Zadanie A: AMPPZ w czasach zarazy

Limit czasowy: 12s, limit pamięciowy: 1GB.

Organizacja Akademickich Mistrzostw Polski w Programowaniu Zespołowym w czasach pandemii to nie lada wyzwanie. Twoim zadaniem jako Głównego Sędziego ds. Dystansu Społecznego jest dopilnowanie, aby zachowany był odpowiedni dystans pomiędzy uczestnikami. Zawodnicy z jednego uniwersytetu są dla siebie praktycznie jak rodzina, więc głównie martwi Cię dystans pomiędzy studentami z różnych uniwersytetów. Intuicyjnie, chcesz aby reprezentanci każdego uniwersytetu skupili się w zwartą grupę zachowującą odpowiedni dystans od innych grup.

Aby opisać swoją intuicję w sposób formalny, wprowadziłeś następującą zasadę. Niech A oznacza największą odległość (euklidesową, czyli standardową odległość na płaszczyźnie) pomiędzy dwójką studentów należących do tego samego uniwersytetu, zaś B oznacza najmniejszą odległość euklidesową pomiędzy dwójką studentów należących do różnych uniwersytetów. Musi wtedy zachodzić $A < B$.

Twoi goście ochoczo przyjęli zalecenia i stosowali się do nich podczas całej imprezy. Niestety, jest pewien szkopuł: po zawodach otrzymałeś polecenie aby udowodnić, że zasada dystansu społecznego faktycznie była respektowana. Wszyscy rozjechali się już do domów i jedyne co Ci pozostaje, to użyć jednego ze zdjęć grupowych jako dowodu. Problem w tym, że nie wiesz którzy zawodnicy pochodzą z których uniwersytetów. Ale *skoro wiesz, że zasada dystansu społecznego faktycznie była zachowana*, może uda Ci się odtworzyć podział na uniwersytety?

Znając pozycje wszystkich studentów na zdjęciu (opisane jako punkty na płaszczyźnie [1]) oraz liczbę uniwersytetów, znajdź podział, który respektuje Twoją zasadę. **Każdy uniwersytet musi mieć co najmniej jednego studenta. Ponadto możesz założyć, że rozwiązanie zawsze istnieje.**

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 100\,000$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera dwie liczby całkowite n, k ($2 \leqslant n \leqslant 2\,000\,000$, $2 \leqslant k \leqslant \min(n, 20)$), oznaczające odpowiednio liczbę studentów i liczbę uniwersytetów.

Każda z kolejnych n linii zawiera dwie liczby całkowite x_i, y_i ($0 \leqslant x_i, y_i < 10^9$), oznaczające współrzędne i-tego studenta. **Żadna dwójka studentów nie stoi w tym samym punkcie.**

Sumaryczna liczba studentów we wszystkich zestawach danych nie przekroczy 10^7.

Wyjście

Dla każdego zestawu danych wypisz n liczb $c_1, \ldots, c_n$ ($1 \leqslant c_i \leqslant k$), gdzie c_i jest numerem uniwersytetu, z którego pochodzi i-ty student. Przypisanie studentów do uniwersytetów powinno spełniać opisaną powyżej zasadę dystansu społecznego. Jeśli istnieje wiele rozwiązań, możesz wypisać dowolne z nich.

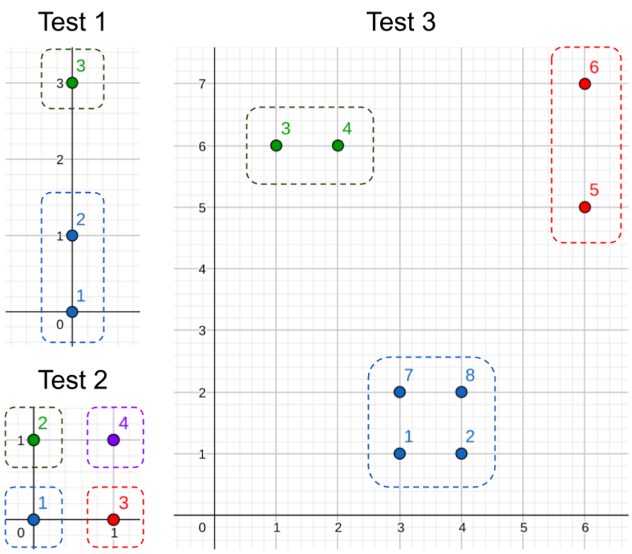
[1] Zdjęcie grupowe zrobione zostało pionowo z góry za pomocą drona, gdyż z tego właśnie kąta zawodnicy wyglądali najkorzystniej.

Przykład

Dla danych wejściowych:	Możliwą poprawną odpowiedzią jest:
3 3 2 0 0 0 1 0 3 4 4 0 0 0 1 1 0 1 1 8 3 3 1 4 1 1 6 2 6 6 5 6 7 3 2 4 2	1 1 2 4 1 3 2 2 2 1 1 3 3 2 2

Uwaga: Puste linie w wejściu testu przykładowego zostały dodane dla czytelności. Nie są one obecne w plikach testowych, na których uruchamiane będzie Twoje rozwiązanie.

Poniższe rysunki przedstawiają testy przykładowe z zaznaczonymi odpowiedziami.



Zadanie B: Babcia i pierogi

Limit czasowy: 15s, limit pamięciowy: 1GB.

Babcia Bajtmiła postanowiła urządzić przyjęcie, na którym pochwali się swoimi najznakomitszymi wyrobami kuchni garmażeryjnej – pierogami. Na przyjęciu będzie n stanowisk, zaś na i-tym z nich ma się znajdować talerz z dokładnie p_i pierogami, przy czym wszystkie liczby pierogów są różne. Zadanie wydawało się przekraczać możliwości jednej starszej pani, ale Bajtmiła wykonała je błyskawicznie, przygotowując zgodnie z planem n talerzy z $p_1, p_2, \ldots, p_n$ pierogami. Niestety, w pośpiechu pomyliła stanowiska i rozmieściła talerze w zupełnie innej kolejności.

Bajtmiła jest już porządnie zmęczona, a dodatkowo boi się całkiem pogubić w sytuacji. Nie chce już ruszać samych talerzy, ale może przenosić pierogi z jednego stanowiska na inne tak, żeby zamieniać wartości miejscami. Dokładniej, może w jednym ruchu wybrać dwa stanowiska, na których znajduje się odpowiednio x i y pierogów, po czym przenieść dokładnie $|x - y|$ między nimi tak, aby na pierwszym było teraz y, a na drugim x. Każda taka operacja zajmie jej dokładnie $C + |x - y|$ sekund (C na znalezienie łyżki, 1 za każdy przeniesiony pieróg).

Przyjęcie już niedługo! Babcia nie pozwoli Ci dotknąć niczego w kuchni, ale jedno możesz zrobić: wyznacz sekwencję zamian, która naprawi sytuację, w najkrótszym możliwym czasie sprawiając, żeby na każdym stanowisku znalazła się właściwa liczba pierogów.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 1\,000$). Potem kolejno podawane są zestawy w następującej postaci:

W pierwszej linii zestawu znajdują się dwie liczby całkowite n oraz C ($1 \leqslant n \leqslant 200\,000$, $1 \leqslant C \leqslant 10^9$).

Następne n linii zestawu zawiera opisy stanowisk. W i-tej linii podane są dwie liczby całkowite a_i oraz p_i ($1 \leqslant a_i, p_i \leqslant 10^9$), które oznaczają odpowiednio aktualną oraz planowaną liczbę pierogów na i-tym stanowisku.

W każdym zestawie liczby a_i są różne. Wiadomo też, że zbiory $\{a_1, a_2, ..., a_n\}$ i $\{p_1, p_2, ..., p_n\}$ są takie same.

Suma wartości n we wszystkich zestawach nie przekracza $1\,000\,000$.

