A: Aparat na dronie

Limit pamięci: **256 MB**

Małgosia dostała w prezencie drona z aparatem fotograficznym. Na próbę chce nim zrobić zdjęcia dwóm przebiegającym w pobliżu jej domu prostym (i co ciekawsze nieskończonym) drogom, przy czym najlepsze zdjęcia dróg robi się bezpośrednio nad nimi. Nie wie jeszcze, na ile lotu wystarcza baterii, dlatego chciałaby, aby trasa przelotu była jak najkrótsza. Oczywiście dron musi zacząć i skończyć lot w domu Małgosi! Pomóż Małgosi zaplanować najkrótszą możliwą trasę dla jej drona.

Wejście

W pierwszym wierszu wejścia znajdują się trzy liczby całkowite a, b, c ($-1\,000\,000 \leqslant a, b, c \leqslant 1\,000\,000$, $a \neq 0$ lub $b \neq 0$), pooddzielane pojedynczymi odstępami, opisują one pierwsza drogę, która spełnia równanie $ax + by = c$. W drugim i ostatnim wierszu wejścia znajduje się analogiczny opis drugiej drogi.

Drogi są różne, to znaczy nie pokrywają się, oraz nie przechodzą przez dom Małgosi, który znajduje się w punkcie $(0, 0)$.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczbę rzeczywistą – najmniejszą długość trasy drona. Odpowiedź zostanie zaakceptowana, jeśli błąd względny lub bezwzględny wynosi najwyżej 10^{-6}.

Przykład

Wejście	Wyjście
`1 0 2` `-1 1 1`	`5.099019513593`

B: Bony Fibonacciego

Limit pamięci: **256 MB**

W szkole Jasia organizowany jest festyn upamiętniający Fibonaccich. W czasie festynu wprowadzone zostaną specjalne bony o nominałach będących (dodatnimi) liczbami Fibonacciego, które można wydać w sklepiku prowadzonym przez Jasia. Dziwne nominały sprawiają Jasiowi dużo problemów, postanowił więc, że będzie od klientów przyjmował dokładnie k bonów, niekoniecznie różnych nominałów; nie ma też zamiaru wydawać reszty. Pozostaje ustalić ceny – w sklepie jest n różnych rzeczy, a Jasio chciałby, żeby każdy miała inną cenę, zaczął więc mozolnie wyliczać kolejne możliwe ceny. Jedną cenę można czasami zapłacić na wiele różnych sposobów, w takim przypadku Jasio wciąż traktuje ją jako jedną cenę. Jasio chciałby wiedzieć, czy się nie pomylił – nie zamierza sprawdzać wszystkich cen, wystarczy mu, że powiesz, ile wynosi ostatnia, n-ta cena. Pomóż Jasiowi i napisz program, który dla danego k oraz n obliczy n-tą najmniejszą cenę, którą można zapłacić przy użyciu dokładnie k bonów.

Wejście

W pierwszym i jedynym wierszu wejścia znajdują się dwie liczby naturalne k i n ($1 \leqslant k \leqslant 100, 1 \leqslant n \leqslant 10^{18}$), oddzielone pojedynczym odstępem.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać n-tą najmniejszą cenę, którą można zapłacić przy użyciu pewnych (niekoniecznie różnych) k bonów o nominałach będących liczbami Fibonacciego, o ile jest ona nie większa niż 10^{18}, lub `NIE`, jeśli jest ona większa niż 10^{18}.

Przykład

Wejście	Wyjście
`2 11`	`13`

Przy pomocy dokładnie dwóch bonów klienci Jasia nie są w stanie zapłacić cen 1 i 12.

Wejście	Wyjście
`1 100`	`NIE`

Liczba Fibonacciego numer 100 jest (dużo) większa niż 10^{18}.

C: Chińskie twierdzenie o resztach

Limit pamięci: **256 MB**

Jasio, student informatyki, w tym semestrze uzyskał wielką biegłość w stosowaniu Chińskiego Twierdzenia o Resztach. Właśnie przypadkiem usłyszał pod salą, że Małgosia nie potrafi rozwiązać zadania na ćwiczenia; wychwycił słowa kluczowe: reszta modulo, układ równań, ... Chcąc zaimponować koleżance bez wahania powiedział, że on potrafi zrobić to zadanie. Po chwili trochę zrzedła mu mina, bo zadanie Małgosi jest zupełnie inne niż te, które potrafi rozwiązywać. Układ Małgosi jest postaci:

$$\begin{cases} a_1 & \equiv & b_1 & \pmod{m} \\ a_2 & \equiv & b_2 & \pmod{m} \\ \vdots & \vdots & & \vdots \\ a_n & \equiv & b_n & \pmod{m} \end{cases}$$

(gdzie $\equiv$ oznacza przystawanie modulo) i dla podanych $a_1, b_1, \ldots, a_n, b_n$ ma ona wyznaczyć największą wartość m, dla której układ równań jest spełniony. Dla ułatwienia zadania Małgosia uporządkowała już strony równań w taki sposób, że dla każdego i zachodzi $a_i \geqslant b_i$. Jasio nie może stracić twarzy – pomóż mu rozwiązać to zadanie.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \leqslant n \leqslant 100\,000$) oznaczająca liczbę równań.

