

Para efeitos da nota atribuída à resolução de exercícios ao longo do semestre - **Submeter até 23:59 de 1 de Novembro** (o problema continuará depois disponível para submissão, mas sem contar para a nota)
[para perceber o contexto do problema deve [ler o guião da aula #03](#)]

[DAA 012] Sequenciação de ADN

Este é um **problema de output-only**.

Ao contrário dos outros problemas em que deves fazer leitura de dados e escrita do output, neste problema deves apenas submeter um ficheiro de texto.

(este problema é essencialmente o mesmo que foi usado na Olimpíadas Nacionais de Informática na prova de seleção para as IOI'2020)

A Espaço Portugal, a maior empresa de exploração espacial Portuguesa, acaba de voltar de uma missão secreta a Marte e trazem um passageiro muito especial: um exemplar de vida extraterrestre!

Este ser vivo é muito diferente de nós a começar pelo mais essencial: o ADN! Em vez dos 4 nucleótidos clássicos, o ser extraterrestre contém 26, um por cada letra do alfabeto inglês. Como tal, todas as técnicas conhecidas para sequenciação de ADN não funcionam neste ser e por isso vai ser preciso um método diferente.

O problema de sequenciação de ADN é o seguinte. São dadas N palavras de ADN, ou seja, conjuntos de letras maiúsculas. O objetivo é encontrar uma palavra mestre que contenha cada uma das N palavras de ADN como subsequências contíguas (ou seja, como *substrings*). O objetivo é encontrar uma palavra mestre curta, por isso é dado ainda um inteiro Q que é o comprimento máximo que a palavra mestre pode ter.

Por exemplo, vamos supor que $N = 3$ e temos as seguintes palavras: AAB, BBCD, EA. Uma possível palavra mestre seria EAABBCD, que tem comprimento 7, por isso se Q for maior ou igual a 7, esta palavra é aceitável.



O Problema

Serão fornecidos 6 conjuntos de palavras de ADN, o objetivo é construir uma palavra mestre de comprimento no máximo Q .

Input

Cada caso de teste começa com um inteiro N , representando o número de palavras, e um inteiro Q , representando o comprimento máximo da palavra mestre.

Seguem-se N linhas, cada uma com uma palavra não vazia.

Casos de teste

Há 6 casos de teste, cada um com uma pontuação fixa (sem pontuações parciais para cada um). **Devem olhar para cada caso de teste individualmente, os casos não são aleatórios e devem observar e aproveitar a estrutura de cada um para o resolver.**

Os casos de teste são os seguintes:

- **Caso 1:** A valer 10 pontos, [inp1.txt](#);
- **Caso 2:** A valer 15 pontos, [inp2.txt](#);
- **Caso 3:** A valer 15 pontos, [inp3.txt](#);
- **Caso 4:** A valer 20 pontos, [inp4.txt](#);
- **Caso 5:** A valer 20 pontos, [inp5.txt](#);
- **Caso 6:** A valer 20 pontos, [inp6.txt](#);

Output/Submissão

Deve ser submetido um único ficheiro de txt (o formato é importante, o vosso ficheiro deve terminar em .txt) com as strings pretendidas para cada caso de teste, sendo que o formato é o seguinte:

M_1 (tamanho da palavra mestre para o caso de teste 1)

$C_1C_2...CM_1$ (string de tamanho M_1 que representa a palavra mestre para o caso de teste 1)

...

M_6 (tamanho da palavra mestre para o caso de teste 6)

$C_1C_2...CM_6$ (string de tamanho M_6 que representa a palavra mestre para o caso de teste 6)

Observações importantes:

- Cada C_i deve ser uma letra maiúscula.
- **Não** devem incluir espaços entre as letras.
- Caso não tenham solução para um dos casos de teste, basta colocarem $M_i = 0$, ou seja, imprimir 0 seguido de uma linha vazia para representar a resposta a este caso de teste.
- Se algum dos outputs for inválido, seja porque M_i é menor que 0 ou maior que **Q**, ou porque o tamanho da palavra for diferente de M_i , o resultado da submissão será **Wrong Answer** e terão 0 pontos.

Um exemplo válido de um ficheiro solução seria o seguinte (a sua pontuação é de 0 pontos):

```
4
ABCD
2
AB
0

1
A
0

0
```

Dicas para gerar o output

Nota: estas dicas apenas se aplicam a quem estiver a utilizar uma shell de Linux.

Visto que é necessário criar um ficheiro de txt para a vossa submissão, pode ser útil usar o terminal para gerar os ficheiros de texto (isto não é obrigatório, podem criar o ficheiro como quiserem).

Se para correrem o vosso programa usarem o comando `X` (por exemplo, para C++ isto seria algo como `./a.out`), depois de compilar com `g++ codigo.cpp`, podem usar o seguinte comando para ler o input de um ficheiro `inp.txt` e escrever para um ficheiro `out.txt`: `X < inp.txt > out.txt`.

Para facilitar a organização do ficheiro de submissão, é aconselhado gerar um txt por cada caso de teste, por exemplo, para o caso de teste 2 ter um `out2.txt`, que teria o formato:

M_2 (tamanho da palavra mestre para o caso de teste 2)

$C1C2...CM_2$ (string de tamanho M_2)

Para posteriormente gerar um ficheiro `out.txt` para submeter, podem usar o seguinte comando (assumindo que têm um ficheiro `outI.txt` para cada um dos 6 casos de teste I): `cat out1.txt > out.txt;cat out2.txt >> out.txt;cat out3.txt >> out.txt;cat out4.txt >> out.txt;cat out5.txt >> out.txt;cat out6.txt >> out.txt`