

E-fólio R

Alínea A)

O e-fólio R é constituído por 4 alíneas, valendo 1 valor cada, devendo as mesmas serem realizadas sequencialmente, e podendo ser reutilizado código entre alíneas. A cotação total do e-fólio é de 4 valores. Os critérios de correção encontram-se no espaço da UC, sendo que 50% é destinado à funcionalidade, resultante da percentagem de casos de teste corretos. A realização do e-fólio na plataforma *HackerRank* não dispensa a entrega do relatório no espaço da UC.

O relatório deve indicar as alíneas realizadas e resultados obtidos, e descrever o código realizado e opções tomadas, o qual não deve ultrapassar as 4 páginas. Se realizou parcialmente uma das alíneas, descreva o que fez e como planeava completar a alínea. Deve colocar o código das alíneas realizadas no anexo, mesmo as que foram realizadas parcialmente, e não colocar código no corpo do relatório.

Este e-fólio inspira-se no jogo *mastermind*. Este jogo consiste em tentar adivinhar uma chave que consiste numa sequência K de peças de W cores. Para tal pode-se fazer apostas numa dada sequência, sendo dada informação de quantas peças estão corretas na posição correta (preto), e quantas peças estão certas em termos de cor, mas na posição errada (branco). O objetivo é adivinhar a chave com o menor número de apostas.

Nesta alínea, dados os valores K e W, deve gerar uma string aleatória com K letras e W possibilidades. Cada cor deve ser uma letra maiúscula. Por exemplo, se K=6, W=4, a string deve ter 6 letras de 'A' a 'D'. Deve utilizar a função aleatória *randaux*, fornecida, e no caso de gerar 4 cores, associar as 4 cores às letras 'A' a 'D', sendo o 'A' associado ao 0 da sequência aleatória, o 'B' associado ao 1, e assim sucessivamente, sendo o 'D' associado ao 3. Em baixo encontram-se algumas execuções de exemplo.

Execuções de exemplo:

K: 6
W: 4
BDCABA

K: 8
W: 4
BDCABACC

K: 6
W: 12
FLKEFE

Pode-se ver na primeira execução de exemplo que 6 letras entre 'A' e 'D' são geradas. Os números correspondentes a essas letras são: 1, 3, 2, 0, 1, 0. Na segunda execução, são geradas 8 letras, cujas primeiras 6 são iguais uma vez que o gerador aleatório é o mesmo e o W é igual (4 em ambas as execuções), pelo que os números iniciais saídos são os mesmos. Na terceira execução, são geradas 6 letras mas com 12 possibilidades.

Alínea B)

Partindo da alínea A, deve-se agora na alínea B solicitar também uma adivinha, de modo a dar o número de letras certas na posição certa (pretas), e letras certas na posição errada (se não fizer esta parte, retorne zero). Mostre para efeitos de testes, a chave, tal como nos exemplos de execução.

Exemplos de execução:

K: 6
W: 4
Aposta: **ABCDAB**
(chave: BDCABA) p1 b5

K: 6
W: 4
Aposta: **ABCAAA**
(chave: BDCABA) p3 b1

K: 6
W: 4
Aposta: **ADDDDA**
(chave: BDCABA) p2 b1

Pode-se ver na primeira execução, que há uma letra certa na posição certa (p1), correspondendo à letra C na terceira posição, e as restantes 5 letras estão certas mas na posição errada (b5). Na segunda execução, há três letras certas na posição certa, as letras CAA, nas posições 3, 4 e 6, e uma outra letra certa na posição errada, a letra B. Note-se que as letras A da aposta, não contam, uma vez que ambas as letras A da chave estavam certas na posição certa, e já foram contabilizadas. Na última execução, existem duas letras certas na posição certa, as letras D e A, nas posições 2 e 6, e uma letra certa na posição errada, a letra A.

Alínea C)

Construa esta alínea sobre a alínea B, e aceite agora várias apostas, até que seja introduzida uma aposta com menos caracteres que os necessários, ou uma aposta igual à chave. No entanto dê a indicação ao utilizador, para além das pretas e brancas, se a aposta dada é compatível com as apostas anteriores, e com qual a aposta é incompatível. Considere uma aposta K incompatível com uma aposta W, se a aposta K, se fosse realmente a chave, tivesse relativamente à aposta W um valor distinto de pretas e brancas. No caso de todas as apostas anteriores serem compatíveis, retornar -1.

Execuções de exemplo:

```
K: 6
W: 4
Aposta 0: ABCDAB
(chave: BDCABA) p1 b5 Incompativel com: -1
Aposta 1: ABCAAA
(chave: BDCABA) p3 b1 Incompativel com: 0
Aposta 2: ADDDDA
(chave: BDCABA) p2 b1 Incompativel com: 0
Aposta 3:
```

```
K: 6
W: 4
Aposta 0: ABCDAB
(chave: BDCABA) p1 b5 Incompativel com: -1
Aposta 1: DACABB
(chave: BDCABA) p3 b3 Incompativel com: 0
Aposta 2: BDCABA
(chave: BDCABA) p6 b0 Incompativel com: -1
```

```
K: 6
W: 4
Aposta 0: CCCABB
(chave: BDCABA) p3 b1 Incompativel com: -1
Aposta 1: ACAACB
(chave: BDCABA) p1 b3 Incompativel com: -1
Aposta 2: CACDDDB
(chave: BDCABA) p1 b3 Incompativel com: -1
Aposta 3: BDCABA
(chave: BDCABA) p6 b0 Incompativel com: -1
```

Pode-se ver na primeira execução, que a segunda aposta é incompatível com a primeira, dado que a aposta 0 e a aposta 1 têm 3 letras certas na posição certa, enquanto que a chave tem apenas uma letra certa na posição certa. Na segunda execução pode-se ver que a aposta 2 é compatível com a aposta 0 e 1, uma vez que têm o mesmo número de brancas e pretas, pelo que a aposta 2 poderia ser a chave. A terceira execução pode-se ver uma sequência de apostas todas compatíveis com as anteriores, terminando na aposta 3 com a própria chave.

