

## ☆ UAb E-fólio B, 1819, Alínea A

O e-fólio B 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 50% da nota destinado à avaliação da funcionalidade, resultante da percentagem de casos de teste corretos. A realização do e-fólio deve ser feita na plataforma *HackerRank*, sendo válidas apenas as submissões dadas como terminadas, e não dispensando 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, e 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.

Neste e-fólio deve alocar memória dinamicamente, dependente da entrada de dados (utilizando `malloc` e `free`). Caso pretenda fazer o e-fólio para metade da cotação, pode alocar memória de forma estática, independente da entrada de dados (penalização apenas nas alíneas C e D).

Este e-fólio tem por base um jogo de palavras, em que existe um conjunto de palavras seleccionadas, e tem de se escrever uma palavra que se inicie com a última letra da palavra anterior, não podendo serem repetidas palavras.

Considere que não existem nomes maiores ou iguais a 80 caracteres, mas deve fazer o programa para no caso de ser introduzido uma linha maior, possa ler os primeiros 79 caracteres. Deve considerar a entrada de dados pelo `stdin`.

Leia na alínea A uma palavra, e retorne o número de caracteres.

**Nota:** retire os caracteres no final da palavra que sejam espaços, tabs, ou fins de linha ("`\t\n`") da string, deixando o resto da string intacta.

Os casos de teste encontram-se visíveis, podendo ser utilizados para auxiliar a compreensão do enunciado.

### YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. ✕

Start tour

📘 For help on how to read input and write output in C, [click here](#). ✕

Original Code

C



```
1 #include <math.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <stdlib.h>
5 #include <assert.h>
6 #include <limits.h>
7 #include <stdbool.h>
8
9 int main() {
10
11     return 0;
12 }
13
14
```



Line: 1 Col: 1

1

Test against custom input

Run Code

Submit code & Continue

(You can submit any number of times)

2

[Download sample test cases](#) *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

3

4

## ☆ UAb E-fólio B, 1819, Alínea B

Leia agora um conjunto de palavras, uma palavra por linha, terminando quando a linha for vazia. Deve retornar:

- número de palavras lidas;
- número total de caracteres;
- número total de consoantes;
- número total de vogais.

Considere que a lista de consoantes é "BCDFGHJKLMNPQRSTVWXYZ" e a lista das vogais é "AEIOU". Considere maiúsculas e minúsculas como sendo iguais. Contabilize todos os restantes caracteres, no caso de ocorrerem, como não sendo nem consoantes nem vogais (letras acentuadas, traços, ou mesmo espaços).

São os casos de teste de modo a complementar o enunciado, e explicam-se os dois primeiros aqui.

Caso de teste 1:

Entrada

```
Codigo
```

Saída esperada

```
1 6 3 3
```

Neste caso, apenas foi dada uma palavra, pelo que o primeiro número a retornar é 1. A palavra tem 6 caracteres, pelo que o segundo número é 6. Tem 3 consoantes e 3 vogais, pelo que os outros dois números são 3 e 3.

Caso de teste 2:

Entrada

```
Código  
Linguagem  
Programa  
Instrução  
Atribuição  
Execução  
Output  
Input
```

Saída esperada

```
8 68 28 26
```

Neste segundo exemplo há 8 linhas com palavras, pelo que o primeiro número é 8. O número de letras, seriam 61 ( $6+9+8+9+10+8+6+5$ ), mas como as letras acentuadas são 7, e ocupam dois bytes, a contagem deve ser de 68 (utilizar simplesmente o `strlen` para contar as letras). O número de consoantes é de 28 ( $3+5+5+4+3+2+3+3$ ), atendendo a que as letras acentuadas não contam, e por esse motivo o número de vogais é de 26 ( $2+4+3+3+5+4+3+2$ ).

### YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. [Start tour](#)





1

2

3

4

Original Code

C



```
1 #include <math.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <stdlib.h>
5 #include <assert.h>
6 #include <limits.h>
7 #include <stdbool.h>
8
9 int main() {
10
11     return 0;
12 }
13
14
```

Line: 1 Col: 1

Test against custom input

Run Code

Submit code & Continue

(You can submit any number of times)



[Download sample test cases](#)

*The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

## ☆ UAb E-fólio B, 1819, Alínea C

Pretende-se na alínea C que após a leitura das palavras de acordo com a alínea B, em que pára ao encontrar uma palavra em branco, inicie de seguida a validação das palavras seguintes, de acordo com os seguintes pontos:

- Em cada linha tem de existir uma palavra do conjunto anterior, podendo cada palavra ocorrer apenas uma vez;
- Entre duas palavras seguidas A e B, a letra com que termina a palavra A tem de ser a letra com que se inicia a palavra B, caso contrário a palavra B é considerada inválida;
- O jogo termina após se inserir a primeira palavra inválida, sendo portanto contabilizadas apenas as palavras válidas, ou após se encontrar uma linha vazia;
- No final do jogo, tem de se dar as seguintes estatísticas relativas às palavras válidas: número de palavras; número de caracteres; número de consoantes; número de vogais; número de consoantes distintas; número de vogais distintas.

As estatísticas são iguais à alínea B, excepto o número de consoantes/vogais distintas. Pretende-se que sejam contadas de entre o conjunto das consoantes/vogais, quantas é que foram utilizadas em uma ou mais palavras. Por exemplo, se as vogais que ocorrem nas palavras forem as letras A, E e I, não existindo ocorrências das letras O e U, o número de vogais distintas é 3, independente do número total de vogais que pode ser um valor muito superior.

São disponibilizados os casos de teste, de modo a complementar o enunciado. São explicados os dois primeiros casos de teste.

Caso de teste 1:

Entrada

```
Codigo
Codigo
outra palavra
```

Saída esperada

```
1 6 3 3 3 2
```

Este caso de teste é idêntico ao caso de teste 1 da alínea B, sendo os primeiros 4 números iguais, dado que a palavra introduzida foi jogada, sendo rejeitada a segunda palavra que não existia. Os dois últimos números são o número de consoantes distintas, que são também 3, já que não há repetições, mas em termos de vogais, foram utilizadas duas vogais, as letras I e O, sendo a letra O utilizada duas vezes.

Caso de teste 2:

Entrada

```
Código
Linguagem
Programa
Instrução
Atribuição
Execução
Output
Input

Código
Output
Input
Código
```

Saída esperada



1

2

3

4

Neste caso, o jogo iniciou-se com a palavra **código**, seguida da palavra **output**. A letra de terminação/início é a mesma, a letra O, pelo que a segunda palavra é válida. No entanto, a terceira palavra, **input**, inicia-se com a letra I, quando a anterior palavra terminou com a letra T, pelo que é uma palavra inválida e a contagem deve parar de imediato. Existem portanto 2 palavras bem introduzidas, que têm 13 letras no total, das quais 6 são consoantes, e 5 são vogais. Em termos de letras distintas, há 5 consoantes distintas (a letra T foi utilizada duas vezes), e 3 vogais distintas, as vogais "IOU".

**YOUR ANSWER**

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. [Start tour](#) ✕

**i** For help on how to read input and write output in C, [click here](#). ✕

Original Code

C ▾

```
1 #include <math.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <stdlib.h>
5 #include <assert.h>
6 #include <limits.h>
7 #include <stdbool.h>
8
9 int main() {
10
11     return 0;
12 }
13
14
```

Line: 1 Col: 1

 Test against custom input

Run Code

Submit code &amp; Continue

(You can submit any number of times)

[Download sample test cases](#)*The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

## ☆ UAb E-fólio B, 1819, Alínea D

Na alínea D vamos colocar o computador a fazer as jogadas, após a entrada das palavras, devendo ser implementado o seguinte algoritmo:

