



Edytor tekstu

Robert bierze udział w konkursie CEOI 2024. Prawie skończył pisać rozwiązanie najtrudniejszego problemu (tego dnia zawodów), a w dodatku jest przekonany, że otrzyma ono 100 punktów! Jest tylko jeden mały problem: zrobił literówkę! Co więcej, mysz, którą przyniósł na konkurs (używał jej od 2008 r.), rozpadła się na kawałki, kiedy wymyślił rozwiązanie zadania i uderzył pięściami w stół w radosnym uniesieniu. Pozostało jedynie użyć klawiszy strzałek (na klawiaturze).

Program Roberta ma N linii o długościach l_1, l_2, \dots, l_N . Robert zawsze kończy swoje programy pustą linią, więc $l_N = 0$. Cursor może być umieszczony między dwoma kolejnymi znakami, lub na początku/końcu linii. W linii nr i jest $l_i + 1$ możliwych pozycji kursora (tzw. kolumn), ponumerowanych od 1 do $l_i + 1$. Dla przykładu, oto cursor umieszczony w linii 2 w kolumnie 6:

```

1 | i | n | t | . | s | q | u | a | r | e | ( | i | n | t | . | a | ) | . | { |
2 | . | . | r | e | t | u | r | n | . | a | . | * | . | a | ; |
3 | } |
4 | |
(2, 6)

```

Robert chce przenieść cursor z linii s_l i kolumny s_c do linii e_l i kolumny e_c . Twoim zadaniem jest powiedzieć mu, ile razy (co najmniej) musi nacisnąć klawisze strzałek.

Działanie strzałek poziomych jest dość proste. Naciśnięcie strzałki *w lewo* przesuwa cursor do poprzedniej kolumny, chyba, że cursor był na początku linii. W takim wypadku cursor przeniesie się na koniec poprzedniej linii. Podobnie, naciśnięcie strzałki *w prawo* przeniesie cursor do następnej kolumny, lub na początek następnej linii, jeśli cursor znajdował się na końcu linii.

Ruch kursora po dwóch strzałkach *w lewo*:

```

1 | i | f | . | ( | s | e | e | n | [ | n | ] | ) |
2 | . | . | b | r | e | a | k | ; |
3 | s | e | e | n | [ | n | ] | . | = | . | t | r | u | e | ; |
4 | |
(3, 2)

```

←

```

1 | i | f | . | ( | s | e | e | n | [ | n | ] | ) |
2 | . | . | b | r | e | a | k | ; |
3 | s | e | e | n | [ | n | ] | . | = | . | t | r | u | e | ; |
4 | |
(3, 1)

```

←

```

1 | i | f | . | ( | s | e | e | n | [ | n | ] | ) |
2 | . | . | b | r | e | a | k | ; |
3 | s | e | e | n | [ | n | ] | . | = | . | t | r | u | e | ; |
4 | |
(2, 9)

```

Ruch kursora po dwóch strzałkach *w prawo*:

```

1 | i | f | . | ( | s | e | e | n | [ | n | ] | ) |
2 | . | . | b | r | e | a | k | ; |
3 | s | e | e | n | [ | n | ] | . | = | . | t | r | u | e | ; |
4 | |
(1, 12)

```

→

```

1 | i | f | . | ( | s | e | e | n | [ | n | ] | ) |
2 | . | . | b | r | e | a | k | ; |
3 | s | e | e | n | [ | n | ] | . | = | . | t | r | u | e | ; |
4 | |
(1, 13)

```

→

```



1 | i | f | . | ( | s | e | e | n | [ | n | ] | ) |
2 | . | . | b | r | e | a | k | ; |
3 | s | e | e | n | [ | n | ] | . | = | . | t | r | u | e | ; |
4 | |
(2, 1)

```



Naciśnięcie strzałki *w lewo* na samym początku pliku, lub strzałki *w prawo* na samym końcu, nie zmieni pozycji kursora.

Strzałki pionowe działają w sposób odrobinę bardziej złożony. Naciśnięcie strzałki *w górę* przesunie kursor do poprzedniej linii, a naciśnięcie strzałki *w dół* przesunie kursor do następnej, bez zmieniania kolumny. Jeśli jednak umieściłoby to kursor dalej, niż na końcu linii, to zostaje on przeniesiony na jej koniec.

