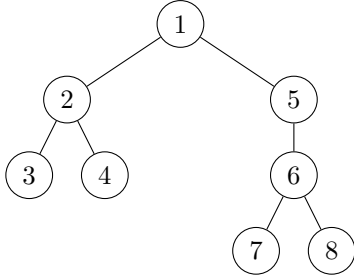


Wypisywanie drzewa poziomami (A)

Limit pamięci: 256 MB

Limit czasu: 2.50 s

Jasio kupił sobie w sklepie drzewo. Wygląda dość typowo, jak każde drzewo ze sklepu z drzewami, czyli jakoś mniej więcej tak:



Otrzymujesz od Jasia opis jego drzewa, składającego się z N wierzchołków. Korzeń drzewa ma numer 1, zaś pozostałe wierzchołki numerowane są kolejnymi liczbami naturalnymi od 2 do N włącznie. Dla każdego wierzchołka i , który nie jest korzeniem drzewa, znasz numer P_i jego bezpośredniego rodzica w drzewie.

Wypisz drzewo poziomami, tzn. w kolejnych wierszach należy wypisać wierzchołki, na coraz większych głębokościach (coraz większych odległościach od korzenia).

Wejście

W pierwszym (jedynym) wierszu wejścia znajduje się jedna liczba naturalna N , określająca liczbę wierzchołków drzewa. W drugim (ostatnim) wierszu wejścia znajduje się ciąg $N - 1$ liczb naturalnych P_2, P_3, \dots, P_N , podzielanych pojedynczymi odstępami.

Wyjście

Twój program powinien wypisać na wyjście opis drzewa z wejścia. W $(d + 1)$ -tym wierszu powinien się znaleźć rosnący ciąg numerów wierzchołków znajdujących się w odległości d od korzenia.

Ograniczenia

$$1 \leq N \leq 1\,000\,000, 1 \leq P_{i+1} \leq i.$$

Przykład

Wejście

```
8
1 2 2 1 5 6 6
```

Wyjście

```
1
2 5
3 4 6
7 8
```

Wyjaśnienie

Ten test przykładowy opisuje sytuację z rysunku powyżej.

Gra w życie (B)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Jaś spał pod kamieniem przez ostatnie dziesięciolecia i dopiero dzisiaj odkrył jeden z najbardziej popularnych automatów komórkowych, nazywanych po polsku *Gra w życie*.

Idąc za Wikipedią, gra toczy się na nieskończonej planszy podzielonej na kwadratowe komórki. Każda komórka ma ośmiu sąsiadów (komórki przylegające do niej bokami lub rogami). Komórki mogą być żywe lub martwe. Stan komórek z momentu T jest używany do obliczenia stanu komórek w momencie $T + 1$. Martwa komórka, która ma dokładnie trzech żywych sąsiadów, staje się żywa w następnej jednostce czasu. Żywa komórka z dwoma lub trzema żywymi sąsiadami pozostaje w następnej jednostce czasu żywa. Przy innej liczbie sąsiadów żywa komórka umiera (a martwa pozostaje martwa).

Okazuje się, że tak proste reguły pozwalają na uzyskiwanie bardzo ciekawych struktur (polecamy po turnieju przejrzeć podlinkowany artykuł z Wikipedii).

Jasio nie ma nieskończonej pamięci w komputerze, żeby symulować grę w życie. Dlatego symuluje ją na skończonej planszy o wymiarach 15×15 . Zastanawia się czy symulacja ewolucji z jego początkowego układu ma sens na tak małej planszy: być może w którymś momencie struktura wyewoluuje poza zakres planszy. A być może nie? Jaś chciałby to wiedzieć. Pomożesz mu?

Wejście

Wejście składa się z piętnastu wierszy po piętnaście znaków. Każdy znak to `.` lub `0` (duża litera o), oznaczające odpowiednio martwą lub żywą komórkę.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna liczba naturalna T , określającą pierwszy moment, w którym struktura przekroczy ramy początkowej planszy.

Jeżeli $T > 100$ lub struktura nigdy nie przekroczy ramy początkowej planszy, to zamiast tego należy wypisać tylko jedno słowo NIE.

Zadanie bonusowe

Po turnieju spróbuj wygenerować test, dla którego odpowiedź T jest skończona, ale większa niż 100.

