



עורך טקסט

רוברט מתחרה בתחרות CEOI 2024. הוא כמעט סיים את הפתרון שלו לבעיה הקשה של היום, ולא רק זה, הוא די בטוח שיקבל 100 נקודות! יש רק עניין אחד קטן: הוא עשה טעות הקלדה! וגרוע מכך, נראה שעכבר המחשב האהוב עליו, ששימש אותו מאז שנת 2008, נהרס סוף סוף לגמרי והפסיק להגיב לחלוטין. לכן, הוא יהיה חייב לנווט אל טעות ההקלדה באמצעות מקשי החיצים שעל המקלדת שלו.

לתוכנית של רוברט יש N שורות באורכים l_1, l_2, \dots, l_N . רוברט תמיד מסיים את התוכניות שלו בשורה ריקה, ולכן $l_N = 0$. הסמן (cursor) יכול להיות ממוקם בין שני תווים, או בהתחלה או בסוף של שורה. לפיכך, בשורה i ישנם $l_i + 1$ מיקומים אפשריים לסמן (שנקראים עמודות) הממוספרים מ-1 עד $l_i + 1$. למשל, כך סמן הממוקם בשורה 2 ובעמודה 6 יראה:

```

1 | i | n | t | e | r | n | a | t | i | o | n | a | l |
2 | . | . | r | e | t | u | r | n | a | * | . | a |
3 | } |
4 |
(2, 6)
```

רוברט רוצה להזיז את הסמן שלו משורה s_l ועמודה s_c לשורה e_l ועמודה e_c . הוא רוצה לדעת את המספר המינימלי של לחיצות מקשים הנדרש כדי לעשות זאת.

מקשי החיצים האופקיים הם יחסית פשוטים. לחיצה על שמאל תזיז את הסמן לעמודה הקודמת, אלא אם כן הסמן כבר היה בתחילת שורה, ובמקרה כזה הוא יזוז לסוף השורה הקודמת. באופן דומה, לחיצה על ימין תזיז את הסמן לעמודה הבאה, או לתחילת השורה הבאה אם הסמן היה בסוף שורה.

למשל, לחיצות על שמאל יכולות להיראות כך:

```

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)  ← (3, 1)  ← (2, 9)
```

ולחיצות על ימין יכולות להיראות כך:

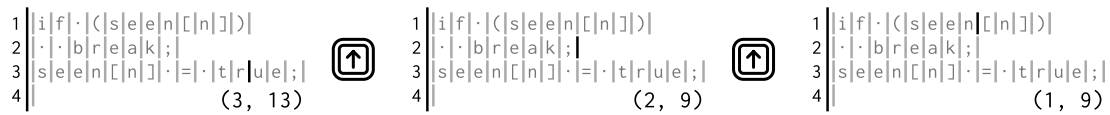
```

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, 13)  → (2, 1)
```

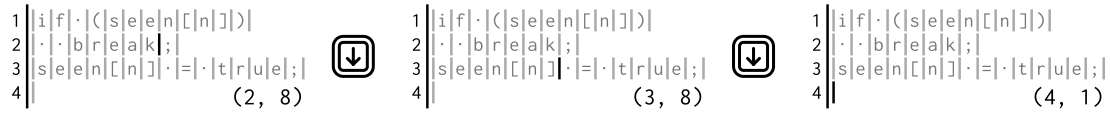
לחיצה על שמאל ממש בתחילת הקובץ או לחיצה על ימין ממש בסוף הקובץ לא תשפיע על כלום.

מקשי החיצים האנכיים הם מעט מורכבים יותר. לחיצה על למעלה תזיז את הסמן לשורה הקודמת ולחיצה על למטה תזיז אותו לשורה הבאה, מבלי לשנות מספר העמודה. אבל, אם זה ישים את הסמן מעבר לסוף של השורה החדשה, הסמן יקפוץ לסוף השורה הזאת במקום.

למשל, לחיצות על למעלה יכולות להיראות כך:



ולחיצות על למטה יכולות להיראות כך:



אם לחיצה על למעלה או על למטה תשים את הסמן על שורה שלא קיימת, הסמן לא יזוז כלל.

קלט

השורה הראשונה של הקלט מכילה את המספר השלם N — מספר השורות בפתרון של רוברט. השורה השנייה מכילה שני מספרים שלמים s_l ו- s_c אשר מופרדים על ידי רווחים — המיקום ההתחלתי של הסמן. באופן דומה, השורה השלישית מכילה שני מספרים שלמים e_l ו- e_c — מיקום המטרה של הסמן. השורה הרביעית מכילה N מספרים שלמים אשר מופרדים על ידי רווח l_1, l_2, \dots, l_N — האורך של כל שורה.

