



# Problema A - Eficiência no Teclado

Consulta a página de instruções para informações detalhadas sobre a qualificação e o formato deste problema.

A Laura, antiga participante das ONI, entrou recentemente para a universidade e planeia participar em competições de programação universitárias em equipa, juntamente com outros antigos concorrentes das ONI. Nestas competições, para além de haver muitos mais problemas do que nas ONI, caso duas equipas façam o mesmo número de problemas a velocidade com que os resolveram é um fator decisivo no desempate, em vez de simplesmente empatarem como nas ONI. Para maximizar a velocidade da equipa, a Laura quer experimentar algumas otimizações que poucas equipas fazem, de forma a garantir que a sua equipa não fica em desvantagem e conseguem programar o mais depressa possível. Para tal, começaram por testar vários teclados para ver com qual é que a equipa se safa melhor, mas encontraram alguns obstáculos e decidiram pedir a tua ajuda.



## Parte I

Para testar cada teclado, decidiram escrever o nome de  $N$  variáveis, cada uma de comprimento  $a_i$ , neste caso todas strings de letras minúsculas, e embora tenham encontrado alguns teclados muito rápidos, encontraram um problema: alguns dos teclados capitalizam, consistentemente, algumas das letras quando não era suposto. A Laura quer descobrir quais das teclas estão avariadas e precisam de ser arrançadas. O teu objetivo, para ajudares a Laura e a sua equipa, é dada uma lista de  $N$  variáveis, determinar que letras aparecem erradamente capitalizadas.

## Exemplo

Por exemplo, a equipa da Laura tentou escrever as variáveis `contador`, `iterador`, `max`, `min`, `i`, mas no ecrã apareceu `cOnTadOr`, `ITeradOr`, `MaX`, `MIn` e `I`, pelo que as letras  $\{i, m, o, t, x\}$  apareceram capitalizadas quando não deviam, enquanto que as letras  $\{a, c, d, e, n, r\}$  apareceram no ecrã como era suposto. Repara que no input deste problema nenhuma letra irá aparecer simultaneamente em maiúscula e minúscula.

## Restrições

São garantidos os seguintes limites em todos os casos de teste desta Parte que irão ser colocados ao programa:

$1 \leq N \leq 10^5$                       Número de variáveis  
 $1 \leq a_1 + a_2 + \dots + a_n \leq 10^5$     A soma dos comprimentos das variáveis

É garantido que no input da Parte I todas as letras do alfabeto aparecem pelo menos uma vez, e que nenhuma letra aparece tanto minúscula como maiúscula.

Os casos de teste desta Parte do problema estão organizados num único grupo:

Grupo	Número de Pontos	Restrições adicionais
1	30	Sem restrições adicionais

## Parte II

Após arranjam aquele que acreditam ser o seu melhor teclado, encontraram outro problema: não sabem que formato de variáveis devem usar. Não só não concordam entre si, como nem sempre são consistentes nos seus próprios códigos. Mas, neste tipo de competições, é comum ter um colega de equipa a ajudar com a escrita e especialmente com o debug dos códigos, pelo que querem garantir que são coerentes no formato de variável que usam nas sessões de treino para se habituarem e depois ser muito ligeiramente mais fácil lerem os códigos uns dos outros no dia da prova.

Os tipos de variáveis usados por membros da equipa em códigos antigos podem resumir-se aos seguintes 4 tipos:

1. *PascalCase*: variáveis compostas por várias palavras onde cada palavra tem a sua inicial capitalizada.
2. *dromedaryCase*: variáveis compostas por várias palavras onde cada palavra, exceto a primeira, tem a sua inicial capitalizada.
3. *snake\_case*: variáveis compostas por várias palavras onde estas são divididas por underscores ‘\_’.
4. *kebab-case*: variáveis compostas por várias palavras onde estas são divididas por hífenos ‘-’.

Decidiram compilar a lista das  $N$  variáveis compostas por várias palavras, de comprimento total  $a_i$ , que utilizaram no passado, e querem a tua ajuda para determinar rapidamente qual o formato de cada para depois poderem decidir que formato usar.

Desta forma, dada uma lista de  $N$  variáveis, determina o formato de cada.

## Exemplo

Se a lista contiver as variáveis `mid-point`, `ListaDeAdjacencia`, `max_flow`, `edgeCase`, `NodePai`, `olimpiadas-de-informatica` deves imprimir, respetivamente, *kebab-case*, *PascalCase*, *snake\_case*, *dromedaryCase*, *PascalCase* e *kebab-case*.

Nota que nesta Parte é garantido que todas as variáveis serão de um dos 4 tipos {*PascalCase*, *dromedaryCase*, *snake\_case*, *kebab-case*}, ou seja, não haverá variáveis de apenas uma palavra

para evitar ambiguidades, e não haverá variáveis que “misturem” tipos como `pastel-de_nata` nem `Departamento-de-Ciencias-de-Computadores`.