W drugim wierszu znajduje się n liczb całkowitych $a_1, a_2, \ldots, a_n$, pooddzielanych pojedynczymi odstępami, są to liczby występujące po lewej stronie kolejnych równań.

W trzecim wierszu wejścia znajduje się n liczb całkowitych $b_1, b_2, \ldots, b_n$, pooddzielanych pojedynczymi odstępami, są to liczby występujące po prawej stronie kolejnych równań.

Dla każdego i ($1 \leqslant i \leqslant n$) zachodzi $0 \leqslant b_i \leqslant a_i \leqslant 10^{18}$. Układ równań jest też nietrywialny: dla pewnego i ($1 \leqslant i \leqslant n$) zachodzi $a_i \neq b_i$.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać największą liczbę m, dla której podany układ równań jest spełniony.

Przykład

Wejście	Wyjście
3 7 17 9 3 5 1	4

Układ równań

$$\begin{cases} 7 & \equiv & 3 & \pmod{4} \\ 17 & \equiv & 5 & \pmod{4} \\ 9 & \equiv & 1 & \pmod{4} \end{cases}$$

jest spełniony i łatwo sprawdzić, że nie jest spełniony dla $m > 4$.

Wejście	Wyjście
3 4 6 5 2 2 2	1

Układ równań

$$\begin{cases} 4 & \equiv & 2 & \pmod{1} \\ 6 & \equiv & 2 & \pmod{1} \\ 5 & \equiv & 2 & \pmod{1} \end{cases}$$

jest spełniony i łatwo sprawdzić, że nie jest spełniony dla $m > 1$.

D: Droga

Limit pamięci: **512 MB**

Jasio został kontrolerem w Centrum Dróg. Jego zadaniem jest zbadanie efektywności odśnieżania pewnej drogi w czasie serii zamieci; standardowo droga podzielona jest na następujące bezpośrednio po sobie kilometrowe odcinki, ponumerowane kolejnymi liczbami naturalnymi od 1 do n. Jasio zabrał się szybko do pracy i skompletował już informacje o istotnych zdarzeniach:

- Centrum Meteorologii dostarczyło Jasiowi informacji o zamieciach. Intensywność zamieci określana jest przez dwa parametry f, g, które oznaczają, że w i-tej minucie ($i \geqslant 1$) od rozpoczęcia takiej zamieci spada na długości całej drogi $f \cdot i + g$ milimetrów nowego śniegu. Każda zamieć kończy się w minucie poprzedzającej pierwszą minutę kolejnej zamieci. Czas kontroli jest tak dobrany, że pierwsza z zamieci rozpoczęła się w dodatniej minucie, a w minucie 0 na całej drodze nie ma żadnego śniegu.
- Centrum Odśnieżania dostarczyło Jasiowi informacji o pracy pługów i piaskarek. Przejazd pługu w minucie t sprawia, że na koniec minuty t na całej trasie przejazdu nie ma śniegu. Analogicznie, rozsypanie przez piaskarkę soli jakości s sprawia, że na końcu każdej minuty $t, t+1, \ldots, t+s$ na całej trasie przejazdu nie ma śniegu. Działania różnych soli, nawet tej samej jakości, są niezależne i nie wpływają na siebie, a przejazd pługu nie usuwa soli, która jest obecnie na drodze. Każda trasa przejazdu piaskarki lub pługu składa się z pewnej liczby kilometrowych odcinków drogi o kolejnych numerach.
- Centrum Dróg przysłało wygenerowane przez siebie zapytania. Dla zapytania o minutę t należy podać wysokość w milimetrach najwyższej pokrywy śnieżnej na końcu danej minuty na danym fragmencie drogi składającym się z pewnej liczby kilometrowych odcinków drogi o kolejnych numerach.

