

Textový editor

Robert soutěží na CEOI 2024. Už je téměř hotov s řešením nejtěžší úlohy a je si docela jistý tím, že za ni dostane plný počet 100 bodů! Zbývá mu vyřešit už jenom jeden malý problém: má ve svém řešení překlep. Navíc se mu ovšem zrovna rozbila jeho oblíbená počítačová myš, kterou používá již od roku 2008. Myš úplně přestala reagovat. Robert se proto bude muset přesunout v textu na místo překlepu pomocí šipek na klávesnici.

Robertův program je tvořen N řádky s délkami l_1, l_2, \dots, l_N . Robert ukončuje každý svůj program prázdným řádkem, takže $l_N = 0$. Kurzor může být umístěn mezi dvěma sousedními znaky na řádku, případně na začátku nebo na konci řádku. Na řádku i tedy máme $l_i + 1$ možných pozic kurzoru (nazývaných sloupce), které si očíslováme postupně zleva doprava od 1 do $l_i + 1$. Takto například vypadá kurzor umístěný na řádku 2 ve sloupci 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 potřebuje přesunout kurzor z řádku s_l a sloupce s_c na řádek e_l do sloupce e_c . Chtěl by zjistit minimální počet stisků kláves, který k tomu musí vykonat.

Klávesy s vodorovnými šípkami fungují celkem jednoduše. Stiskem klávesy *left* přemístíme kurzor do předcházejícího sloupce téhož řádku. Pokud ovšem byl kurzor na začátku řádku, přesune se na konec předchozího řádku. Podobně stiskem klávesy *right* přemístíme kurzor do následujícího sloupce téhož řádku, nebo případně na začátek řádku následujícího, pokud byl kurzor na konci řádku.

Zde je příklad, co se stane, když dvakrát stiskneme klávesu *left*.

```

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)

```

A zde pro dva stisky klávesy *right*.

```

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)

```



Jestliže stiskneme klávesu *left* na úplném začátku souboru, nebo stiskneme klávesu *right* na úplném konci souboru, pozice kurzoru se nezmění.

Chování kláves se svislými šipkami je o něco složitější. Stiskem klávesy *up* se kurzor přesune na předchozí řádek a stiskem klávesy *down* se přesune na následující řádek, v obou případech beze změny sloupce. Pokud by se ale tímto přesunem kurzor dostal za konec nového řádku, posune se navíc na konec tohoto řádku.

Příklad ilustrující dva stisky klávesy *up*.

<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>		<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>		<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>
(3, 13)		(2, 9)		(1, 9)

A klávesy *down*.

<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>		<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>		<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>
(2, 8)		(3, 8)		(4, 1)

Pokud by stisknutí klávesy *up* nebo *down* posunovalo kurzor na neexistující řádek, pozice kurzoru se nezmění.

Vstup

První řádek vstupu obsahuje celé číslo N — počet řádků Robertova programu. Na druhém řádku vstupu jsou dvě celá čísla s_l a s_c oddělená mezerami — počáteční pozice kurzoru. Podobně na třetím řádku jsou dvě celá čísla e_l a e_c — cílová pozice kurzoru. Čtvrtý řádek vstupu obsahuje N celých čísel oddělených mezerami l_1, l_2, \dots, l_N — délky jednotlivých řádků.

Výstup

Váš program musí vypsát jeden řádek obsahující jedno celé číslo — minimální počet stisků kláves potřebný k přemístění kurzoru z pozice (s_l, s_c) na pozici (e_l, e_c) .

Příklady

Příklad 1

Vstup:

```

5
3 1
2 8
7 10 9 9 0


```

Výstup:


3

Robert může přemístit kurzor na cílovou pozici pomocí tří stisků kláves tak, že postupně stiskne klávesy *up*, *left* a *down*:


1		i		.		-		=		.		1		;							
2		i		f		.		(i		.		=		.		0)	
3		.		.		r		e		t		u		r		n		;			
4		a		[i]		.		=		.		v		;			
5																					
																			(3, 1)		



1		i		.		-		=		.		1		;							
2		i		f		.		(i		.		=		.		0)	
3		.		.		r		e		t		u		r		n		;			
4		a		[i]		.		=		.		v		;			
5																					
																			(2, 1)		



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



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

Jinou stejně rychlou možností, jak přemístit kurzor na cílovou pozici, je stisknout postupně klávesy *left*, *up* a *down*. Lze snadno ukázat, že cílové pozice nemůžeme dosáhnout pomocí jednoho nebo dvou stisků kláves.

Příklad 2

Vstup:

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

Výstup:

16

Nejkratší možná posloupnost stisků kláves je tvořena dvěma stisky klávesy *down* a poté čtrnácti stisky klávesy *right*.

Omezení

- $1 \leq N \leq 10^6$
- $0 \leq l_i \leq 10^9$ (pro všechna i taková, že $1 \leq i \leq N$)
- $l_N = 0$
- $1 \leq s_l, e_l \leq N$
- $1 \leq s_c \leq l_{s_l} + 1$
- $1 \leq e_c \leq l_{e_l} + 1$.

Podúlohy

1. (5 bodů) $N \leq 2$

2. (14 bodů) $N \leq 1\,000$, $l_i \leq 5\,000$ (pro všechna i taková, že $1 \leq i \leq N$)
3. (26 bodů) $N \leq 1\,000$
4. (11 bodů) $l_i = l_j$ (pro všechna i, j taková, že $1 \leq i, j \leq N - 1$)
5. (44 bodů) *žádná další omezení*