Ruch kursora po dwóch strzałkach *w górę*:

<pre>1 i f . (s e e n [n]) 2 . . b r e a k ; 3 s e e n [n] . = . t r u e ; 4 </pre> <p>(3, 13)</p>		<pre>1 i f . (s e e n [n]) 2 . . b r e a k ; 3 s e e n [n] . = . t r u e ; 4 </pre> <p>(2, 9)</p>		<pre>1 i f . (s e e n [n]) 2 . . b r e a k ; 3 s e e n [n] . = . t r u e ; 4 </pre> <p>(1, 9)</p>
---	---	--	--	--

Ruch kursora po dwóch strzałkach *w dół*:

<pre>1 i f . (s e e n [n]) 2 . . b r e a k ; 3 s e e n [n] . = . t r u e ; 4 </pre> <p>(2, 8)</p>		<pre>1 i f . (s e e n [n]) 2 . . b r e a k ; 3 s e e n [n] . = . t r u e ; 4 </pre> <p>(3, 8)</p>		<pre>1 i f . (s e e n [n]) 2 . . b r e a k ; 3 s e e n [n] . = . t r u e ; 4 </pre> <p>(4, 1)</p>
--	---	--	--	--

Jeśli naciśnięcie strzałki *w górę* lub strzałki *w dół* umieściłoby kursor w linii, która nie istnieje, kursor pozostanie w miejscu.

Wejście

Pierwszy wiersz wejścia zawiera liczbę N – liczbę linii w kodzie Roberta. Drugi wiersz zawiera dwie liczby całkowite dodatnie s_l i s_c oddzielone spacjami, oznaczające początkową pozycję kursora. Podobnie, trzeci wiersz zawiera liczby e_l i e_c – pozycję, na której Robert musi umieścić kursor aby naprawić literówkę. W czwartym wierszu znajdują się oddzielone spacjami liczby naturalne l_1, l_2, \dots, l_N – długości linii w kodzie Roberta.

Wyjście

Twój program powinien wypisać jeden wiersz, zawierający jedną liczbę – minimalną liczbę naciśnieć klawiszy strzałek, która przeniesie kursor z pozycji (s_l, s_c) do (e_l, e_c) .

Przykłady

Przykład 1




Wejście:

```
5
3 1
2 8
7 10 9 9 0
```

Wyjście:

```
3
```

Robert może nacisnąć kolejno strzałki *w górę*, *w lewo* i *w dół*.

<pre>1 i · - = · 1 ; 2 i f · (i · = · 0) 3 · · r e t u r n ; 4 a [i] · = · v ; 5 </pre>		<pre>1 i · - = · 1 ; 2 i f · (i · = · 0) 3 · · r e t u r n ; 4 a [i] · = · v ; 5 </pre>		<pre>1 i · - = · 1 ; 2 i f · (i · = · 0) 3 · · r e t u r n ; 4 a [i] · = · v ; 5 </pre>		<pre>1 i · - = · 1 ; 2 i f · (i · = · 0) 3 · · r e t u r n ; 4 a [i] · = · v ; 5 </pre>
(3, 1)		(2, 1)		(1, 8)		(2, 8)

Mógłby też przesunąć kursor naciskając strzałki *w lewo*, *w górę* i *w dół*. Można łatwo wykazać, że osiągnięcie takiego rezultatu dwoma (lub mniej) strzałkami nie jest możliwe.

Przykład 2

Wejście:

```
5
1 20
3 25
25 10 40 35 0
```

Wyjście:

```
16
```

Najkrótszą sekwencją klawiszy są dwie strzałki *w dół* oraz czternaście strzałek *w prawo*.

Ograniczenia

- $1 \leq N \leq 10^6$
- $0 \leq l_i \leq 10^9$ (dla $1 \leq i \leq N$)

- $l_N = 0$
- $1 \leq s_l, e_l \leq N$
- $1 \leq s_c \leq l_{s_l} + 1$
- $1 \leq e_c \leq l_{e_l} + 1.$

Podzadania

1. (5 punktów) $N \leq 2$
2. (14 punktów) $N \leq 1\,000, l_i \leq 5\,000$
3. (26 punktów) $N \leq 1\,000$
4. (11 punktów) $l_i = l_j$ (dla $1 \leq i, j \leq N$)
5. (44 punkty) *brak dodatkowych ograniczeń*