

Bombberman

WWI 2024 – grupa 2
Dzień 4 – 18 sierpnia 2024

Kod zadania: **bom**
Limit pamięci: **300 MB**



Bajtazar gra w bajtocką wersję gry *Bombberman*. Gracz porusza się postacią na planszy o wymiarach $n \times n$ pól. Każde pole jest albo puste, albo znajduje się na nim przeszkoda w postaci muru z cegieł lub litej skały. Na pola z przeszkodami nie można wchodzić.

Celem gry jest jak najszybsze przejście z pola początkowego do pola końcowego. Można przechodzić z jednego pola na drugie, tylko jeśli mają one wspólny bok. Takie przejście na sąsiednie pole zajmuje jedną sekundę.

Na początku gry gracz umieszcza bombę na dowolnym polu planszy, które nie zawiera litej skały. W szczególności może to być pole z murem z cegieł, pole startowe, końcowe lub puste. Taka bomba wybucha, niszcząc wszystkie mury z cegieł znajdujące się w tym samym wierszu lub w tej samej kolumnie co bomba, o ile pomiędzy nie znajduje się żadne pole z litą skałą. Innymi słowy, jeśli pole z murem z cegieł znajduje się w tej samej kolumnie lub tym samym wierszu co bomba oraz pomiędzy tym polem a bombą nie ma żadnego pola z litą skałą, to zostanie ono zmienione na pole puste, przez które można przejść.

Bomba detonowana jest jeszcze przed umieszczeniem gracza na planszy, zatem nie zostanie on w żaden sposób skrzywdzony niezależnie od umieszczenia ładunku.

Bajtazar chce znaleźć najlepsze rozwiązania dla różnych plansz gry. Napisz program, który pomoże mu w tym.

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita n ($2 \leq n \leq 1000$) oznaczająca rozmiar planszy.

W kolejnych n wierszach znajduje się opis kolejnych wierszy planszy; każdy z nich składa się z n -literowego słowa opisującego kolejne pola w wierszu, składającego się ze znaków . (pole puste), # (pole z murem z cegieł), X (pole z litą skałą), P (pole początkowe) i K (pole końcowe).

W opisie pojawi się dokładnie jedno pole początkowe i dokładnie jedno końcowe. Przyjmujemy, że pola te są puste.

Wyjście

W pierwszym wierszu wyjścia Twój program powinien wypisać liczbę całkowitą T oznaczającą minimalny czas przejścia (w sekundach) z pola początkowego do pola końcowego przy optymalnym wyborze umiejscowienia bomby. W drugim wierszu wyjścia powinny się znaleźć dwie liczby całkowite W oraz K oznaczające odpowiednio numer wiersza oraz numer kolumny, w których należy umieścić bombę. Wiersze jak i kolumny numerujemy liczbami od 1 do n (z góry na dół i od lewej do prawej). W trzecim (i ostatnim) wierszu powinien znaleźć się ciąg liter długości T opisujący sekwencję ruchów pozwalającą na przejście z pola początkowego do pola końcowego. Ciąg ten powinien składać się z liter L, P, G lub D oznaczających kolejno ruchy w lewo, prawo, górę lub dół.

Jeśli nie jest możliwe przejście z pola początkowego do pola końcowego niezależnie od umieszczenia bomby, zamiast tego należy wypisać jeden wiersz zawierający jedno słowo NIE.

Jeżeli istnieje wiele poprawnych rozwiązań, Twój program powinien wypisać dowolne z nich.



Przykład

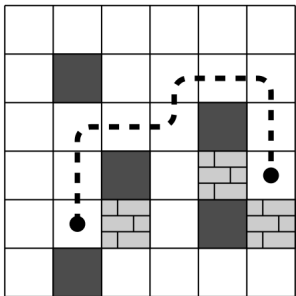
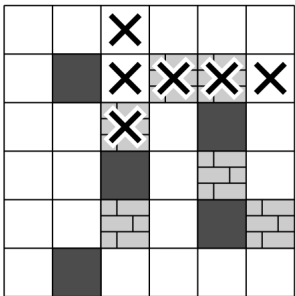
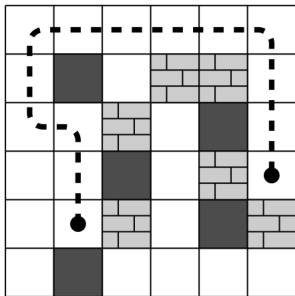
Wejście dla testu bom0:

```
6
.....
.X.##.
..#.X.
..X.#K
.P#.X#
.X....
```

Wyjście dla testu bom0:

```
9
2 3
GGPGPPDD
```

Wyjaśnienie do przykładu: Minimalny czas przejścia bez użycia bomby wynosi 13 sekund. Bomba umieszczona na przecięciu drugiego wiersza i trzeciej kolumny zniszczy trzy mury z cegieł, umożliwiając przejście w ciągu 9 sekund.



Ocenianie

Podzadanie	Ograniczenia	Limit czasu	Liczba punktów
1	plansza nie zawiera murów z cegieł	8 s	10
2	$n \leq 50$	2 s	20
3	$n \leq 200$	3 s	30
4	$n \leq 1000$	8 s	40