Jasio wstępnie obrobił i posortował dane. Niestety, dokonanie samych obliczeń go przerasta. Pomóż mu! Napisz program, który wylicza odpowiedzi na zapytania Centrum Dróg.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne n i q ($1 \leqslant n \leqslant 10^9, 1 \leqslant q \leqslant 300\,000$) oddzielone pojedynczym odstępem, oznaczające odpowiednio liczbę kilometrowych odcinków drogi i liczbę zdarzeń. W każdym z kolejnych q wierszy znajduje się opis zdarzenia jednego z czterech poniższych typów:

- `t L a b`, które oznacza, że pług przejechał w minucie t od a-tego do b-tego odcinka drogi (włącznie).
- `t S a b s`, oznaczające przejazd piaskarki w minucie t od a-tego do b-tego odcinka drogi (włącznie) i rozsypania soli jakości s.
- `t ? a b`, oznaczające, że Centrum Dróg chce poznać największą wysokość pokrywy śnieżnej w minucie t na fragmencie drogi od a-tego do b-tego odcinka.
- `t B f g`, oznaczające, że minuta t jest ostatnią minutą poprzedniej zamieci (o ile ta istnieje), zaś minuta $t + 1$ jest pierwszą minutą zamieci o intensywności f, g.

We wszystkich zdarzeniach spełnione są warunki $1 \leqslant t \leqslant 10^9, 1 \leqslant a \leqslant b \leqslant n, 1 \leqslant s, f, g \leqslant 10^9$.
Dodatkowo, wartości t w kolejnych wierszach są rosnące, a pierwsze zdarzenie jest zawsze typu `B`.

Wyjście

Dla każdego zdarzenia typu `?` wypisz w osobnym wierszu największą wysokość w milimetrach pokrywy śnieżnej na odcinku drogi pomiędzy kilometrami a i b na koniec minuty t. Wynik podaj modulo $10^9 + 7$.

Przykład

Wejście	Wyjście
3 4 2 B 1 2 3 ? 2 2 4 L 1 3 5 ? 1 3	3 5

Poniższa tabela przedstawia wysokość pokrywy śnieżnej na poszczególnych odcinkach drogi oraz opad śniegu na koniec poszczególnych minut; wytłuszczone liczby odpowiadają zapytaniom.

minuta	1	2	3	opad
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	3	**3**	3	3
4	0	0	0	4
5	**5**	**5**	**5**	5

Do momentu określonego w pierwszym zapytaniu na całą drogę opada $1 \cdot 1 + 2 = 3$ milimetrów śniegu. Pomiędzy przejazdem pługu w minucie 4 a drugim zapytaniem opada $3 \cdot 1 + 2 = 5$ milimetrów śniegu.

Wejście	**Wyjście**

`1 3`
`1 B 1 1`
`2 B 3 3`
`3 ? 1 1`
	`8`

minuta	1	opad
0	0	0
1	0	0
2	2	2
3	**8**	6

Do momentu określonego w zapytaniu, przez pojedynczą minutę trwa pierwsza zamieć i następnie przez pojedynczą minutę druga zamieć.

Wejście	**Wyjście**

`5 5`
`1 B 1 2`
`2 S 1 3 5`
`3 ? 3 4`
`4 ? 1 1`
`10 ? 1 1`
	`7`
	`0`
	`30`

minuta	1	2	3	4	5	opad
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	3	3	3
3	0	0	0	**7**	**7**	4
4	**0**	0	0	12	12	5
5	0	0	0	18	18	6
6	0	0	0	25	25	7
7	0	0	0	33	33	8
8	9	9	9	42	42	9
9	19	19	19	52	52	10
10	**30**	30	30	63	63	11

E: Ewaluacja

Limit pamięci: **256 MB**

Oktagon zajmuje się utrzymywaniem licznych ważnych obiektów, takich jak centra komunikacyjne, kantyny, biura weteranów, magazyny odzieży czy silosy konwencjonalnych poddźwiękowych pocisków manewrujących ziemia-ziemia. Sprawą wagi państwowej jest zabezpieczanie siatki połączeń między obiektami przed atakiem wroga. Dla każdej dwukierunkowej drogi łączącej różne obiekty określony został Współczynnik Groźby Ataku, zwany potocznie współczynnikiem. Jasio, od niedawna pracownik Oktagonu, nawet zbudzony w środku nocy potrafi uporać się z problemem wyznaczenia podzbioru dróg minimalizującego sumę współczynników i takiego, że wciąż możliwe jest przemieszczenie się między dowolną parą obiektów jedynie po drogach tego podzbioru; droga, która znajduje się w przynajmniej jednym takim zbiorze, nazywana jest kluczową.