Wyjście

Dla każdego zestawu danych, w pierwszej linii wypisz dwie liczby całkowite S oraz K – odpowiednio sumaryczny czas oraz liczbę operacji w Twoim rozwiązaniu.

Następne K linii powinno opisywać kolejne operacje Twojego rozwiązania. W k-tej linii wypisz dwie liczby x_k oraz y_k oznaczające, że k-ta operacja polega na przeniesieniu pierogów między stanowiskami o numerach x_k oraz y_k. Po wykonaniu wszystkich operacji, dla każdego $1 \leqslant i \leqslant n$, na i-tym stanowisku musi się znajdować p_i pierogów.

Jeżeli istnieje więcej niż jedna możliwa sekwencja zamian o sumarycznym koszcie równym S, możesz wypisać dowolną z nich.

Przykład

Dla danych wejściowych:	Możliwą poprawną odpowiedzią jest:
1 4 2 2 4 3 2 1 1 4 3	6 2 2 1 4 1

Wyjaśnienie

Na stanowiskach leżą kolejno 2, 3, 1 i 4 pierogi. Chcemy doprowadzić do sytuacji, w której będzie to odpowiednio 4, 2, 1 i 3.

W pierwszym ruchu zamieniamy pierogi na stanowisku pierwszym i drugim. Koszt tej zamiany to stała $C = 2$ oraz różnica $|3 - 2| = 1$, czyli łącznie 3 sekundy. Po tej zamianie pierogi leżą w kolejności $(3, 2, 1, 4)$. W drugim ruchu zamieniamy pierogi na pierwszym i ostatnim stanowisku. Koszt to $2+|4-3|$, czyli znowu 3 sekundy. Teraz kolejność to $(4, 2, 1, 3)$, czyli taka, jaką chcieliśmy uzyskać. Łączny koszt zamian wynosi 6 sekund i jest minimalny możliwy.

Zadanie C: Ciasto

Limit czasowy: 15s, limit pamięciowy: 1GB.

Babcia Bajtmiła (znana już z zadania B, w którym wsławiła się znakomitymi pierogami) tym razem upiekła sernik. Pokroiła go na $2n$ prostokątnych kawałków (w dwóch rzędach po n kawałków) i na każdy z nich nałożyła lukier w jednym z wybranych przez siebie kolorów. Z dumą spojrzała na swoje dzieło i zaniemówiła: powstała mozaika kolorów wyglądała *paskudnie*.

Bajtmiła zdecydowała, że musi zmienić układ kolorów na ładniejszy. Zamienianie kawałków ciasta po jednym nie wchodzi jednak w grę: próba wyjęcia pojedynczego kawałka z sernika niewątpliwie spowodowałaby ukruszenie się jego brzegów. Kto jak kto, ale babcia Bajtmiła nigdy nie podałaby gościom ciasta, które sprawia wrażenie nierówno pokrojonego!

Na szczęście, Bajtmiła dysponuje prostokątną łopatką do ciast, na której mieszczą się dokładnie cztery kawałki (w dwóch rzędach po dwa kawałki). Za jej pomocą jest w stanie ostrożnie wyciągnąć takie cztery kawałki z ciasta, po czym – obróciwszy łopatkę – włożyć je od przeciwnej strony z powrotem w to samo miejsce. Operacja taka może być opisana jako wybranie pewnego kwadratu o wymiarach 2 na 2 i obrócenie go o 180 stopni.

Nauczona doświadczeniem, babcia od razu zwróciła się o pomoc do Ciebie i poprosiła o wyznaczenie najmniejszej możliwej liczby ruchów, która zamieni układ kolorów na ładniejszy. Oczywiście zrobiłaś to, co każdy zrobiłby na Twoim miejscu: odpowiedziałaś, że nie podejmiesz się tego zadania, ponieważ specyfikacja wymagań jest nieprecyzyjna. Bajtmiła nie zraziła się jednak: narysowała na tablicy pewną konkretną *ładnie wyglądającą* mozaikę kolorów (również o wymiarach 2 na n) i poprosiła Cię o wyznaczenie minimalnej liczby ruchów, doprowadzającej sernik do dokładnie tej postaci.

Oczywiście, mogło się również zdarzyć, że babcia (zmęczona przenoszeniem pierogów w zadaniu B) pomyliła się i układu kolorów narysowanego na tablicy w ogóle nie da się w opisany sposób uzyskać. W takim przypadku również jak najszybciej musisz jej o tym powiedzieć.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 10\,000$). Potem kolejno podawane są zestawy w następującej postaci:

W pierwszej linii zestawu znajduje się liczba całkowita n ($2 \leqslant n \leqslant 500\,000$). Następne dwie linie wejścia opisują początkowe kolory dwóch rzędów sernika. Każda z tych linii zawiera n liczb całkowitych z przedziału $[1, 10^9]$ oddzielonych pojedynczymi spacjami — identyfikatory kolorów na kolejnych kawałkach sernika. Zauważ, że kolory mogą być użyte wielokrotnie, zatem identyfikatory mogą się powtarzać.

Kolejne dwie linie opisują, w takim samym formacie, docelowy układ kolorów.

Suma wszystkich wartości n we wszystkich zestawach danych nie przekracza $2\,000\,000$.

Wyjście

Dla każdego zestawu danych wypisz jedną liczbę — minimalną liczbę operacji niezbędnych do uzyskania pożądanego układu kolorów. Jeśli wymaganej konfiguracji nie da się osiągnąć, zamiast tego wypisz liczbę -1.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
2 4 1 2 3 2 4 3 1 3 3 2 1 1 2 3 4 3 2 1 2 3 4 3 4 1 2	3 -1

Wyjaśnienie

W pierwszym teście sernik o początkowej konfiguracji

```
1 2 3 2
4 3 1 3
```

możemy przeprowadzić w żądaną konfigurację za pomocą następujących trzech kroków:
1. odwrócenie skrajnie lewego kwadratu,

```
3 4 3 2
2 1 1 3
```

2. odwrócenie skrajnie prawego kwadratu,

```
3 4 3 1
2 1 2 3
```

3. odwrócenie środkowego kwadratu.

```
3 2 1 1
2 3 4 3
```

W drugim teście pierwszej konfiguracji nie da się w żaden sposób przeprowadzić na drugą.

Zadanie D: Dwuczęściowy mechanizm

Limit czasowy: 10s, limit pamięciowy: 1GB.

Maszyna *Bytegate*, najnowszy wynalazek Bajtazara, to mechanizm składający się z dwóch części, których – cytując podekscytowanego Bajtazara – "*po prostu nie da się rozdzielić!*". Jego dwuletni syn Bajtuś za chwilę pokaże mu, jak bardzo się myli.

Początkowy stan *Bytegate* możemy opisać jako tablicę o rozmiarach $n \times m$, wypełnioną znakami `A`, `B` oraz kropkami. Litery `A` oznaczają pierwszą część maszyny, litery `B` oznaczają drugą część maszyny, zaś kropki oznaczają pustą przestrzeń. Przykładowy rysunek znajduje się poniżej:

B	B	A	A	.
.	B	B	A	A
A	.	B	B	A
A	.	.	B	A
A	A	A	A	A

Jest to dwuczęściowy mechanizm, dlatego na tablicy znajdzie się co najmniej jeden znak `A` oraz co najmniej jeden znak `B`. Oprócz tego obydwie części maszyny są spójne, tzn. dla dowolnych dwóch pól oznaczonych etykietą `A` istnieje pewna ścieżka łącząca te pola, w której każde kolejne pole ma wspólny bok z poprzednim, a także każde pole na ścieżce ma etykietę `A`. W ten sam sposób spójna jest część B.

