), drugi wiersz to liczby od $w+1$ do $2 \cdot w$ itd. Po zakończonej lekcji nauczyciel zmazał wszystkie liczby poza spójnym kwadracikiem 2×2, zawierającym cztery sąsiadujące ze sobą liczby z oryginalnego układu.

Rysunek przedstawia zapisaną tablicę dla $h = 3$, $w = 4$ oraz przykładowy niezmazany kwadracik 2×2:





Jeden z przedszkolaków, Kamilek, niespecjalnie uważał podczas lekcji i zastanawia się jakich liczb h i w użył nauczyciel. Znajdź dowolną możliwą parę (h, w) albo wypisz liczbę `-1` jeśli dany kwadracik nie jest możliwy do uzyskania. Cała sytuacja powtarza się przez t dni, dane jako niezależne przypadki testowe.

Wejście

Pierwszy wiersz zawiera liczbę całkowitą t ($1 \leq t \leq 10$), oznaczającą liczbę przypadków testowych.

Następne $2 \cdot t$ wierszy opisuje przypadki testowe. Każdy przypadek to dwa wiersze po dwie liczby całkowite z przedziału $[1, 50\,000]$, reprezentujące niezmazany kwadracik 2×2.

Wyjście

Wypisz t wierszy, i-ty z rozwiązaniem i-tego przypadku testowego:

- Jeśli dany kwadracik jest możliwy do uzyskania, wypisz dwie liczby całkowite oddzielone spacją – dowolne możliwe wymiary planszy h, w ($2 \leq h, w \leq 100\,000$). Da się udowodnić, że jeśli istnieje jakakolwiek poprawna para (h, w), to istnieje też para spełniająca zadane nierówności. Jeśli istnieje wiele rozwiązań, wypisz jedno dowolne z nich.
- Jeśli kwadracik nie mógł powstać, wypisz jedną liczbę `-1`.

Przykład

Dla danych wejściowych:

```
4
6 7
10 11
2 3
4 5
8 5
5 13
1 2
5 6
```

jednym z poprawnych wyników jest:

```
3 4
-1
-1
3 4
```

Wyjaśnienie przykładu: W pierwszym przypadku testowym mamy kwadracik $[[6,7],[10,11]]$. Rysunek powyżej przedstawia jedno z wielu poprawnych rozwiązań: $h = 3, w = 4$. Jest to też jedno z poprawnych rozwiązań czwartego przypadku $[[1,2],[5,6]]$.

W drugim przypadku liczby $[[2,3],[4,5]]$ mogą być początkowo wypisane przez nauczyciela, ale nigdy nie układają się w kwadracik 2×2. Należy więc wypisać `-1`.

ORGANIZATORZY
UNIWERSYTET WARSZAWSKI
FUNDACJA ROZWOJU INFORMATYKI
PARTNER
IDEAS NCBR
DIAMENTOWY SPONSOR
HUAWEI
ZŁOCI SPONSORZY
DIGITAL TECHNOLOGY POLAND
IIElevenLabs
Jane Street
neptune.ai
SPONSORZY
ATENDE
Google
intel

L – Licz trójki PKN

Limit pamięci: **1024 MB** *AMPPZ 2024*
Limit czasu: **4 s** *2024-11-17*

amppz

W pojedynku w papier-kamień-nożyce dwóch graczy pokazuje jednocześnie jeden z trzech tytułowych znaków. Gdy dwóch graczy pokaże różne znaki, pojedynek kończy się zwycięstwem zgodnie z zasadą: papier bije kamień, kamień bije nożyce, nożyce biją papier. Jeśli obaj gracze pokazali ten sam znak, próbują ponownie. Pojedynek może trwać w nieskończoność, co oznacza remis.

Już dzisiaj odbywają się AMPPKN*. Na tak wysokim poziomie nikt nie wnioskuje z ruchów przeciwnika, obawiając się podstępnej strategii, która właśnie na to liczy. Losowanie ruchów też jest trudne, więc zamiast tego każdy zawodnik przed startem spisał na swoim ramieniu strategię – słowo s złożone z liter `P`, `K` i `N`. W każdym pojedynku zawodnik powtarza w nieskończoność swoją strategię od pierwszego znaku, na przykład `PKK` oznacza granie kolejno: papier, kamień, kamień, papier, kamień, kamień itd.

W AMPPKN bierze udział n zawodników, i-ty ze strategią opisaną słowem s_i. Organizatorzy są zainteresowani trójkami zawodników, które zachowują się jak tytułowe papier-kamień-nożyce, czyli każdy zawodnik wygrałby pojedynek z jednym z dwóch pozostałych.

Mówiąc formalnie, zlicz nieuporządkowane trójki zawodników, że ci trzej zawodnicy w pewnej kolejności (A, B, C) spełniają, że A wygrałby z B, B wygrałby z C, C wygrałby z A.

Wejście

Pierwszy wiersz zawiera liczbę całkowitą n ($3 \leq n \leq 10^5$).

Każdy z n kolejnych wierszy zawiera jedno niepuste słowo s_i złożone z liter `P`, `K`, `N`. Suma długości słów nie przekracza 10^6.

Wyjście

Wypisz jedną liczbę całkowitą – liczbę szukanych trójek zawodników.

Przykład

Dla danych wejściowych:

```
6
P
PN
KK
N
PKK
PN
```

poprawnym wynikiem jest:

```
6
```

Wyjaśnienie przykładu: Jest 6 takich trójek zawodników:

`(P, KK, N), (P, PKK, PN), (P, PKK, PN'), (PN, KK, N), (KK, N, PKK), (KK, N, PN'),`

gdzie `PN'` oznacza drugie wystąpienie słowa `PN`.

Przykładowo w drugiej trójce: `P` wygrałby z `PKK` (powtórzony papier bije kamień w drugim ruchu), `PKK` wygrałby z `PN`, a `PN` wygrałby z `P`.

*Akademickie Mistrzostwa Polski w Papier-Kamień-Nożyce

ORGANIZATORZY
UNIWERSYTET WARSZAWSKI
FUNDACJA ROZWOJU INFORMATYKI
PARTNER
IDEAS NCBR
DIAMENTOWY SPONSOR
HUAWEI
ZŁOCI SPONSORZY
DIGITAL TECHNOLOGY POLAND
IIElevenLabs
Jane Street
neptune.ai
SPONSORZY
ATENDE
Google
intel