Bezpieczeństwo kraju wymaga jednak sięgania myślą dalej – współczynniki mogą przecież ulec zmianie. W ramach corocznych ewaluacji Oktagon postanowił dla każdej drogi obliczyć największą wartość x taką, że gdyby zmienić współczynnik tej drogi na x (a wszystkie pozostałe współczynniki pozostawić bez zmian), to droga ta byłaby kluczowa. Zadanie to przypadło Jasiowi, który nie może pomylić się w tak istotnej sprawie. Pomóż mu!

Wejście

W pierwszym wierszu wejścia podano dwie liczby całkowite n i m ($1 \leqslant n \leqslant 100\,000$, $n-1 \leqslant m \leqslant 1\,000\,000$), oddzielone pojedynczym odstępem, oznaczające odpowiednio liczbę obiektów pod zarządem Oktagonu oraz liczbę dróg pomiędzy nimi. Obiekty ponumerowane są kolejnymi liczbami naturalnymi od 1 do n.

W każdym z kolejnych m wierszy znajdują się trzy liczby całkowite: a, b i c ($1 \leqslant a, b \leqslant n$, $a \neq b$, $0 \leqslant c \leqslant 10^9$), pooddzielane pojedynczymi odstępami, oznaczające, że pomiędzy obiektami o numerach a i b istnieje droga o współczynniku c. Każda nieuporządkowana para $\{a, b\}$ wystąpi co najwyżej raz.

Możesz założyć, że istniejące drogi umożliwiają przejazd pomiędzy dowolnymi dwoma obiektami.

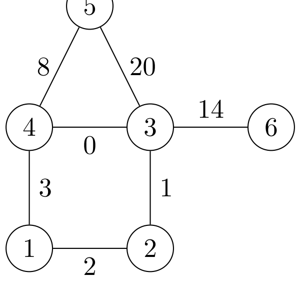
Wyjście

Należy wypisać m wierszy, po jednym dla każdej drogi z wejścia, w kolejności takiej samej, w jakiej drogi występują na wejściu. Dla danej drogi należy wypisać jedną liczbę naturalną – maksymalną wartość taką, że po zmianie na nią współczynnika tej drogi (i pozostawieniu innych współczynników bez zmian) droga ta jest kluczowa. Gdyby wartość ta miała być większa niż 10^9 lub dowolnie duża, należy wypisać 10^9.

Przykład

Wejście	Wyjście
6 7 1 2 2 2 3 1 3 4 0 1 4 3 3 5 20 4 5 8 3 6 14	3 3 3 2 8 20 1000000000

Sieć połączeń z przykładu wygląda następująco:



F: Foremki na ciasta

Limit pamięci: **256 MB**

Małgosia uwielbia piec ciasta. Kupiła ostatnio trzy foremki o okrągłych podstawach i znanych polach powierzchni. Chcąc je natychmiast wypróbować, upiekła ciasta w dwóch pierwszych foremkach i ma zamiar zanieść je do koleżanki; niestety, dwie foremki niesie się dość niewygodnie, więc Małgosia wolałaby włożyć oba ciasta do trzeciej foremki i zanieść tylko ją. Nie umie jednak stwierdzić, czy ciasta się zmieszczą, przy czym nie mogą one być umieszczone w trzeciej foremce „na styk” – trzeba je oddzielić przynajmniej papierem, żeby kremy ciast się nie wymieszały. Pomóż Małgosi – oblicz, czy podane dwa ciasta mieszczą się w trzeciej foremce.

Opis testów

W pierwszym wierszu wejścia podana jest jedna liczba naturalna $1 \leqslant t \leqslant 1\,000$ oznaczająca liczbę zestawów danych, które są opisane w kolejnych wierszach, po jednym zestawie na wiersz. Każdy z zestawów jest zgodny ze specyfikacją podaną w części *Jeden zestaw danych*. Wyniki dla poszczególnych zestawów należy wypisać w osobnych wierszach i powinny być one zgodne ze specyfikacją opisaną w części *Wynik dla jednego zestawu*; należy je wypisać w takiej kolejności, w jakiej zestawy występują na wejściu.

Jeden zestaw danych

Jeden zestaw składa się z trzech liczb naturalnych p, d, t ($1 \leqslant p, d, t \leqslant 8 \cdot 10^9$), oddzielonych pojedynczymi odstępami, oznaczających odpowiednio pola powierzchni pierwszej, drugiej i trzeciej foremki na ciasto.

Wynik dla jednego zestawu

Dla każdego zestawu danych należy wypisać osobny wiersz zawierający jedno słowo `TAK`, jeżeli ciasta z pierwszej i drugiej foremki wejdą razem „nie na styk” do trzeciej foremki, lub `NIE` w przeciwnym przypadku.