Część A maszyny pozostawała nieruchoma, podczas gdy Bajtuś pchał część B w różnych kierunkach. Jego zabawę możemy więc opisać jako sekwencję q liter `N`, `S`, `E`, `W` (oznaczających odpowiednio kierunek północny, południowy, wschodni i zachodni). Za każdym razem Bajtuś pchał część B mechanizmu "do oporu" tj. do momentu, w którym dalsze jej przesunięcie oznaczałoby nałożenie na siebie dwóch części maszyny. Mogło się zdarzyć, że Bajtuś był w stanie przesuwać część B w nieskończoność – w takim przypadku powiemy, że udało mu się rozdzielić obie części. Nie oznacza to jednak, że Bajtuś przestał w tym momencie szarpać mechanizmem. Niemniej jednak uznajemy, że raz rozdzielone części maszyny pozostają w tym stanie do końca zabawy Bajtusia.

Pomóż stwierdzić czy w trakcie zabawy Bajtuś rozdzielił obie części *Bytegate*.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 1\,000$). Potem kolejno podawane są zestawy w następującej postaci:

W pierwszej linii zestawu znajdują się trzy liczby całkowite n, m, q ($1 \leqslant n, m \leqslant 10$, $1 \leqslant q \leqslant 100$). Następne n linii wejścia opisuje początkowy stan maszyny. Każda z tych linii to napis długości m złożony ze znaków `A`, `B` oraz kropek. Obydwie części maszyny są niepuste oraz spójne.

Ostatnia linia zestawu zawiera ciąg q liter należących do zbioru $\{N, S, E, W\}$, o znaczeniu podanym w treści zadania.

Wyjście

Dla każdego zestawu danych wypisz słowo `TAK` lub `NIE` oznaczające, czy Bajtusiowi udało się rozdzielić od siebie dwie części maszyny.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:

<pre>3
5 5 3
BBAA.
.BBAA
A.BBA
A..BA
AAAAA
WNW
5 5 7
BBAA.
.BBAA
A.BBA
A..BA
AAAAA
WNWNSEN
6 5 3
.....
.AAA.
.A.A.
.AB..
.A.A.
.AAA.
SNE</pre>
	<pre>NIE
	TAK
	NIE</pre>

Wyjaśnienie

Kiedy Bajtuś zakończył zabawę maszynami z pierwszego i trzeciego zestawu, znajdowały się one – odpowiednio – w następujących stanach:

B	B	.	.	.	.	.
.	B	B	.	A	A	.
.	.	B	B	.	A	A
.	.	A	B	.	.	A
.	.	A	.	.	.	A
.	.	A	A	A	A	A

A	A	A
A	B	A
A	.	.
A	.	A
A	A	A

W drugim zestawie Bajtusiowi udało się rozdzielić obie części mechanizmu w czwartym ruchu.

Zadanie E: Epidemia

Limit czasowy: 9s, limit pamięciowy: 1GB.

Bajtocja zamknęła granice z powodu wykrycia u jej sąsiadów zakażeń nowym szczepem bajtobakterii. Badania wskazują, że szczep ten nie tylko jest wysoce zaraźliwy, lecz również nie powoduje żadnej reakcji układu immunologicznego Bajtocjan, przez co raz zarażona osoba pozostanie zarażona – oraz będzie zarażać innych – dożywotnio (lub przynajmniej do czasu wynalezienia skutecznego lekarstwa).

Aby nie dopuścić do rozprzestrzenienia się bajtobakterii, rząd Bajtocji wprowadził daleko idące obostrzenia oraz uruchomił Narodowy System Inwigilacji, pozwalający monitorować wszystkie kontakty społeczne w kraju. Zapowiedziano przy tym, że obostrzenia zostaną zniesione dopiero wtedy, gdy pewnym będzie, że nikt oprócz osób przebywających na kwarantannie nie jest zarażony bajtobakterią. Jako naczelnemu informatykowi Bajtocji, Tobie została powierzona analiza danych z systemu inwigilacji i określenie, w którym momencie obostrzenia będą mogły zostać uchylone.

Na terenie Bajtocji znajduje się n osób, z których początkowo każda może być zarażona bajtobakterią albo zdrowa. Po zamknięciu granic następuje ciąg k zdarzeń, z których każde ma jedną z następujących form:

- Otrzymujesz z systemu inwigilacji informację, że pewna grupa osób spotyka się. Jeżeli którakolwiek z nich była zarażona, to wszystkie stają się zarażone (i pozostaną już zarażone do końca życia). Kontakt taki jest jedynym możliwym sposobem zarażenia się (wbrew początkowym doniesieniom, dotykanie zakażonych powierzchni nie może spowodować zarażenia się bajtobakterią).
- Pewnej osobie zostaje wykonany test na obecność bajtobakterii i daje on wynik negatywny.
- Pewnej osobie zostaje wykonany test na obecność bajtobakterii i daje on wynik pozytywny. Osoba taka zostaje bezzwłocznie skierowana na bezterminową kwarantannę i nie będzie od tego momentu uczestniczyć w żadnych kontaktach społecznych (może natomiast zdarzyć się, że zostanie jej w przyszłości wykonany kolejny test [1]).
- Otrzymujesz od ministra zdrowia Bajtocji zapytanie, czy zniesienie obostrzeń jest już możliwe, to znaczy czy w oparciu o wszystkie zebrane do tej pory informacje da się udowodnić, że nikt oprócz przebywających na kwarantannie nie może być zarażony. Jeśli wciąż mogą istnieć osoby zarażone, musisz podać przykład takiej osoby, według wytycznych Ministerstwa (opisanych w sekcji *Wyjście*).

Co ważne, Twój program musi działać *online*, to znaczy udzielać odpowiedzi bezpośrednio po każdym zapytaniu ministra, przed wczytaniem kolejnych zapytań.

[1] Możesz zastanawiać się, po co ktoś miałby wykonywać kolejny test takiej osobie, skoro z góry przesądzone jest, że musi on również dać wynik pozytywny. Autorzy zadania wystosowali w tej sprawie zapytanie do Ministerstwa Zdrowia Bajtocji, otrzymali jednak informację, że czas przeznaczony na udzielenie odpowiedzi został przedłużony o trzy miesiące z uwagi na skomplikowany charakter sprawy.

Wejście

Właściwa interpretacja danych wejściowych zależeć będzie od aktualnej wartości zmiennej *shift*. Na początku każdego zestawu danych wynosi ona 0, zaś jej kolejne wartości będą zależne od udzielanych przez Twój program odpowiedzi. Taki opis wejścia ma za zadanie wymusić, aby Twój program odpowiadał na każde z zapytań bezpośrednio po jego wczytaniu.

Funkcja dekodująca zdefiniowana jest w następujący sposób:

```
decode (p) = ((p - 1 + shift) mod n) + 1,
```

gdzie p jest liczbą całkowitą spełniającą $1 \leqslant p \leqslant n$, a `mod` to operacja reszty z dzielenia.

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 1\,000$). Potem kolejno podawane są zestawy w następującej postaci:

W pierwszej linii znajdują się dwie liczby całkowite n oraz k ($1 \leqslant n \leqslant 500\,000$, $1 \leqslant k \leqslant 1\,000\,000$), oznaczające odpowiednio liczbę osób przebywających na terenie Bajtocji oraz liczbę zdarzeń. Osoby ponumerowane są od 1 do n.

Kolejnych k linii opisuje następujące po sobie zdarzenia. Mogą one być następującej postaci:

- Litera `K` oraz liczba całkowita c ($2 \leqslant c \leqslant n$), po której następuje c różnych liczb całkowitych $p_1, \ldots, p_c$ ($1 \leqslant p_i \leqslant n$) – kontakt społeczny, w którym uczestniczy c osób o indeksach $decode(p_1), \ldots, decode(p_c)$.
- Litera `N` oraz liczba całkowita p ($1 \leqslant p \leqslant n$) – osobie o indeksie $decode(p)$ wykonany zostaje test, który daje wynik negatywny.
- Litera `P` oraz liczba całkowita p ($1 \leqslant p \leqslant n$) – osobie o indeksie $decode(p)$ wykonany zostaje test, który daje wynik pozytywny, zaś osoba ta trafia na kwarantannę. Możesz założyć, że nie będzie ona od tego momentu uczestniczyć w żadnych kontaktach społecznych (zaś każdy wykonany jej w przyszłości test oczywiście da również pozytywny wynik).
- Litera `Q` oraz liczba całkowita p ($1 \leqslant p \leqslant n$) – zapytanie ministra zdrowia, w którym *wartością startową* (patrz sekcja *Wyjście*) jest $decode(p)$.