Przykład

Wejście

```
.....
.00.....
.00.....
.....
.00.....
.0.0.....
..0.....
.....
.....000.....
.....0.....
.....0.....
.....
.....
.000.....
.....
```

Wyjście

```
21
```

Okup (c)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Jasio postanowił zrobić straszną rzecz. . . postanowił zorganizować porwanie dla okupu. Wszyscy w klasie próbowali mu to wybić z głowy, ale bezskutecznie. Postanowił przygotować kartkę z odpowiednim napisem, by wysłać ją osobom, które mają mu zapłacić pieniądze. Ponieważ Jasio naoglądał się filmów akcji, kartka ma powstać z wycinków gazet i literek tam umieszczonych. Niestety, Jasio nie ma zbyt wielu gazet (a więc i dostępnych literek) i może się okazać, że nie każdy napis daje się przygotować z dostępnych znaków. Postanowił więc, że jeżeli zabraknie mu jakichś literek to w ostateczności dopisze je długopisem. Ponieważ z pisaniem również u Jasia krucho, chciałby dopisać jak najmniej literek. Ile to będzie? Pomóż Jasiowi, to może Ciebie nie porwie.

Wejście

W pierwszym wierszu wejścia znajduje się ciąg małych liter alfabetu angielskiego – treść kartki, którą chce przygotować Jasio. W drugim (ostatnim) wierszu wejścia znajduje się ciąg małych liter alfabetu angielskiego – sklejona treść wszystkich gazet posiadanych przez Jasia.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna nieujemna liczba całkowita – minimalna liczba liter, które Jasio będzie musiał dopisać długopisem, żeby uzyskać treść swojej kartki (oprócz tych liter, które przyklei z gazet).

Ograniczenia

Długość każdego z napisów na wejściu nie przekracza 1 000 000 znaków.

Przykład

Wejście

```
dawajkase  
jasiojestgupi
```

Wyjście

```
5
```

Wyjaśnienie

Jasio będzie musiał długopisem napisać literki d, w, k oraz dwie kopie literki a (ponieważ w gazecie jest tylko jedno a, a liście Jasia są trzy).

Liczby rosnące (D)

Limit pamięci: 256 MB

Limit czasu: 0.50 s

Liczbę nazywamy *rosnącą*, jeżeli jej cyfry, czytane od najbardziej znaczącej do najmniej znaczącej, tworzą ciąg (ściśle) rosnący.

Przykładowo: liczba 123 jest rosnąca, podobnie jak 6, 28 oraz 45 789, zaś liczby 72, 555 oraz 45 719 nie są rosnące.

Napisz program, który: wczyta liczbę naturalną N , wyznaczy najmniejszą liczbę rosnącą, która jest większa niż N i wypisze wynik na standardowe wyjście.

Wejście

W pierwszym (jedynym) wierszu wejścia znajduje się liczba naturalna N .

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć najmniejsza możliwa liczba rosnąca M , spełniająca warunek $M > N$.

Jeżeli taka liczba nie istnieje, zamiast tego należy wypisać tylko jedno słowo NIE.

Ograniczenia

$$1 \leq N \leq 10^9.$$

Przykład

Wejście

214

Wyjście

234

Daltonista (E)

Limit pamięci: 256 MB

Limit czasu: 0.50 s

Ludzie tym różnią się od komputerów, że wielu rzeczy potrafią się domyślić. Ciekawe, czy potrafisz rozwiązać zadanie, w którym nie ma treści?

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinno się znaleźć jedno ze słów: czerwony, różowy, pomarańczowy, żółty, zielony, niebieski, fioletowy, brązowy, biały, szary, czarny (bez polskich znaków).

Gwarantowane jest, że dane wejściowe zostały zaprojektowane w taki sposób, że nikt (poza daltonistami) nie będzie miał wątpliwości co do poprawności odpowiedzi.

Uwaga

Inne kolory, w szczególności morski, amarantowy, bordowy, granatowy nie istnieją. Każdy nie-daltonista patrząc na testy, wiedziałby jaka jest poprawna odpowiedź dla każdego z nich (przy tym zbiorze możliwych odpowiedzi jaki jest w zadaniu).

Przykład

Wejście

#ff69b4

Wyjście

różowy

Wejście

#9400d3

Wyjście

fioletowy

Wejście

#080808

Wyjście

czarny

Morfizm (F)

Limit pamięci: 256 MB

Limit czasu: 0.50 s

Jasio, zgłębiając tajniki informatyki, przeczytał ostatnio o *morfizmach* na słowach i ich zastosowaniach do tworzenia zbiorów słów takich jak słowa Thuego-Morse'a lub słowa Fibonacciego.