Przykład

Wejście	Wyjście
3 3 9 25 4 9 25 5 9 25	TAK NIE NIE

G: Gusta malarskie

Limit pamięci: **256 MB**

Małgosia jest szczęśliwą posiadaczką obszernej kolekcji malarstwa nowoczesnego. Szczególną dumą napawa ją zbiór obrazów, z których każdy zawiera inną liczbę wielobarwnych kwadratów. Niedawno postanowiła w końcu zorganizować prywatną wystawę dla swoich przyjaciół, jednakże ich wysublimowany gust wymaga szczególnej ostrożności. Z doświadczenia Małgosia wie, że jeśli dla pewnego k wśród prezentowanych obrazów znajdują się trzy zawierające dokładnie k, $2k$ i $3k$ kwadratów, cała wystawa zostanie uznana za *przewidywalną*, zatem nudną, czyli kompletnie nieudaną. Jeśli taka liczba nie istnieje, wydarzenie będzie wielkim sukcesem towarzyskim. Jednocześnie wystawa powinna prezentować jak najwięcej obrazów, zatem kolekcjonerka całymi dniami głowi się nad tym, jak wybrać możliwie dużo dzieł tak, by złożona z nich wystawa nie okazała się przewidywalna. Pomóż jej w tym trudnym zadaniu.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna n ($1 \leqslant n \leqslant 50\,000$), oznaczająca liczbę obrazów z kwadratami w kolekcji Małgosi. W drugim i ostatnim wierszu wejścia znajduje się ciąg n parami różnych liczb naturalnych a_i ($1 \leqslant a_i \leqslant 10^9$), pooddzielanych pojedynczymi odstępami, oznaczających liczby kwadratów na poszczególnych obrazach.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczbę całkowitą – największą możliwą liczbę obrazów z kolekcji, z których można stworzyć nieprzewidywalną wystawę.

Przykład

Wejście	Wyjście
3 6 9 3	2

Wszystkie trzy obrazy tworzą przewidywalną wystawę, ale dowolne dwa spełniają warunki Małgosi.

Wejście	Wyjście
6 1 2 3 4 5 6	5

Największy zestaw obrazów, który nie jest przewidywalny, zawiera wszystkie poza drugim obrazem.

H: Hobby

Limit pamięci: **256 MB**

Małgosia jest ostatnio coraz bardziej sfrustrowana, ponieważ Jasio nie daje się nigdzie wyciągnąć. Ciężko nawet z nim porozmawiać, całymi dniami tylko siedzi z nosem w książeczkach z łamigłówkami logicznymi i wpisuje liczby w kratki. Sudoku, KenKen, Kakuro, Kuromasu — Małgosia nie jest już w stanie spamiętać tych wszystkich osobliwych nazw. Jej cierpliwość dawno się już wyczerpała, wpadła więc na śmiały pomysł. By udowodnić Jasiowi, że uzupełnianie plansz liczbami jest szalenie powtarzalnym i nudnym zadaniem, napisze program błyskawicznie rozwiązujący tego rodzaju łamigłówki.

Najnowszą ulubioną grą Jasia jest Suko. Wypełnia się w niej różnymi cyframi od 1 do 9 planszę rozmiaru 3×3. Pola planszy numerowane są liczbami od 1 do 9: w i-tym od góry wierszu znajdują się kolejno od lewej pola $3i-2$, $3i-1$ oraz $3i$. Dodatkowo każde pole posiada przypisany kolor: czerwony, zielony lub niebieski. Wypełniona plansza musi spełniać następujące warunki dotyczące podkwadratów planszy wielkości 2×2 oraz pól każdego koloru:

- W każdym polu umieszczona jest cyfra od 1 do 9, przy czym żadna cyfra się nie powtarza.
- Suma cyfr na polach 1, 2, 4 i 5 wynosi v_1.
- Suma cyfr na polach 2, 3, 5 i 6 wynosi v_2.
- Suma cyfr na polach 4, 5, 7 i 8 wynosi v_3.
- Suma cyfr na polach 5, 6, 8 i 9 wynosi v_4.
- Suma cyfr na czerwonych polach wynosi $v_{\mathtt{A}}$.
- Suma cyfr na zielonych polach wynosi $v_{\mathtt{B}}$.
- Suma cyfr na niebieskich polach wynosi $v_{\mathtt{C}}$.

Pomóż Małgosi udowodnić Jasiowi miałkość jego hobby i napisz program wypełniający podaną planszę zgodnie z powyższymi warunkami.