Sumy liczb n oraz k we wszystkich zestawach nie przekraczają, odpowiednio, $500\,000$ oraz $1\,000\,000$. Suma liczb c we wszystkich zapytaniach wszystkich zestawów nie przekracza $1\,000\,000$.

Wyjście

Dla każdego zestawu danych wypisz tyle linii, ile było w nim zapytań (zdarzeń typu `Q`).

Jeżeli w momencie otrzymania i-tego zapytania da się udowodnić, że nikt oprócz osób przebywających na kwarantannie nie jest zarażony bajtobakterią, w i-tej linii wypisz pojedyncze słowo `TAK` [2]. Po takim zapytaniu, zmienna *shift* zmienia swoją wartość na 0.

W przeciwnym przypadku, w i-tej linii wypisz słowo `NIE` oraz identyfikator jednej osoby. Jeżeli $decode(p)$ jest *wartością startową* tego zapytania, to musisz wypisać identyfikator **pierwszej osoby w ciągu** $(decode(p), decode(p) + 1, \ldots, n, 1, 2, \ldots, decode(p) - 1)$, która może być zarażona bajtobakterią, a **nie przebywa na kwarantannie**. Wypisana liczba staje się nową wartością zmiennej *shift*.

[2] Może zdarzyć się, że wszyscy mieszkańcy Bajtocji zostaną poddani kwarantannie. Oczywiście, w takiej sytuacji zdanie to staje się pustospełnione i Twój program również powinien wypisać "TAK".

Uwagi

- Wartość zmiennej *shift* nie zmienia się przy wydarzeniach typu `K`, `N` oraz `P`.
- Podane w danych wejściowych pozytywne i negatywne wyniki testów zawsze opisują wiarygodny scenariusz, tj. dla każdego zestawu danych istnieje co najmniej jeden możliwy zbiór osób początkowo zarażonych, dla którego podane dane nie zawierają sprzeczności.

Przykład

<table>
<tr><th>Dla danych wejściowych:</th><th>Poprawną odpowiedzią jest:</th></tr>
<tr><td><pre>1
6 14
K 3 3 4 5
K 2 6 5
N 3
Q 3
P 1
K 2 6 2
P 6
Q 4
P 6
K 2 1 3
N 3
Q 4
N 2
Q 1</pre></td><td><pre>NIE 5
NIE 1
TAK
TAK</pre></td></tr>
</table>

- Po zdekodowaniu, powyższy przykład wygląda następująco:

```
1
6 14
K 3 3 4 5
K 2 6 5
N 3
Q 3
P 6
K 2 5 1
P 5
Q 3
P 1
K 2 2 4
N 4
Q 5
N 2
Q 1
```

Wyjaśnienie

- Przed pierwszym zapytaniem, osoby 3, 4 i 5 spotykają się, następnie osoby 5 i 6 spotykają się, po czym osoba 3 otrzymuje negatywny wynik testu. W momencie otrzymania pierwszego zapytania, możemy w oparciu o dotychczas zebrane informacje wyprowadzić następujące wnioski (poczynając od *wartości startowej* zapytania, czyli 3). Osoba 3 musi być w tym momencie zdrowa (otrzymała ona właśnie negatywny wynik testu). Osoba 4 również musi być zdrowa (nie mogła ona być chora w momencie pierwszego spotkania, gdyż zaraziłaby wtedy osoby 3 i 5, co stanowiłoby sprzeczność z uzyskanym później przez osobę 3 negatywnym wynikiem testu; od czasu tego spotkania osoba 4 nie mogła się zaś zarazić). Osoba 5 może potencjalnie być zarażona (musiała być ona zdrowa w momencie spotkania z osobami 3 i 4, jednak później spotkała się z osobą 6, której statusu zdrowotnego nie mamy szans wywnioskować z zebranych informacji). Odpowiedzią dla zapytania jest więc `NIE 5`, a zmienna *shift* przyjmuje wartość 5.

- Następnie, osoba 6 otrzymuje pozytywny wynik testu, po czym osoby 1 i 5 spotykają się, później zaś osoba 5 otrzymuje pozytywny wynik testu. W momencie otrzymania drugiego zapytania, możemy w oparciu o dotychczas zebrane informacje wyprowadzić następujące wnioski (poczynając od *wartości startowej* 3). Osoby 3 i 4 nadal muszą być zdrowe. Osoby 5 i 6 przebywają na kwarantannie. Osoba 1 może być zarażona (co więcej, jesteśmy nawet w stanie wywnioskować, że osoba 1 *musi* być w tym momencie zarażona, jednak rozstrzygnięcie tego nie jest wymagane w zadaniu). Odpowiedzią dla zapytania jest więc `NIE 1`, a zmienna *shift* przyjmuje wartość 1.

- Następnie, osoba 1 otrzymuje pozytywny wynik testu, po czym osoby 2 i 4 spotykają się, później zaś osoba 4 otrzymuje negatywny wynik testu. W momencie otrzymania trzeciego zapytania, możemy w oparciu o dotychczas zebrane informacje wyprowadzić następujące wnioski (zaczynając od *wartości startowej* 5). Osoby 5, 6 i 1 przebywają na kwarantannie. Osoba 2 musi być zdrowa, z uwagi na otrzymany przez osobę 4 negatywny wynik testu. Osoby 3 i 4 muszą być zdrowe. Odpowiedzią dla zapytania jest więc `TAK`, ponieważ można udowodnić, że wszystkie osoby przebywające poza kwarantanną (2, 3 i 4) są zdrowe. Zmienna *shift* przyjmuje wartość 0.

- Po trzecim zapytaniu, osoba 2 otrzymuje negatywny wynik testu. Jak łatwo możesz zauważyć, ta część wejścia nie ma *de facto* znaczenia: po pierwszej odpowiedzi `TAK`, epidemia jest już opanowana i odpowiedź `TAK` musi powtarzać się do końca zestawu danych.

Zadanie F: Farba

Limit czasowy: 3s, limit pamięciowy: 1GB.

Bajtazar powszechnie uważany jest za największego skąpca w całej gminie. Na poparcie tej tezy można przytoczyć wiele przykładów, najmniej istotnym spośród nich jest zaś to, że jego posesja nie posiada nawet płotu. Ostatnio jednak odnalazł on w piwnicy n starych desek, postanowił więc wykonać chociaż kawałek ogrodzenia.

Bajtazar położył deski na stosie tak, że kolejne z nich miały długości $a_1, \ldots, a_n$. Wziął pierwszą deskę, odciął z niej fragment o długości b i przybił jako pierwszą sztachetę, następnie z pozostałej części odciął kolejny fragment o długości b i przybił tuż obok. Kontynuował ten proces, aż w rękach pozostała mu resztka deski, której długość mieściła się w przedziale $[1, b]$. *Cóż, taka dobra deska nie może się wszak zmarnować, mimo że zdaje się być trochę przykrótka*, pomyślał Bajtazar... i również przybił ją do płotu jako kolejną sztachetę. Następnie wziął drugą deskę ze stosu, a potem kolejną, i dla każdej z nich powtórzył opisaną procedurę.

