

Teste Técnico

Estágio Desenvolvimento de Software

Melhor Plano - 2019

Instruções

Resolva os problemas abaixo utilizando a linguagem que você tem mais domínio. Tente criar soluções concisas mas de boa legibilidade.

⇒ Use nome de variáveis, funções e classes sem abreviação que ajudam o leitor do código a entender o que o código quer dizer.

⇒ Evite funções com mais de 10 linhas de código.

Cada problema tem entrada de dados que estão disponíveis para download no link do drive abaixo. Sua solução deve executar e produzir a saída com mesmo nome do arquivo de entrada, alterando a extensão ".in" para ".out".

Exemplo: "teste1.in", ao aplicar a solução deve produzir a solução "teste1.out"

Entrega

Crie um arquivo para cada problema.

Ao finalizar o teste, envie o link da implementação para product@melhorplano.net.

A entrega deve ser feita através do **GitHub**.

Assegure que as saídas dos testes também estão salvas no GitHub (veja instruções acima).

Você tem 10 dias para enviar a solução a partir do email de envio do teste.

Se terminar antes, melhor ainda! Envie pelo menos o primeiro problema, se conseguir fazer o problema extra será ainda melhor!!!

Após a entrega, vamos avaliar o código. Os candidatos que performarem bem serão convidados para uma entrevista técnica. Durante a entrevista técnica, faremos perguntas sobre o teste, tirando dúvidas e aprofundando sobre algumas variações.

Boa sorte!

Problema

Ao fazer a compressão de símbolos, é muito útil ter vários símbolos que seguem um ao outro. Assim eles podem ter a codificação simplificada em array que ocupa menos memória. Ex:

```
encode("aaaabbbccc") === [ [4,'a'], [3,'b'], [3,'c'] ]
```

Existe uma transformação que codifica uma string de tal maneira que os símbolos da string acabam ficando juntos, facilitando a compressão. A transformação de uma string funciona criando uma matriz $n \times n$, onde n é o tamanho da string; cada linha matrix é composta pela rotação de um elemento da linha anterior:

Input: "abracadabra"

```
abracadabra
aabracadabr
raabracadab
braabracada
abraabracad
dabraabraca
adabraabrac
cadabraabra
acadabraabr
racadabraab
bracadabraa
```

Em seguida, a matrix é ordenada por sua linha. A saída do algoritmo de transformação corresponde da string da última coluna e o índice da string de entrada:

```
aabracadabr
abraabracadd
abracadabra  <= 2
acadabraabr
adabraabrac
braabracadaa
bracadabraa
cadabraabra
dabraabraca
raabracadab
racadabraab
Output: ["rdarcaaaabb", 2]
```

Para decodificar: Como vimos a string transformada representa a última coluna da matrix de codificação. A primeira coluna pode ser obtida ordenando os caracteres da última coluna. Como cada linha da coluna é composta pela rotação dos caracteres, as duas primeiras letras da matrix podem ser obtidas juntando a última coluna com a primeira e ordenando novamente a matrix pelas linhas. Adicionando novamente a última coluna e ordenando, temos os 3 primeiros, e assim por diante. Ao final, basta selecionar a palavra formada no índice da transformação.

Decodificação			
Input			
["rdarcaaaaabb", 2]			
Add 1	Sort 1	Add 2	Sort 2
r d a r c a a a a b b	a a a a a b b c d r r	ra da aa ra ca ab ab ac ad br br	aa ab ab ac ad br br ca da ra ra
Add 3	Sort 3	Add 4	Sort 4
raa dab aab rac cad abr abr aca ada bra bra	aab abr abr aca ada bra bra cad dab raa rac	raab dabr aabr raca cada abra abra acad adab braa braa brac cada dabr raab raca	aabr abra abra acad adab braa brac cada dabr raab raca
...	...	Add 11	Sort 11

...	...	raabracadab dabraabraca aabracadabr racadabraab cadabraabra abraabracad abracadabra acadabraabr adabraabrac braabracada bracadabraa	aabracadabr abraabracad abracadabra acadabraabr adabraabrac braabracada bracadabraa cadabraabra dabraabraca raabracadab racadabraab
Output			
abracadabra			

Seu objetivo é implementar um programa que lê os arquivos de entrada (arquivos ".in") e produz arquivos de saída (".out"). Veja um exemplo em "encode-example-1.in" e "encode-example-1.out"

Às entradas que começam com "encode" (exemplo "encode1.in") deve ser aplicado a função de transformação de string. O resultado deve ser salvo como por exemplo "encode1.out".

Às entradas que começam com "decode" (exemplo "decode2.in") deve ser aplicado a função de decodificação de string. O resultado deve ser salvo como por exemplo "decode1.out".

Problema Extra

Contagem de moléculas em uma string. Seu código deve ler a string de uma molécula e contar quantos elementos tem na fórmula. Abaixo estão vários exemplos de fórmulas químicas e a contagem na linha logo abaixo.

Considere usar regular expressions (<https://regexone.com/>). Regexs são muito úteis para processar strings e utilizamos bastante no Melhor Plano. Seu uso não é obrigatório.

H₂O

H: 2, O: 1

C₆H₁₂O₆

C: 6, H: 12, O: 6

Mo(CO)₆

Mo: 1, C: 6, O: 6

As₂{Be₄C₅[BCo₃(CO)₂]₃}₂4Cu₅

As: 2, Be: 16, C: 44, B: 8, Co: 24, O: 48, Cu: 5

Seu objetivo é implementar um programa que lê os arquivos de entrada (arquivos ".in") e produz arquivos de saída (".out"). Veja um exemplo em "example.in" e "example.out"