Wejście

W pierwszym wierszu wejścia znajdują się trzy liczby naturalne $v_{\mathtt{A}}, v_{\mathtt{B}}$ i $v_{\mathtt{C}}$ ($1 \leqslant v_{\mathtt{A}}, v_{\mathtt{B}}, v_{\mathtt{C}} \leqslant 42$), pooddzielane pojedynczymi odstępami. W drugim wierszu znajdują się cztery liczby naturalne v_1, v_2, v_3, v_4 ($10 \leqslant v_1, v_2, v_3, v_4 \leqslant 30$), pooddzielane pojedynczymi odstępami. W trzecim wierszu znajduje się opis kolorów pierwszego od góry wiersza planszy. Składa się on z trzech znaków ze zbioru {`A`, `B`, `C`}, oznaczających kolory czerwony, zielony i niebieski, które opisują kolory kolejnych pól w pierwszym wierszu planszy. W czwartym i piątym wierszu wejścia znajdują się analogiczne opisy drugiego i trzeciego wiersza planszy.

Dla każdego z trzech kolorów istnieje przynajmniej jedno pole tego koloru.

Wyjście

Należy wypisać trzy wiersze opisujące rozwiązanie zadanej łamigłówki – w i-tym wierszu wyjścia powinien znaleźć się opis i-tego wiersza planszy w postaci trzech cyfr umieszczonych w kolejnych polach, od lewej do prawej.

Jeśli istnieje wiele rozwiązań, Twój program może podać dowolne z nich.

Jeśli nie istnieje plansza spełniająca warunki zadania, należy w pierwszym i jedynym wierszu wypisać jedno słowo `NIE`.

Przykład

Wejście	Wyjście
8 19 18 18 18 20 25 BBB BAA CCC	537 462 198

I: Indeks

Limit pamięci: **256 MB**

Na przekór modzie Małgosia wciąż ma papierowy indeks i po każdym semestrze zbiera do niego wpisy od wykładowców. Pokoje wykładowców znajdują się wzdłuż nieskończenie długiego korytarza i są numerowane kolejnymi liczbami naturalnymi zaczynając od 1. Wpis z każdego przedmiotu można odebrać codziennie, ale tylko w określonym pokoju i tylko przez jedną minutę w ciągu dnia. Odebranie wpisu zajmuje zaniedbywalną ilość czasu, ale przejście między sąsiednimi pokojami, w dowolną stronę, zajmuje dokładnie 1 minutę. Dany wykładowca może prowadzić kilka różnych przedmiotów i wtedy może, choć nie musi, ustalić dla części z nich tą samą minutę odbioru wpisów; w takim przypadku odebranie dowolnej liczby wpisów wciąż zajmuje zaniedbywalnie mało czasu.

Małgosia uczęszczała na n przedmiotów i dla każdego z nich wie, w którym pokoju oraz w której minucie dnia można odebrać potrzebny wpis. Każdego dnia Małgosia wstaje na tyle wcześnie, że w minucie 1 może być w dowolnym pokoju. Pomóż jej wyznaczyć minimalną liczbę dni potrzebnych na zebranie wszystkich wpisów.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \leqslant n \leqslant 500\,000$), oznaczająca liczbę wpisów, które chce uzyskać Małgosia. W każdych z kolejnych n wierszy znajduje się opis jednego wpisu. Składa się on z dwóch liczb całkowitych p i t ($1 \leqslant p, t \leqslant 10^9$), oddzielonych pojedynczym odstępem, oznaczających, że wpis z tego przedmiotu można dostać codziennie w pokoju p w minucie t (liczonej od początku dnia).

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczę całkowitą – minimalną liczbę dni potrzebnych Małgosi na zebranie wszystkich wpisów.

Przykład

Wejście	Wyjście
7 2 1 1 4 3 2 1 1 4 2 5 3 1 1	3

Pierwszego dnia Małgosia może odebrać wszystkie wpisy z pokoju nr 1. Kolejnego dnia jest w stanie odebrać wpisy w pokojach 2 i 3, a trzeciego dnia – w pokojach 4 i 5, co wyczerpuje zbiór wpisów do zebrania.