Po ukończonej pracy, Bajtazar spojrzał na swoje dzieło i stwierdził, że przybijanie sztachet o różnych długościach mogło jednak nie być najlepszym pomysłem. *Wygląda to bardziej jak zbiór losowych desek, niż jak przemyślana konstrukcja*, pomyślał. Postanowił więc pomalować płot białą farbą, licząc że dzięki temu będzie wyglądał on choć odrobinę bardziej profesjonalnie. *Ale*, zdał sobie sprawę po chwili, *jeżeli pomaluję na biało tylko co drugą sztachetę, resztę zaś pozostawię brązową, to zużyję (około) dwukrotnie mniej farby, a ogrodzenie nadal będzie sprawiało wrażenie spójnej i przemyślanej całości!*

Jak pomyślał, tak zrobił, i pomalował w ogrodzeniu tylko co drugą sztachetę, rozpoczynając od pierwszej [1]. Dopiero przed snem Bajtazarowi przyszła do głowy przerażająca myśl: być może gdyby wybrał inną długość b, to zużyłby mniej farby? Cóż, teraz już niewiele da się zrobić, lecz sama świadomość możliwego błędu nie daje Bajtazarowi spokoju. Zastanawia się więc, ile farby musiałby zużyć, gdyby zdecydował się na zbudowanie płotu o innej spośród możliwych wysokości.

Pomóż Bajtazarowi rozwiązać ten problem i spraw, aby mógł on w końcu spokojnie (bądź niespokojnie, zależnie od wyniku Twoich obliczeń) usnąć.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 5$). Potem kolejno podawane są zestawy w następującej postaci:

W pierwszej linii zestawu znajduje się liczba całkowita n ($1 \leqslant n \leqslant 1\,000\,000$). W drugiej linii znajduje się n liczb całkowitych a_i ($1 \leqslant a_i \leqslant 1\,000\,000$, $\sum_{i=1}^{n} a_i \leqslant 1\,000\,000$), oznaczających długości kolejnych desek.

Wyjście

Niech M oznacza największą spośród wszystkich wartości a_i w wybranym zestawie danych. Na wyjściu dla tego zestawu danych wypisz M linii. W i-tej z nich powinna znajdować się jedna liczba całkowita f_i: sumaryczna długość sztachet, które Bajtazar musiałby pomalować na biało, gdyby zdecydował się na budowanie płotu o wysokości $b = i$.

[1] Jak widać, plotki o skąpstwie Bajtazara były cokolwiek przesadzone. Mógł przecież zacząć malowanie od drugiej sztachety.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
1 4 10 7 2 8	14 13 15 13 15 16 21 23 24 12

Wyjaśnienie

Dla wysokości płotu $b = 4$, kolejne sztachety miałyby wysokości: 4 4 2 4 3 2 4 4. Bajtazar musiałby pomalować deski o długościach 4, 2, 3 oraz 4, więc odpowiedzią w czwartej linii jest 13.

Dla wysokości płotu $b = 5$, kolejne sztachety miałyby wysokości: 5 5 5 2 2 5 3. Bajtazar musiałby pomalować deski o długościach 5, 5, 2 oraz 3, więc odpowiedzią w piątej linii jest 15.

Zadanie G: Glątwa Gebajta

Limit czasowy: 12s, limit pamięciowy: 1GB.

Jak co wiosnę, Wiedźmak Gebajt na szlak swój wyrusza, coby użyć nieco wiedźmińskiego fachu, a i groszem kmiotów prostych sakiewkę wypchać. Szlak wiedźmaka od zachodu na wschód się ciągnie, długi na n stajań, co stajanie inny obiekt czeka na niego, o charakterze trojakim:

- B b_i: Leże bestyji srogiej, co na chłopkach żer bezdusznie prowadzi. Gdy wiedźmak do potwory zawita, ta go zoczy, szponem a kłem rani niechybnie, i b_i żywotności odbierze. Jeśli żywotność wiedźmaka do zera zejdzie samego, scześnie on; jeśli zaś nie, brzeszczotem swym potworzycę wnet tnie, i na miejscu ubije. Żywotność Gebajta tedy zmieni się jako

  if $H \leqslant b_i$ then *death* else $H := H - b_i$.

- K k_i: Karczma wioskowa, gdzie Gebajt (wielce na gorzałkę łasy) wstąpić nie omieszka. Jeśli z żywotnością mniejszą niźli k_i do oberży zawita, trunkiem nad umiar upity zemrze niechybnie. W przeciwnym zaś razie, glątwą zmożony, z żywotnością do k_i pomniejszoną przybytek o brzasku opuści. Żywotność Gebajta zmieni się jako

  if $H < k_i$ then *death* else $H := k_i$.

- C c_i: Czarownicy potężnej chata, w urokach a eliksirach obeznanej, co rany zdolne są zabliźnić, glątwę zaś uleczyć. Jeśli Gebajt z żywotnością niższą jak c_i do wiedźmy przybieży, ta czaramy swemi żywotność do poziomu c_i przywrócić mu zdoła. Żywotność Gebajta zmieni się jako

  $H := max(H, c_i)$.

Duma tedy wiedźmak, jaki to fragment szlaku obrać, ażeby radości wiedźmińskiej zaznać, ale życie swe zachować. Dzień za dniem mija, a dnia i-tego jedna z dwóch rzeczy się dzieje:

- Jeden obiekt na szlaku zmianie ulega, przykładem chatkę czarownicy miejscowy kupiec nabywa i w karczmę zmienia, albo też nowa bestyja spod ziemi wyłazi, karczmę ogniem z pyska pali i leże w tym miejscu zakłada;
- Gebajt przed dom wychodzi, pod ulubionym drzewem siada i duma: gdyby od obiektu l_i wędrować zaczął, i na wschód jechał, jak daleko zajechać by zdołał życia nie tracąc? W dumaniu swym Twojej pomocy Gebajt szuka, cobyś tajemną sztuką kodów zaklinania odpowiedź na jego pytanie znalazł.

Bacz, że wiedźmak duma jeno co począć, a nie po prawdzie na szlak rusza, tedy **zmiany na szlaku na stałe zostają**, ale **każde wiedźmaka dumanie niezależne od inszych ostaje**, i w każdym żywotność Gebajta spoczątku H_0 jednostek wynosi.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 100\,000$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera trzy liczby całkowite n, q oraz H_0 ($1 \leqslant n \leqslant 2\,000\,000$, $1 \leqslant q \leqslant 4\,000\,000$, $1 \leqslant H_0 \leqslant 10^{12}$) – długość szlaku, liczbę dni i początkową żywotność Gebajta.

Kolejne n linii zestawu opisuje początkowy stan szlaku; i-ta zawiera literę odpowiadającą typowi i-tego obiektu (B, K lub C), oraz liczbę (b_i, k_i lub c_i; $1 \leqslant b_i, k_i, c_i \leqslant 10^{12}$) o znaczeniu takim jak wyjaśniono powyżej.

Kolejne q linii zestawu opisuje poszczególne dni. Linia i-ta rozpoczyna się literą Z jeśli danego dnia następuje zmiana na szlaku, lub D jeśli następuje dumanie Gebajta.

W przypadku zmiany na szlaku, reszta linii zawiera: liczbę całkowitą x_i ($1 \leqslant x_i \leqslant n$), oznaczającą obiekt ulegający zmianie, oraz literę i liczbę, w takim samym formacie jak przy opisie stanu początkowego, oznaczające nowy obiekt. W przypadku dumania Gebajta, linia zawiera jedną liczbę całkowitą l_i ($1 \leqslant l_i \leqslant n$), oznaczającą stajanie od którego zaczyna wędrować.

Łączna długość szlaków oraz łączna liczba dni nie przekroczą odpowiednio 2 000 000 i 4 000 000.