## Restrições

São garantidos os seguintes limites em todos os casos de teste desta Parte que irão ser colocados ao programa:

$$\begin{array}{ll} 1 \leq N \leq 10^5 & \text{Número de variáveis} \\ 1 \leq a_1 + a_2 + \dots + a_n \leq 10^5 & \text{A soma dos comprimentos das variáveis} \end{array}$$

Os casos de teste desta Parte do problema estão organizados num único grupo:

Grupo	Número de Pontos	Restrições adicionais
2	30	Sem restrições adicionais

## Parte III

Após analisarem com atenção a tua lista, decidiram que iam usar o formato *dromedaryCase* para todas as suas variáveis. No entanto, a Laura já tinha começado a escrever um programa novo e este tinha já algumas variáveis noutros formatos. A Laura deu-te então uma lista das  $N$  variáveis e pediu-te para as converteres para o formato *dromedaryCase*.

Nesta parte variáveis de uma única palavra contam como *dromedaryCase* se forem compostas por apenas letras minúsculas e como *PascalCase* se a primeira letra for maiúscula.

## Exemplo

Se a lista contiver as variáveis `mid-point`, `ListaDeAdjacencia`, `max_flow`, `Grelha`, `edgeCase`, `NodePai`, `olimpiadas-de-informatica` debes imprimir, respetivamente, `midPoint`, `listaDeAdjacencia`, `maxFlow`, `grelha`, `edgeCase`, `nodePai` e `olimpiadasDeInformatica`

Nota que nesta Parte é garantido que todas as variáveis serão de um dos 4 tipos {*PascalCase*, *dromedaryCase*, *snake\_case*, *kebab-case*}, ou seja, não haverá variáveis de apenas uma palavra para evitar ambiguidades, e não haverá variáveis que “misturem” tipos como `pastel-de_nata` nem `Departamento-de-Ciencias-de-Computadores`.

## Restrições

São garantidos os seguintes limites em todos os casos de teste desta Parte que irão ser colocados ao programa:

$$\begin{array}{ll} 1 \leq N \leq 10^5 & \text{Número de variáveis} \\ 1 \leq a_1 + a_2 + \dots + a_n \leq 10^5 & \text{A soma dos comprimentos das variáveis} \end{array}$$

Os casos de teste desta Parte do problema estão organizados num único grupo:

Grupo	Número de Pontos	Restrições adicionais
3	40	Sem restrições adicionais

## Sumário de subtarefas

Os casos de teste do problema estão organizados em três grupos com restrições adicionais diferentes:

Grupo	Número de Pontos	Parte	Restrições adicionais
1	30	Parte I	Sem restrições adicionais
2	30	Parte II	Sem restrições adicionais
3	40	Parte III	Sem restrições adicionais

## Formato de Input

A primeira linha contém um inteiro  $P$ , que representa a Parte do caso de teste. Se este for 1, então o caso de teste refere-se à Parte I, se for 2, então refere-se à Parte II e se for 3, então refere-se à Parte III.

Segue-se uma linha com um inteiro  $N$ , o número de variáveis na lista.

Por fim seguem-se  $N$  linhas, cada uma com o nome de uma variável.

## Formato de Output

Na Parte I, o output deve conter apenas uma linha, composta por todas as letras que apareceram incorretamente maiúsculas no input separadas por espaços, por ordem alfabética e em **minúsculas**.

Na Parte II, o output deve conter  $N$  linhas, onde a  $i$ -ésima linha indica o formato (*PascalCase*, *dromedaryCase*, *kebab-case* ou *snake\_case*) da  $i$ -ésima palavra.

Na Parte III, o output deve conter  $N$  linhas, onde a  $i$ -ésima linha contém a  $i$ -ésima variável convertida ao formato *dromedaryCase*.

**Nota:** não deve haver nenhum espaço no final de cada linha (ou seja, após cada letra/string deve aparecer apenas uma mudança de linha). Se este formato não for respeitado o resultado de uma submissão será **Presentation Error** (consulta as instruções para mais informações).

## Input do Exemplo 1

```
1
12
ArrOz
blOc0
ItErAdOr
fAcE
grElhA
jOkEr
mAx
nOdE
pAI
qUIcksOrt
vErtIcE
why
```

## Output do Exemplo 1

```
a e i o u
```

## Input do Exemplo 2

```
2
6
mid-point
ListaDeAdjacencia
max_flow
edgeCase
NodePai
olimpiadas-de-informatica
```

## Output do Exemplo 2

```
kebab-case  
PascalCase  
snake_case  
dromedaryCase  
PascalCase  
kebab-case
```

## Input do Exemplo 3

```
3  
7  
mid-point  
ListaDeAdjacencia  
max_flow  
Grelha  
edgeCase  
NodePai  
olimpiadas-de-informatica
```

## Output do Exemplo 3

```
midPoint  
listaDeAdjacencia  
maxFlow  
grelha  
edgeCase  
nodePai  
olimpiadasDeInformatica
```



### Alto Patrocínio

Com o Alto Patrocínio  
de Sua Excelência



O Presidente da República



### Patrocinadores



FUNDAÇÃO  
CALOUSTE  
GULBENKIAN