J: Jednakowe szaliki

Limit pamięci: **256 MB**

Małgosia dorabia wieczorami, robiąc szaliki na drutach. Ma szczęście – dziś zjawił się kupiec, który kupi dowolnie wiele szalików; stawia tylko jeden warunek, mianowicie muszą składać się one z tej samej liczby rzędów oczek (bo inaczej źle wyglądają w sklepie). Kupiec oznajmił, że wróci po zamówienie dokładnie za k chwil. Małgosia zna obecne długości wszystkich szalików i zarówno sprucie, jak i dodanie jednego rzędu oczek zajmuje jej jedną chwilę. Pomóż Małgosi – oblicz, ile maksymalnie szalików zdąży wyrównać do powrotu kupca.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite n oraz k ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant k \leqslant 10^9$). Są to odpowiednio liczba szalików oraz liczba chwil, po których wróci kupiec. W drugim i ostatnim wierszu wejścia znajduje się n liczb naturalnych a_i ($1 \leqslant a_i \leqslant 10^9$), pooddzielanych pojedynczymi odstępami. Są to długości kolejnych szalików liczone w rzędach oczek.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczbę całkowitą – maksymalną liczbę szalików tej samej długości, które jest w stanie dostarczyć Małgosia do powrotu kupca.

Przykład

Wejście	Wyjście
5 6 1 2 3 4 4	5

Małgosia zdąży w ciągu sześciu chwil wyrównać wszystkie szaliki do długości 2.

K: Kieszonkowe

Limit pamięci: **256 MB**

Jasio pragnie zostać kolarzem, znalazł nawet idealny rower, ale niestety nie ma za co go kupić. Uprosił mamę o dzienne kieszonkowe, ta postawiła jednak warunki: początkowo kieszonkowe wynosi 0 zł, a później każdego kolejnego dnia mama wypłaci aktualną wartość kieszonkowego i następnie sprawdzi oceny Jasia – jeśli przyniesie on więcej piątek oraz szóstek niż dwójek i jedynek, mama podwyższy przyszłą wysokość kieszonkowego o jedną złotówkę, jeśli mniej – obniży ją o jedną złotówkę, w pozostałym przypadku wysokość kieszonkowego nie zmieni się. Jeśli kieszonkowe spadnie poniżej 0, mama przestanie w ogóle je wypłacać i Jasio nie nazbiera na kolarzówkę.

Po latach Jan z rozrzewnieniem wspomina te czasy. Wiele wciąż pamięta: uzbierał dokładnie tyle, ile kosztował rower i końcowa wartość kieszonkowego wynosiła 0. Nie może sobie tylko przypomnieć, ile kosztował jego rower. Odnalazł nawet swój dzienniczek z ocenami, jest on jednak dość zniszczony i nie dla wszystkich dni był pewien, jakie dostał wtedy oceny. Czy jesteś w stanie pomóc Janowi i powiedzieć ile minimalnie, a ile maksymalnie mógł kosztować jego rower? Niestety jest możliwe, że Jan pomylił się przy odczytywaniu ocen i nie można utworzyć poprawnego ciągu kolejnych wysokości kieszonkowego, który byłby spójny z odtworzonymi przez niego informacjami.

Wejście

W pierwszym i jedynym wierszu wejścia znajduje się jedno niepuste słowo długości co najwyżej 1 000 000 znaków, składające się z symboli `+`, `-`, `0` oraz `_`. Znaki reprezentują zmiany wysokości kieszonkowego w kolejnych dniach: `+` oznacza, że danego dnia mama Jasia podniosła jego kieszonkowe, `-` oznacza, że danego dnia jego kieszonkowe zmalało, `0` oznacza, że kieszonkowe się nie zmieniło, a `_` oznacza, że Jan nie był w stanie odtworzyć co się stało z kieszonkowym danego dnia.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać dwie liczby naturalne oddzielone pojedynczym odstępem, oznaczające odpowiednio minimalną i maksymalną możliwą cenę roweru.

Jeśli podane na wejściu słowo nie odpowiada żadnemu poprawnemu ciągowi kolejnych wysokości kieszonkowego należy wypisać tylko jedno słowo `NIE`.

Przykład

Wejście	Wyjście
`+_+-0_0_+-`	`3 13`

Najmniejsza cena roweru osiągana jest dla sekwencji `+-+-0000+-`, największa zaś dla `+++-0-0-+-`.

Wejście	Wyjście
`__---_`	`NIE`

Niezależnie od ocen w dwóch pierwszych dniach, wartość kieszonkowego po piątym dniu byłaby ujemna.

L: Linie

Limit pamięci: **256 MB**

Jasio nauczył się dziś grać w kółko i krzyżyk i postanowił uogólnić tę grę na większe plansze. Najwyraźniej nie zna zasad gomoku, bo w jego uogólnieniu wygrywa gracz, który zapełni całą linię (pionową, poziomą lub którąś z głównych przekątnych) swoim symbolem. Opracowanie strategii dla tej gry wygląda na trudne, więc Jaś postanowił zająć się prostszym problemem zliczania ile jest remisowych plansz rozmiaru $n \times n$, to znaczy całkowicie wypełnionych kółkami lub krzyżykami, tak że żadna linia (pionowa, pozioma ani żadna z głównych przekątnych) nie jest wypełniona jednym symbolem. Jasio chce najpierw rozwiązać najbardziej podstawową wersję problemu, w której proporcje kółek i krzyżyków są dowolne, oraz *nie utożsamiamy* plansz, które można przekształcić na siebie przez obrót lub symetrię. Nawet w tej wersji ten problem dalej jest trudny. Pomóż mu go rozwiązać!

