

# Cursor do editor de texto

João e Leo, dois jovens alunos de engenharia, adoram usar um editor de texto chamado `miv`, que possui dois modos de operação:

1. **Modo de edição:** Modo em que você pode editar texto
2. **Modo cursor:** Modo em que você apenas posiciona o cursor para pode editar texto.

João percebeu algo muito curioso, você pode colocar o cursor em qualquer caractere do seu texto, mas não pode colocar o cursor onde não existe caractere. Por exemplo o código abaixo:

```
1 int_main(void)
2 {
3     //printf("uma_linha_termina_com_fim_de_linha");
4     //printf("_:\n");
5     //return 0;
6 }
```

- O caractere “`_`” representa um caractere de espaço.

A linha 2 possui somente 1 caractere, que é o `{`, logo, nesta linha o cursor pode ficar somente na coluna 1, pois a coluna 2 não possui caractere algum.

O que intrigou a dupla foi que o cursor pode ser posicionado na linha 1 e coluna 3 (em cima da letra `t`), quando desce uma linha o cursor fica posicionado na coluna 1 (em cima do caractere `{`), descendo mais uma vez, para a linha 3, o cursor volta para a coluna 3 (ficando em cima da letra `p`).

Eles ficaram intrigados com o fato do editor se “lembrar” de onde deveria estar, e ficaram pensando se não pode acontecer do editor de texto estar brincando com eles trocando de coluna para textos muito compridos. Para verificar se o editor está correto, eles pediram a *sua* ajuda para escrever um programa que dado um texto, a localização do cursor e vários comandos de subir uma linha e descer uma linha, em qual coluna o cursor deve estar.

Curioso também, que se você estiver na primeira linha e mandar subir uma linha, o editor apenas ignora o comando. O mesmo ocorre quando você está com o cursor na última linha do texto e manda descer uma linha.

## Entrada

A entrada é composta por um único caso de teste, contendo o texto, a posição do cursor e os comandos. A primeira linha de um caso de texto é por composta por um único número inteiro  $L$  ( $1 \leq L \leq 1000$ ), representando a quantidade de linhas no documento, depois serão lidas  $L$  linhas contendo uma string (com espaços) contendo, no máximo, 1000 caracteres e no mínimo 1. Após o texto existe uma única linha contendo dois números inteiros,  $C_l$  e  $C_c$  representando, respectivamente, a linha e a coluna em que o cursor está posicionado, é garantido que o cursor estará posicionado em cima de um caractere válido. E por fim a entrada possui um número incerto de linhas, terminando com EOF, de comandos para o cursor, `j` para descer uma linha e `k` para subir uma linha.

É garantido que nenhuma linha começa com um caractere de espaço.

## Saída

Para cada comando lido, o seu programa deverá imprimir uma única linha contendo: a linha em que o cursor está; a coluna em que o cursor está e; o caractere em que o cursor está.

## Exemplos

### Exemplo de entrada

```
3
Este texto eh um texto de entrada
conte-me nobre aluno o que voce deseja?
um morango? um sorvete? mais nota?
1 10
```

j  
j  
k

Saída para o exemplo de entrada acima

2 10 n  
3 10 o  
2 10 n

Exemplo de entrada

10  
oi  
tudo bem?  
a  
uma letra a acima de mim  
um doi tres  
que exercicio doido  
miv eh vim?  
1245  
f  
abc  
4 5  
k  
k  
j  
j  
j  
j  
j  
j  
j

Saída para o exemplo de entrada acima

3 1 a  
2 5  
3 1 a  
4 5 l  
5 5 o  
6 5 e  
7 5 e  
8 4 5

Exemplo de entrada

3  
a  
abc  
ab  
2 3  
j  
j  
j  
j  
j  
k  
k  
k  
k  
k  
j

Saída para o exemplo de entrada acima

3 2 b  
3 2 b

3 2 b  
3 2 b  
2 3 c  
1 1 a  
1 1 a  
1 1 a  
1 1 a  
2 3 c

*Author: Bruno Ribas*