Jasio postanowił wymyślić swój własny morfizm. Zdefiniował go następująco: $J_0 = S$ dla ustalonego, ulubionego słowa Jasia S składającego się jedynie ze znaków a oraz b. Następnie Jasio zdefiniował regułę podstawiania, która pozwala ze słowa J_i uzyskać słowo J_{i+1} : każde wystąpienie litery a zamienia na słowo bab, zaś każde wystąpienie litery b zamienia na aaa.

Jeśli przykładowo S to słowo baba, to słowo J_1 jest równe aaababaaabab. W podobny sposób ze słowa J_1 można uzyskać słowo J_2 : zaczynałoby się od babbabbabaaabab. . . . Jaś zastanawia się nad tym jak wygląda słowo J_N . Dokładniej, chciałby poznać wycinek $J_N[L \dots R]$ czyli fragment słowa J_N od L -tego do R -tego znaku. Pomożesz mu?

Wejście

W pierwszym wierszu wejścia znajduje się niepusty ciąg znaków a i b – słowo S . W drugim (ostatnim) wierszu wejścia znajdują się trzy liczby naturalne N , L oraz R , oddzielone pojedynczymi odstępami.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinien się znaleźć ciąg znaków o długości $R - L + 1$ – fragment słowa J_N od L -tego do R -tego znaku włącznie.

Znaki słowa numerujemy kolejnymi liczbami naturalnymi, zaczynając od 1.

Ograniczenia

Długość słowa S nie przekracza 100.

$$1 \leq N \leq 30, 1 \leq L \leq R \leq |J_N|, R - L + 1 \leq 100\,000.$$

Przykład

Wejście

baba

2 4 15

Wyjście

babbabaaabab

Skuteczny hacking (G)

Limit pamięci: 256 MB

Limit czasu: 2.50 s

Jasio chce wystartować w Bajtockich Mistrzostwach Szkół Średnich w Programowaniu Zespołowym. Zastanawia się teraz nad nazwą drużyny. A że ma w sobie żyłkę hakiera, postanowił dobrać ją tak, żeby zajmowała jak największą szerokość ekranu. Jasio ma nadzieję, że wtedy, za każdym razem, gdy system konkursowy wyświetli listę rankingową, tabelka z wynikami się “rozjedzie” (szerokość kolumny z nazwą drużyny będzie zbyt duża, powodując poszerzenie tabelki) i wszyscy zobaczą jego geniusz hackingu.

Jasio pieczołowicie sprawdził, która literka alfabetu daje przy wybranej czcionce największą szerokość (w pikselach), wysłał zapytanie do systemu i okazało się, że system konkursowy odrzucił nazwę drużyny, wykrywając, że jej wypisanie na ekranie przekroczyłoby limit szerokości kolumny w tabeli rankingowej.

Jasio zasmucił się, że jego niecny plan nie zadziałał i dlatego postanowił utrzymać nosa autorom systemu konkursowego. Wysyłając wiele zapytań do systemu, ustalił wartość graniczną P . System odrzuca wszystkie nazwy drużyny, których szerokość przy wyświetleniu przekracza szerokość P pikseli.

Jasio postanowił wysłać do systemu taką nazwę drużyny, która zmieści się w limicie pikseli, a jednocześnie będzie się składała z jak największej liczby znaków. Być może uda mu się zapchać bazę danych wysyłając dużo zapytań i każdy zobaczy, że jest elitarnym hakierem? Pomóż mu!

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna P , określająca limit na szerokość nazwy drużyny (w pikselach).

W drugim wierszu wejścia znajduje się jedna liczba naturalna $|\Sigma|$, określająca liczbę różnych znaków w alfabecie, których pozwala użyć system (wartość ta może być większa niż 26 lub nawet 256, ponieważ system, zależnie od konfiguracji, może pozwalać na używanie znaków z UTF-8).

W trzecim (ostatnim) wierszu wejścia znajduje się $|\Sigma|$ liczb naturalnych W_i pooddzielanych pojedynczymi odstępami. Są to szerokości kolejnych znaków alfabetu podane w pikselach według czcionki, której używa system konkursowy.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna nieujemna liczba całkowita – największa możliwa liczba znaków, z których może składać się akceptowana przez system nazwa drużyny.

Ograniczenia

$$1 \leq P \leq 10^9, 1 \leq |\Sigma| \leq 100\,000, 1 \leq W_i \leq 10^9.$$

Przykład

Wejście

```
110
3
19 30 22
```

Wyjście

```
5
```

Wyjaśnienie

Jeżeli założyć, że kolejne znaki alfabetu to a, b i c, to przykładowa nazwa drużyny Jasia mogłaby być na przykład `acacc` (zajmowałaby $19 + 22 + 19 + 22 + 22 = 104$ piksele szerokości).