Wyjście

Dla każdego zestawu danych wypisz odpowiedzi na wszystkie zapytania. Dla każdego zapytania wypisz jedną liczbę całkowitą, oznaczającą indeks najdalszego obiektu r_i ($l_i \leqslant r_i \leqslant n$) do którego Gebajt może dojechać (i opuścić go żywym), lub -1 jeśli zginie już w starciu na pozycji l_i. Odpowiedź na zapytanie i-tego dnia powinna brać pod uwagę wszystkie zmiany z dni wcześniejszych.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
1 4 12 10 C 10 B 5 K 5 B 6 Z 3 K 6 Z 1 C 11 D 2 D 1 Z 3 C 1 D 3 Z 3 B 20 D 3 Z 1 C 31 D 1 Z 4 K 6 D 1	2 3 4 -1 3 4

Wyjaśnienie

Szlak Gebajta zmienia się sześciokrotnie, w poniższy sposób:

- `[C 10, B 5, K 5, B 6]` (początek)
- `[C 10, B 5, K 6, B 6]` (1 dzień)
- `[C 11, B 5, K 6, B 6]` (2 dzień)
- `[C 11, B 5, C 1, B 6]` (5 dzień)
- `[C 11, B 5, B 20, B 6]` (7 dzień)
- `[C 31, B 5, B 20, B 6]` (9 dzień)
- `[C 31, B 5, B 20, K 6]` (11 dzień)

Gebajt duma podczas sześciu pozostałych dni.

Trzeciego dnia Gebajt zaczyna od drugiego obiektu. Po pokonaniu bestii pozostaje mu 5 punktów żywotności, czyli o 1 za mało aby przetrwać kolejny obiekt (karczmę `K 6`). Najdalej jest więc w stanie dotrzeć do drugiego obiektu.

Czwartego dnia Gebajt zaczyna od pierwszego obiektu; dzięki czarownicy ma jeden punkt żywotności więcej, i przeżywa obiekt trzeci (karczmę), nie dając rady ostatniemu (bestii).

Szóstego dnia Gebajt zaczyna od trzeciego obiektu, który teraz jest czarownicą `C 1`. Z niezmienioną liczbą punktów żywotności podąża dalej i pokonuje bestię (ostatni obiekt na szlaku).

Ósmego dnia Gebajt zaczyna od bestii `B 20` której nie jest w stanie pokonać, więc odpowiedź to `-1`.

Dziesiątego dnia Gebajt zaczyna od potężnej `C 31`, dzięki której pokonuje dwie pierwsze bestie, nie dając jednak rady ostatniej.

Ostatniego dnia, Gebajt jest w stanie przejechać cały szlak.

Zadanie H: Hasła

Limit czasowy: 5s, limit pamięciowy: 1GB.

Po udanym stażu Bajtek został zatrudniony jako ekspert ds. cyberbezpieczeństwa. Aby samemu być dobrym przykładem dla innych, zdecydował się ustawić sobie inne hasło do poczty e-mail, zaś inne do serwisu społecznościowego *Facepalm*. Niestety zapamiętanie dwóch różnych haseł okazało się trudne, a zapisanie ich wprost na kartce naruszałoby zasady bezpieczeństwa ustalone przez niego samego. Przezorny Bajtek wybrał więc pewną sekretną liczbę $d > 0$ i zapisał na kartce oba swoje hasła zakodowane szyfrem Cezara [1] z przesunięciem d. Zadowolony z siebie spojrzał na swoją karteczkę i zdębiał: po zakodowaniu jego hasło do *Facepalm* stało się hasłem do poczty i odwrotnie! "Co ja narobiłem..." – wykrzyknął, łapiąc się za głowę.

Ty też sprawdź się w cyberbezpieczeństwie! Mając dane pierwsze z haseł Bajtka, odtwórz drugie lub ustal, że nie da się tego w sposób jednoznaczny zrobić.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 20$). Potem kolejno podawane są zestawy w następującej postaci:

Każdy zestaw to jedno hasło składające się z małych liter alfabetu angielskiego. Hasło ma co najmniej 1 i co najwyżej 200 000 znaków.

Suma długości wszystkich haseł nie przekracza 1 000 000.

Wyjście

Dla każdego podanego hasła wypisz odpowiadające mu drugie hasło. Jeśli drugiego hasła nie da się odtworzyć (rozwiązanie nie istnieje albo jest więcej niż jedno), wypisz zamiast tego pojedyncze słowo `NIE`.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
`1` `cnffjbeq`	`password`

[1] Szyfr Cezara polega na zastąpieniu każdej litery inną, leżącą o d znaków dalej w alfabecie, z cyklicznym zawinięciem alfabetu. Na przykład dla $d = 3$ litera `a` zmienia się w `d`, litera `b` w `e`,...., litera `w` zmienia się w `z`, litera `x` w `a`, `y` w `b`, a `z` w `c`. Bajtek używa alfabetu angielskiego, choć Cezar raczej nazwałby go łacińskim.

Zadanie I: Interesujące liczby

Limit czasowy: 10s, limit pamięciowy: 1GB.

Operacja $\oplus$ nazywana jest *bitową sumą wyłączającą* albo *bitowym XORem*. Działa ona następująco: aby obliczyć dla dwóch liczb naturalnych wynik $x \oplus y$ zapisujemy obie w systemie dwójkowym, po czym i-ta cyfra dwójkowa wyniku jest równa 1 wtedy i tylko wtedy, gdy dokładnie jedna z i-tych cyfr liczb x i y jest równa 1. Innymi słowy, jeśli x_i, y_i, z_i oznaczają i-tą cyfrę odpowiednio dla x, y i z, gdzie $z = x \oplus y$, to $z_i = (x_i + y_i) \mod 2$. Cyfry numerujemy od najmniej znaczących.

Dana jest liczba całkowita dodatnia k. Ciąg liczb nazwiemy *interesującym*, jeśli dla każdych dwóch jego elementów ich bitowy XOR jest nie większy niż k. Mając dany ciąg liczb, wybierz z niego jak najwięcej elementów tak, aby wybrane liczby tworzyły interesujący ciąg.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 1000$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera dwie liczby całkowite n, k ($1 \leqslant n \leqslant 30\,000$, $1 \leqslant k < 2^{20}$), oznaczające długość ciągu i liczbę określającą maksymalny XOR dwóch elementów.

Druga linia zestawu zawiera dany ciąg n liczb całkowitych nieujemnych, mniejszych niż 2^{20}, oddzielonych spacjami.

Suma długości ciągów we wszystkich zestawach danych nie przekracza 200 000. Suma liczb k we wszystkich zestawach nie przekracza 3 200 000.

Wyjście

Dla każdego zestawu danych wypisz jedną liczbę całkowitą – maksymalną możliwą liczbę elementów, jakie można wybrać z wejściowego ciągu tak, aby tworzyły interesujący ciąg.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
1 7 11 3 12 9 10 16 3 4	4

Wyjaśnienie

Ciąg interesujący tworzą elementy 3, 9, 10 i 3, ponieważ XOR bitowy każdej pary nie przekracza 11. Na przykład $9 \oplus 10 = 1001_2 \oplus 1010_2 = 11_2 = 3 \leqslant 11$. Nie da się wybrać pięciu (ani więcej) elementów o tej własności: na przykład ciąg $(3, 9, 10, 3, 4)$ nie nadaje się ze względu na $4 \oplus 9 = 100_2 \oplus 1001_2 = 1101_2 = 13 > 11$.

Zadanie J: Jadowite węże

Limit czasowy: 15s, limit pamięciowy: 1GB.

Plansza do gry *Jadowite węże* jest prostokątem o n wierszach i m kolumnach, podzielonym na $n \cdot m$ jednostkowych pól. Każde pole może być puste, zablokowane lub zawierać siedlisko jadowitych bądź niejadowitych węży. Gracz może *odwrócić* dowolną liczbę wybranych przez siebie wierszy: wykonanie takiej operacji sprawia, że każde znajdujące się w danym wierszu siedlisko jadowitych węży zamienia się w siedlisko węży niejadowitych, i vice versa. Analogicznie, gracz może również *odwrócić* wybrane przez siebie kolumny. Jeżeli jakieś siedlisko zostanie odwrócone dwukrotnie, powraca do swojego oryginalnego stanu. Po wykonaniu wszystkich tych akcji gracz musi przejść z lewego górnego do prawego dolnego pola planszy, w każdym ruchu przechodząc o jedno pole w prawo lub w dół. Ścieżka gracza nie może przechodzić przez pola zablokowane, ani przez pola z jadowitymi wężami.