- Passar todas as letras não acentuadas para maiúsculas
- Contar quantas palavras iniciam em cada letra -  $\text{FreqInicio}(letra)$
- Contar quantas palavras finalizam em cada letra -  $\text{FreqFim}(letra)$
- Ordenar as palavras com base na seguinte fórmula (ordem decrescente):  $\text{FreqInicio}(letra\text{-fim}) * \text{FreqFim}(letra\text{-inicio}) + \text{número de letras}$
- Mostrar as três primeiras palavras, indicando a importância.
- Escolher as palavras por ordem da lista, escolhendo sempre a primeira palavra válida e removendo-a da lista.
- Ao esgotar-se a lista ou ao não existir uma palavra válida, retornar as palavras selecionadas até ao momento;
- Contabilizar as estatísticas dessas palavras, de acordo com a alínea C.

No caso de uma palavra se iniciar ou terminar com uma letra acentuada, contabilizar a frequência como zero, pelo que a importância dessa palavra será apenas o número de letras.

São fornecidos os casos de teste de modo a complementar o enunciado. Explicam-se os dois primeiros casos de teste.

Caso de teste 1:

Entrada

```
Codigo
```

Saída esperada

```
Primeiras palavras:  
CODIGO [6]  
Jogo:  
CODIGO  
Estatísticas: 1 6 3 3 3 2
```

Neste primeiro caso, existindo uma só palavra, é mostrada como a primeira palavra precisamente esta palavra, e respetiva importância, que é igual apenas ao número de caracteres, já que apenas a  $\text{FreqInicio}(C)=1$ , e  $\text{FreqFim}(O)=1$ , tudo o resto é zero. A fórmula para a palavra é  $\text{FreqInicio}(O)*\text{FreqFim}(C)+\text{número de letra}$ , o que resulta em  $0*0+6$ . Notar que pretende-se valorizar as palavras que se iniciem por letras em que existem muitas a terminar, e que terminem em letras em que existam muitas palavras a iniciar, de modo a que existam muitas possibilidades do jogo continuar. De seguida apresenta-se o jogo, que consiste numa só palavra, terminando-se com as estatísticas finais. Notar ainda que a palavra foi transformada para maiúsculas.

Caso de teste 2:

Entrada:

```
Código  
Linguagem  
Programa  
Instrução  
Atribuição  
Execução  
Output  
Input
```

Saída esperada

```
Primeiras palavras:
```



ATRIBUIÇÃO

OUTPUT

Estatísticas: 2 18 6 8 4 4



1


2

3

4

Este caso de teste tem mais palavras, sendo mostradas as três com maior valor de importância. A primeira palavra tem 12 letras (as acentuadas contam a dobrar, utilizando simplesmente o `strlen`), tendo valor 13 porque  $\text{FreqInicio}(O)=1$  (palavra Output), e  $\text{FreqFim}(A)=1$  (palavra Programa). A segunda palavra tem importância 11, sendo o número de caracteres, já que não existem palavras a terminar pela letra I, ocorrendo o mesmo com a terceira palavra, tem a importância igual ao número de caracteres já que se inicia por uma letra em que nenhuma outra palavra termina. O jogo inicia-se com a palavra mais importante, após a qual existe uma única continuação com a palavra Output (caso contrário seria a palavra mais importante primeiro), e termina de imediato, já que não há palavras a iniciarem-se com a letra T. As estatísticas são dadas de acordo com a alínea C, sobre as duas palavras retornadas.

**YOUR ANSWER**

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. [Start tour](#) 

 For help on how to read input and write output in C, [click here](#). 

Original Code

C 

```
1 #include <math.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <stdlib.h>
5 #include <assert.h>
6 #include <limits.h>
7 #include <stdbool.h>
8
9 int main() {
10
11     return 0;
12 }
13
14
```

Line: 1 Col: 1

 Test against custom input

Run Code

Submit code &amp; Continue

(You can submit any number of times)

[Download sample test cases](#)*The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*





1

2

3

4