

# Zadanie: BOM

## Bombberman



XXIX OI, etap II, dzień drugi. Plik źródłowy bom.\* Dostępna pamięć: 300 MB.

17.02.2022

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

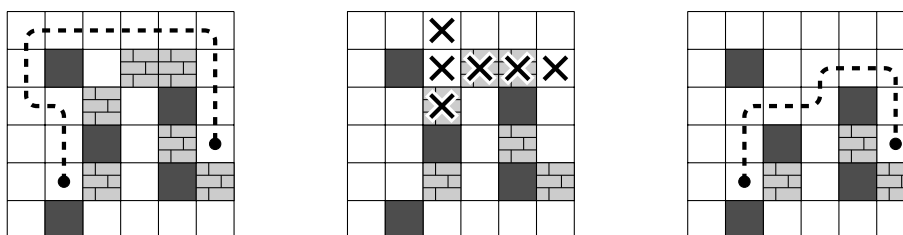
Dla danych wejściowych:

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

poprawnym wynikiem jest:

```
9
2 3
GGPPGPPDD
```

**Wyjaśnienie 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.



### Testy „ocen”:

- 1ocen:**  $n = 7$ , początek i koniec w przeciwległych rogach planszy, brak przeszkód, wynik to 12,
- 2ocen:**  $n = 10$ , początek i koniec oddzielone potrójnym ukośnym murem z litych skał, wynik to NIE,
- 3ocen:**  $n = 21$ , plansza otoczona litymi skałami, dokładnie jedno przejście między początkiem i końcem w kształcie zygzaka, wynik to 198,
- 4ocen:**  $n = 200$ , początek i koniec w przeciwległych rogach planszy, pozostałe pola z murami z cegieł, wynik to 398,
- 5ocen:**  $n = 1000$ , tylko dwa pierwsze wiersze zawierają coś innego niż lite skały, wynik to 1001.

## Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Jeśli Twój program wypisze poprawnie tylko pierwszy wiersz wyjścia, uzyska 80% punktów przewidzianych za test. Zwróć uwagę, że Twój program musi nadal zmieścić się w limicie czasu i pamięci oraz nie spowodować błędu wykonania.

| Podzadanie | Warunki                            | Liczba punktów |
|------------|------------------------------------|----------------|
| 1          | plansza nie zawiera murów z cegieł | 10             |
| 2          | $n \leq 50$                        | 20             |
| 3          | $n \leq 200$                       | 30             |
| 4          | $n \leq 1000$                      | 40             |