Twórcy gry zaimplementowali już z proponowanych plansz. Konieczna jednak jest jeszcze weryfikacja, które z nich da się w ogóle rozwiązać. Niestety, to zadanie zostało przydzielone właśnie Tobie.

Wejście

Pierwsza linia wejścia zawiera liczbę plansz z ($1 \leqslant z \leqslant 500$). Opis każdej z nich jest następującej postaci:

W pierwszej linii znajdują się dwie liczby całkowite n oraz m ($2 \leqslant n, m \leqslant 2\,000$).

Każda z kolejnych n linii zawiera dokładnie m znaków `.`, `#`, `O` (duża litera “o”) oraz `@` (małpa), oznaczających odpowiednio puste pole, zablokowane pole, pole z siedliskiem niejadowitych węży oraz pole z siedliskiem jadowitych węży. Możesz założyć, że pierwszy znak pierwszej linii oraz m-ty znak n-tej linii są różne od `#`, tj. lewe górne ani prawe dolne pole nie są zablokowane.

Suma wartości n oraz suma wartości m we wszystkich zestawach nie przekraczają $15\,000$ każda.

Wyjście

Dla każdego zestawu danych wypisz rozwiązanie w następującym formacie.

W pierwszej linii wypisz pojedynczy napis `TAK` lub `NIE` mówiący, czy daną planszę da się rozwiązać.

Jeżeli Twoją odpowiedzią dla danego zestawu jest `TAK`, w następnych trzech liniach wypisz kolejno:

- Ciąg n znaków `T` lub `N`, gdzie i-ty znak oznacza odpowiednio odwrócenie bądź nieodwrócenie siedlisk znajdujących się w i-tym wierszu planszy;
- Ciąg m znaków `T` lub `N`, gdzie j-ty znak oznacza odpowiednio odwrócenie bądź nieodwrócenie siedlisk znajdujących się w j-tej kolumnie planszy;
- Ciąg $n + m - 2$ znaków `P` lub `D`, oznaczających że kolejne kroki ścieżki rozpoczynającej się w lewym górnym polu planszy prowadzą odpowiednio w prawo bądź w dół. Opisana przez Ciebie ścieżka musi prowadzić do prawego dolnego pola planszy i może używać jedynie pustych pól oraz pól z siedliskami niejadowitych węży.

Jeśli istnieje wiele poprawnych rozwiązań, możesz wypisać dowolne z nich.

Przykład

Dla danych wejściowych:	Możliwą poprawną odpowiedzią jest:

`1`
`4 5`
`..#..`
`@@O@@`
`##@#O`
`..@.@`
	`TAK`
	`NTNN`
	`NNTNT`
	`DPPDDPP`

Wyjaśnienie

Po odwróceniu wskazanych przez gracza wierszy i kolumn stan planszy przedstawia się następująco:

```
..#..
OOOO@
##O#@
..O.O
```

Wskazana przez gracza ścieżka używa jedynie pól pustych oraz pól, na których (po wykonaniu wszystkich odwróceń) żyją węże niejadowite.

Zadanie K: Kot i Roomba

Limit czasowy: 12s, limit pamięciowy: 1GB.

Kot Bitusia, Kapitan, najbardziej na świecie lubi spać. Niestety, jakość jego snu znacząco spadła od kiedy Bituś zdecydował się na kupno Roomby – robota do odkurzania pokojów. Jak się bowiem okazało, Kapitan Kot boi się Roomby jak... no, po prostu bardzo się boi.

Dom Bitusia zawiera n pokojów połączonych $n-1$ dwukierunkowymi korytarzami w taki sposób, że z każdego pokoju da się dojść do każdego innego. Bituś zauważył, że jeśli Roomba wjeżdża do pokoju z Kapitanem, to kot budzi się i ucieka do jednego z sąsiednich pokojów, gdzie natychmiast ponownie zasypia. Spłoszony Kapitan ucieka całkowicie na oślep, jeśli więc z pokoju wychodzi więcej niż jeden korytarz, to *wybór każdej z opcji jest tak samo prawdopodobny* (w szczególności, może on uciec do tego pokoju, z którego właśnie przyjechała Roomba).

Podczas kolejnej długiej nocy w pracy, Bituś otworzył aplikację do obsługi Roomby i zobaczył, że w trakcie sprzątania zwiedziła ona kolejno pokoje $a_1, \ldots, a_m$. Pokoje w tym ciągu mogą się powtarzać, a każde dwa sąsiednie są połączone korytarzem. Bituś pamięta też, że przed włączeniem Roomby kot spał w pokoju c. Co więcej, zachodzi $a_1 \neq c$, gdyż przezorny Kapitan nigdy nie sypia w jednym pokoju z Roombą!

Teraz Bituś zastanawia się, jaka jest *wartość oczekiwana* liczby razy, gdy Roomba zbudziła Kapitana podczas sprzątania. Pomóż odpowiedzieć na dręczące go pytanie, aby mógł znów skupić się na pracy.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 6\,000$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera dwie liczby całkowite n, c ($2 \leqslant n \leqslant 1\,000\,000$, $1 \leqslant c \leqslant n$), oznaczające liczbę pokojów w domu Bitusia oraz pokój, w którym początkowo śpi Kapitan Kot.

Kolejne $n-1$ linii opisuje korytarze. Każda z nich zawiera dwie liczby całkowite u_i, v_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$) oznaczające, że pokoje u_i oraz v_i są połączone korytarzem. Możesz założyć, że z każdego pokoju da się dojść do każdego innego.

Kolejna linia zestawu zawiera liczbę pokojów m ($1 \leqslant m \leqslant 5\,000\,000$) odwiedzonych przez Roombę.

W ostatniej linii zestawu znajduje się ciąg m liczb całkowitych a_i ($1 \leqslant a_i \leqslant n$) – ciąg pokojów odwiedzonych przez Roombę. Każde dwa kolejne pokoje są połączone korytarzem, zachodzi też $a_1 \neq c$.

Suma wartości $n+m$ we wszystkich zestawach nie przekroczy $12\,000\,000$.

Wyjście

Dla każdego zestawu danych wypisz jedną liczbę rzeczywistą – wartość oczekiwaną liczby obudzeń Kapitana Kota przez Roombę. Aby odpowiedź została uznana za poprawną wystarczy, by błąd względny lub bezwzględny nie przekraczał 10^{-5}. Innymi słowy, jeśli Twój algorytm odpowie a, zaś poprawna odpowiedź to b, to wystarczy, by zachodziło $\frac{|a-b|}{\max(1,b)} \leqslant 10^{-5}$.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
1 4 2 1 2 2 3 4 2 4 1 2 3 2	1.666666666666667

Zadanie L: Leniwce

Limit czasowy: 10s, limit pamięciowy: 1GB.

Wyruszasz na wyprawę do dżungli w celu obserwacji mało dotychczas zbadanego gatunku leniwców *Choloepus manhattani*. Cały obszar dżungli, na którym żyją leniwce, to jedno z najdziwniejszych miejsc na świecie: drzewa rosną tam ustawione w idealny prostokąt $n \times m$. Na Twojej mapie oznaczone są parami liczb naturalnych – drzewo (i, j) rośnie na przecięciu i-tego wiersza oraz j-tej kolumny. Wszystkie leniwce z gatunku *Choloepus manhattani* żyją na tym obszarze.