Alínea D)

Pretende-se nesta última alínea que implemente o algoritmo proposto que gera apostas aleatórias, mas que sejam compatíveis com as apostas anteriores.

Algoritmo:

1. Se não há ainda apostas, gerar uma aposta como se fosse gerada uma chave (da alínea A);
2. Nas restantes apostas X, ficar com a última aposta Y como referência, e:
 1. Escolher o número de pretas da aposta Y, e copiar para a aposta X:
 1. Gerar um valor de 0 a K-1, e se o valor não foi escolhido, escolher.
 2. Repetir o passo anterior até o número de pretas da aposta Y ter sido copiado.
 2. Escolher o número de brancas da aposta Y, e copiar para outra posição na aposta X:
 1. Gerar um valor entre 0 a K-1 de onde vem;
 2. Gerar um valor entre 0 e K-1 para onde vai;
 3. Se os valores forem diferentes, o local para onde vai ainda não estiver preenchido, o local de onde vem ainda não tiver sido utilizado, e na aposta Y o valor de onde vem e para onde vai for diferente, então efetuar a mudança e repetir o processo até o número de brancas da aposta Y ter sido satisfeito. Caso contrário incrementar um contador, desistindo assim que chegue a 1000. Nesse repetir o passo 2, existindo também o limite de 1000 tentativas do próprio passo 2. No caso das tentativas do passo 2 se esgotarem, deixar a aposta como está, com alguns locais sem letras.
3. Completar o resto da aposta X da seguinte forma:
 1. Obter as letras não utilizadas da aposta Y na aposta X. Essas letras não podem ser utilizadas.
 2. Para cada letra em falta na aposta X, sortear uma letra (de 0 a W-1). No caso da letra estar nas letras que não se podem utilizar, obtidas no passo anterior, repetir o processo. Colocar um limite local para não ultrapassar as 1000 falhas de colocação de letras. Nesse caso desistir, e tal como no passo 2.2, repetir todo o passo 2, existindo o limite do passo 2 de 1000 tentativas.
4. Ver se a aposta é compatível com as restantes apostas anteriores (a aposta Y será compatível por construção), ou seja, a alínea C. Caso não seja, repetir todo o passo 2, existindo o limite do passo 2 de 1000 tentativas. No caso das tentativas do passo 2 se esgotarem, deixar a aposta como está, incompatível com outras apostas, e se for resultante de falha nos passos em que a aposta não esteja completa, colocar espaços nos locais onde ainda não há letras.

Simule um jogo que termine nas 10 tentativas, ou antes no caso da chave ser igual à última aposta.

Realço que a aposta X baseia-se na aposta anterior, pelo que por exemplo uma aposta 4 baseia-se na aposta 3.

Execução de exemplo:

K: 3

W: 3

Aposta 0: BCB (chave: CCB) p2 b0 incompatível: -1

Aposta 1: BCC (chave: CCB) p1 b2 incompatível: -1

Aposta 2: CCB (chave: CCB) p3 b0 incompatível: -1

K: 6

W: 4

Aposta 0: CCCABB (chave: BDCABA) p3 b1 incompatível: -1

Aposta 1: ACAACB (chave: BDCABA) p1 b3 incompatível: -1

Aposta 2: CACDDB (chave: BDCABA) p1 b3 incompatível: -1

Aposta 3: BDCABA (chave: BDCABA) p6 b0 incompatível: -1

K: 12

W: 4

Aposta 0: BDBDDCDADACB (chave: BDCABACCCABB) p4 b4 incompatível: -1

Aposta 1: CCDDDBCCACCB (chave: BDCABACCCABB) p3 b5 incompatível: -1

Aposta 2: AACCCBDAADDC (chave: BDCABACCCABB) p2 b6 incompatível: -1

Aposta 3: BBDCAAABDBBC (chave: BDCABACCCABB) p3 b7 incompatível: 0

Aposta 4: ADBDABABDCAB (chave: BDCABACCCABB) p2 b7 incompatível: 0

Aposta 5: BABADABDBBBB (chave: BDCABACCCABB) p5 b3 incompatível: 0

Aposta 6: DADCCABABDCB (chave: BDCABACCCABB) p2 b8 incompatível: 0

Aposta 7: DCBBBBABCDAD (chave: BDCABACCCABB) p2 b7 incompatível: 0

Aposta 8: BACBACBCACDD (chave: BDCABACCCABB) p3 b8 incompatível: 0

Aposta 9: CCACBCABABBD (chave: BDCABACCCABB) p2 b10 incompatível: 0

Aposta 10: BACDBCCAACB (chave: BDCABACCCABB) p7 b5 incompatível: 0

Na primeira execução de exemplo, pode-se ver a primeira aposta, aleatória mas com p2 b0. A segunda aposta (aposta 1), é compatível com a primeira aposta, e relativamente à chave tem p1 b2. A terceira aposta acabou por ser a chave, que é naturalmente compatível com todas as apostas realizadas. Sendo igual à chave, a execução pára. Na segunda execução, o algoritmo proposto necessitou de 4 apostas para encontrar a chave. O último exemplo, o algoritmo proposto não consegue realizar a aposta 3, tendo retornado, o que acabou por ser um resultado incompatível com a aposta 0. O mesmo sucedeu nas restantes apostas, até à aposta 10, tendo terminado com insucesso.