Wejście

W pierwszym i jedynym wierszu wejścia znajdują się dwie liczby naturalne n i p ($1 \leqslant n \leqslant 300$, $2 \leqslant p \leqslant 10^9 + 9$), oddzielone pojedynczym odstępem, gdzie n jest rozmiarem planszy Jasia, a p jest liczbą pierwszą.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać resztę z dzielenia przez p liczby wszystkich całkowicie wypełnionych plansz $n \times n$, w których nie ma linii złożonej z jednego symbolu.

Przykład

Wejście	**Wyjście**
`3 101`	`32`

Wejście	**Wyjście**
`4 3`	`2`

M: Magiczny labirynt

Limit pamięci: **512 MB**

Klasa Małgosi wybrała się na wycieczkę do labiryntu! Labirynt ma kształt prostokąta o wysokości n i szerokości m metrów oraz składa się z $n \cdot m$ kwadratowych sal, każda o wymiarach 1 na 1 metr. Między każdą parą sąsiadujących bokiem sal jest przejście prowadzące w jednym kierunku. Niestety, ze względu na remonty część przejść jest nieczynna, w szczególności nie wiadomo nawet, czy da się przejść od wejścia do wyjścia labiryntu.

Przed wejściem do labiryntu Małgosia dostała mapę, na której zaznaczono kierunki przejść między poszczególnymi salami. Z mapy wynika też, że wejście do labiryntu prowadzi do sali w lewym górnym rogu mapy, zaś jedyne wyjście z labiryntu znajduje się w sali w prawym dolnym rogu mapy. Na odwrocie mapy podana jest dodatkowo informacja, że w labiryncie nie da się kręcić w kółko – jeśli wyjdzie się z dowolnej sali dowolnym przejściem, to nie da się już do tej sali wrócić.

Małgosia chce przejść labirynt od wejścia do wyjścia i zapisać w kolejności odwiedzenia numery dwóch spośród mijanych sal. Być może któraś z sal spodoba jej się tak bardzo, że zapisze ją dwa razy. Jeśli Małgosi nie uda się dostać do wyjścia, to obrazi się i nie zapisze nic. Mając mapę labiryntu powiedz, na ile sposobów Małgosia może zapisać te dwa numery.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne n i m ($1 \leqslant n{\cdot}m \leqslant 500\,000$) oddzielone pojedynczym odstępem, oznaczające odpowiednio wysokość i szerokość labiryntu. W kolejnych $2n-1$ wierszach znajduje się mapa labiryntu. W wierszu wejścia o numerze $2i$ ($1 \leqslant i \leqslant n$) znajduje się ciąg $m-1$ znaków ze zbioru { `>`, `<`, `*` } opisujących przejścia między kolejnymi salami w i-tym wierszu mapy: jeśli j-ty znak w ciągu to `>`, to istnieje przejście z sali j do $j+1$, `<` oznacza istnienie przejścia z sali $j+1$ do j, zaś `*` zupełny brak przejścia między tymi salami.

Analogicznie, w wierszu o numerze $2i+1$ ($1 \leqslant i \leqslant n-1$) znajduje się ciąg m znaków ze zbioru { `v`, `^`, `*` } opisujących przejścia między salami w wierszach i oraz $i+1$ mapy: jeśli j-ty znak w ciągu to `v`, to istnieje przejście z sali j w i-tym wierszu do sali j w wierszu $i+1$, `^` oznacza istnienie przejścia z sali j w wierszu $i+1$ do sali j w wierszu i, zaś `*` zupełny brak przejścia między salami j w wierszu i oraz w wierszu $i+1$.

Wejście do labiryntu prowadzi do pierwszej sali w pierwszym wierszu, a wyjście znajduje się w ostatniej sali w ostatnim wierszu.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać na ile sposobów Małgosia może zapisać sobie numery dwóch sal. Jeśli nie ma drogi z wejścia do wyjścia, to należy wypisać 0.

Przykład

Wejście	Wyjście

`2 3`
`>>`
`*^v`
`<>`
	`10`

Zosia musi pójść dwa razy w prawo a potem raz w dół.