פלט

על התכנית שלכם לפלוט שורה אחת המכילה מספר שלם יחיד — מספר לחיצות המקלדת המינימלי כדי להזיז את הסמן מ- (s_l, s_c) אל (e_l, e_c) .

דוגמאות

דוגמה 1

קלט:

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

פלט:

```
3
```

רוברט יכול להזיז את הסמן אל מיקום היעד באמצעות שלוש לחיצות על ידי לחיצה על למעלה, על שמאל ועל למטה, בסדר זה:

$\begin{array}{l} 1 \left \begin{array}{cccc} i & \cdot & - & = & \cdot & 1 & ; \\ 2 \left \begin{array}{cccc} i & f & \cdot & (& i & \cdot & = & \cdot & 0 & > \\ 3 \left \begin{array}{cccc} \cdot & \cdot & r & e & t & u & r & n & ; \\ 4 \left \begin{array}{cccc} a & [& i &] & \cdot & = & \cdot & v & ; \\ 5 \left \end{array} \right. \end{array} \right. \end{array} \right. \end{array} \quad \begin{array}{c} \boxed{\uparrow} \end{array}$	$\begin{array}{l} 1 \left \begin{array}{cccc} i & \cdot & - & = & \cdot & 1 & ; \\ 2 \left \begin{array}{cccc} i & f & \cdot & (& i & \cdot & = & \cdot & 0 & > \\ 3 \left \begin{array}{cccc} \cdot & \cdot & r & e & t & u & r & n & ; \\ 4 \left \begin{array}{cccc} a & [& i &] & \cdot & = & \cdot & v & ; \\ 5 \left \end{array} \right. \end{array} \right. \end{array} \right. \end{array} \quad \begin{array}{c} \boxed{\leftarrow} \end{array}$	$\begin{array}{l} 1 \left \begin{array}{cccc} i & \cdot & - & = & \cdot & 1 & ; \\ 2 \left \begin{array}{cccc} i & f & \cdot & (& i & \cdot & = & \cdot & 0 & > \\ 3 \left \begin{array}{cccc} \cdot & \cdot & r & e & t & u & r & n & ; \\ 4 \left \begin{array}{cccc} a & [& i &] & \cdot & = & \cdot & v & ; \\ 5 \left \end{array} \right. \end{array} \right. \end{array} \right. \end{array} \quad \begin{array}{c} \boxed{\downarrow} \end{array}$	$\begin{array}{l} 1 \left \begin{array}{cccc} i & \cdot & - & = & \cdot & 1 & ; \\ 2 \left \begin{array}{cccc} i & f & \cdot & (& i & \cdot & = & \cdot & 0 & > \\ 3 \left \begin{array}{cccc} \cdot & \cdot & r & e & t & u & r & n & ; \\ 4 \left \begin{array}{cccc} a & [& i &] & \cdot & = & \cdot & v & ; \\ 5 \left \end{array} \right. \end{array} \right. \end{array} \right. \end{array} \quad \begin{array}{c} \boxed{\downarrow} \end{array}$
$(3, 1)$	$(2, 1)$	$(1, 8)$	$(2, 8)$

לחילופין, הוא יכול להזיז את הסמן שלו למיקום היעד במהירות זזה על ידי לחיצה על שמאל, על למעלה ועל למטה. ניתן להראות בקלות שלא ניתן להגיע למיקום היעד על ידי שתי לחיצות לכל היותר.

דוגמה 2

קלט:

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

פלט:

```
16
```

רצף הלחיצות הקצר ביותר האפשרי בנוי משתי לחיצות על למטה ואז 14 לחיצות על ימין.

מגבלות

- $1 \leq N \leq 10^6$
- $0 \leq l_i \leq 10^9$ (לכל i המקיים $1 \leq i \leq N$)
- $l_N = 0$
- $1 \leq s_l, e_l \leq N$
- $1 \leq s_c \leq l_{s_l} + 1$
- $1 \leq e_c \leq l_{e_l} + 1$

תתי משימות

1. $N \leq 2$ (5 נקודות)
2. $l_i \leq 5\,000, N \leq 1\,000$ (לכל i המקיים $1 \leq i \leq N$) (14 נקודות)
3. $N \leq 1\,000$ (26 נקודות)
4. $l_i = l_j$ (לכל i, j המקיימים $1 \leq i, j \leq N - 1$) (11 נקודות)
5. (44 נקודות) ללא מגבלות נוספות