Każdy leniwiec ma swoje stałe legowisko na jednym z drzew, ale czasem wyrusza z niego na okoliczne drzewa w poszukiwaniu pożywienia. Po dżungli leniwce poruszają się wyłącznie przeskakując z drzewa na drzewo – w jednym skoku leniwiec przenosi się na drzewo, które sąsiaduje z poprzednim w poziomie lub w pionie. Aby nie oddalić się za bardzo od legowiska, każdy leniwiec żeruje tylko w zasięgu k skoków od niego. Innymi słowy, jeśli legowisko jest na drzewie (x, y), to obszarem żerowania leniwca jest zbiór drzew o współrzędnych (x', y') spełniających $|x - x'| + |y - y'| \leqslant k$ (oraz $1 \leqslant x' \leqslant n$, $1 \leqslant y' \leqslant m$). Stała k jest wspólna dla wszystkich leniwców, ustalona przez miliony lat ewolucji.

Masz pewne wątpliwości co do tych fantastycznych doniesień, ale nie będziesz w stanie ich skonfrontować z poprzednim badaczem, bowiem ten pewnego dnia zniknął w dżungli w niewyjaśnionych okolicznościach (co między innymi skłoniło Cię do przemyśleń, czy leniwce na pewno są roślinożerne...). Pozostała po nim jedynie mapa o wymiarach $n \times m$, na której odpowiednio oznaczone są wszystkie drzewa, na których żerują leniwce. Na mapie nie zostały jednak zaznaczone ich legowiska.

Sprawdź, czy mapa w ogóle może być poprawna – rozstrzygnij, czy istnieje taki zbiór legowisk, dla którego obszary żerowania leniwców będą dokładnie odpowiadały tej mapie.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 4\,000$). Potem kolejno podawane są zestawy w następującej postaci:

W pierwszej linii zestawu znajdują się liczby całkowite n, m, k ($1 \leqslant n, m, k \leqslant 1\,000$) o znaczeniu podanym w treści zadania.

W kolejnych n liniach znajduje się opis mapy – po n znaków w każdej linii. Jeżeli według mapy leniwce żerują na drzewie (i, j), to w i-tej linii na j-tej pozycji znajdzie się znak `x`, w przeciwnym wypadku znajdzie się tam znak `.` (kropka).

Suma wartości $n + m + k$ we wszystkich zestawach nie przekracza $100\,000$.

Wyjście

Dla każdego zestawu danych wypisz pojedynczy napis `TAK` lub `NIE` oznaczający odpowiedź na pytanie, czy pozostawiona przez poprzedniego badacza mapa może opisywać poprawne obszary żerowania leniwców.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
2 3 3 1 .xx xxx xx. 3 4 1 ..xx x.xx x..x	TAK NIE

Wyjaśnienie

W pierwszym teście zaznaczony obszar jest poprawnym terenem żerowania dla trzech leniwców, jeśli przyjmiemy, że zamieszkują pola $(1, 3)$, $(2, 2)$ oraz $(3, 1)$.

W drugim teście nie istnieje zbiór pól zamieszkanych przez leniwce, który generowałby zaznaczony obszar żerowania.

Zadanie M: Magiczne trójki

Limit czasowy: 20s, limit pamięciowy: 1GB.

Siedzisz na egzaminie i zastanawiasz się, jak wyznaczyć medianę ciągu. Najwyższym wysiłkiem przypominasz sobie, że mediana to element ciągu, który po posortowaniu go niemalejąco zajmowałby środkową pozycję (jeśli długość ciągu jest parzysta, medianą jest mniejszy z dwóch środkowych elementów). Egzamin polega na napisaniu pseudokodu rozwiązania, po czym zasymulowaniu jego działania na przykładowym, podanym przez profesora, ciągu wejściowym.

Mimo dziur w pamięci kojarzysz, że coś takiego pojawiło się na wykładzie. Niestety, z wykładu pamiętasz tylko mgliste fragmenty (noce zarwane nad BajtStation i monotonny głos profesora algorytmiki nie pomagały w skupieniu uwagi). Był jakiś magiczny algorytm... magiczne trójki? Jakoś dzieli się ten ciąg, wywołuje rekurencyjnie, i potem łączy...?

Na bazie fragmentów, które udało Ci się zapamiętać, wymyśliłaś następujący algorytm:

```
funkcja magiczneTrójki(ciąg)
  jeżeli długość ciągu jest nie większa niż 2 to
    zwróć najmniejszą wartość w ciągu
  w przeciwnym razie
    część_1, część_2, część_3 = podzielNaTrzyCzęści(ciąg)
    mediana_i = magiczneTrójki(część_i) dla i = 1, 2, 3
    zwróć medianę ciągu [mediana_1, mediana_2, mediana_3]
```

gdzie `podzielNaTrzyCzęści` dzieli ciąg na trzy spójne fragmenty o jak najbardziej zbliżonych do siebie długościach. Konkretnie, kolejne fragmenty mają długości `[s, s, s]`, `[s + 1, s, s]` lub `[s + 1, s + 1, s]`, w zależności od długości ciągu. Na przykład, ciąg `[8, 2, 6, 6, 3, 5, 7, 1]` podzieli się na `[8, 2, 6]`, `[6, 3, 5]` i `[7, 1]`.

Dopiero po wyjściu z egzaminu zdałaś sobie sprawę, że Twój algorytm nie jest aż tak magiczny, bo nie zawsze działa. *Może przynajmniej działał poprawnie dla ciągu z zadania*, myślisz z nadzieją... Niestety, Twoja pamięć w tej kwestii jest równie rozmyta co w przypadku samego algorytmu: wprawdzie pamiętasz *prawie wszystkie* elementy ciągu profesora, ale niektórych wartości nie jesteś pewna. Pamiętasz jedynie podane w treści zadania ograniczenie na wielkość danych: wszystkie liczby w ciągu miały zawierać się w (domkniętym) przedziale $[0, m-1]$.

Oblicz, na ile sposobów można wpisać nieznane Ci liczby tak, aby algorytm `magiczneTrójki` dla otrzymanego ciągu zwracał jego prawdziwą medianę (zdefiniowaną powyżej). Ponieważ liczba ta może być duża, wystarczy jeśli podasz jej resztę z dzielenia przez $10^9 + 7$.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z. Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera dwie liczby całkowite n, m ($n \geqslant 1$, $1 \leqslant m \leqslant 10^9$), oznaczające długość ciągu testowego i ograniczenie na wartości jego elementów.

Druga linia zestawu zawiera ciąg testowy opisany przez n liczb całkowitych z przedziału $[-1, m-1]$, gdzie wartości -1 oznaczają elementy o nieznanych wartościach.

Testy

Jeśli przez q oznaczymy liczbę elementów o nieznanych wartościach, to każdy plik testowy należy do jednej z trzech poniższych grup:

- $1 \leqslant z \leqslant 100$, $1 \leqslant q \leqslant 10$, $n \leqslant 3^4 = 81$
- $z = 15$, $1 \leqslant q \leqslant 20$, $n \leqslant 3^5 = 243$
- $z = 3$, $q = 30$, $n \leqslant 3^8 = 6\,561$

Wyjście

Dla każdego zestawu danych wypisz jedną liczbę całkowitą r ($0 \leqslant r < 10^9 + 7$) – odpowiedź na pytanie z treści zadania.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
3 3 10 -1 -1 3 4 50 10 20 -1 40 5 100 -1 10 10 -1 20	100 21 1979

Wyjaśnienie

W pierwszym teście algorytm `magiczneTrójki` zwraca poprawną medianę niezależnie od wartości dwóch nieznanych elementów ciągu; odpowiedzią jest więc $10^2 = 100$.

W drugim teście zwraca poprawną medianę tylko jeśli nieznana wartość jest nie większa niż 20, co daje 21 możliwości.

W trzecim teście zwraca *niepoprawną* medianę jeśli obie nieznane wartości są mniejsze niż 10, lub obie większe, co daje $100^2 - (10^2 + 89^2) = 1979$ możliwości.