

E-fólio B

No e-fólio B pretende-se implementar um jogo de palavras. Duas palavras extraídas do dicionário¹ têm em comum o sufixo de uma com o prefixo da outra. É mostrada a parte não comum de ambas as palavras, sendo o desafio a adivinhação da parte comum oculta. A parte comum deve ser única. Por exemplo, a pergunta poderá ser “assi_____lizar”, sendo a resposta “natura”, resultando nas duas palavras “assinatura” e “naturalizar”.

Existem 4 alíneas, que devem ser realizadas sequencialmente, valendo cada alínea 1 valor. Os critérios de correção estão publicados no espaço da UC. Caso não consiga resolver todo o e-fólio, deve deixar por realizar as últimas alíneas. Deve entregar um único ficheiro c, com o seu número de estudante, independentemente das alíneas que realizou. No relatório deve indicar as alíneas realizadas, devendo o programa responder corretamente de acordo com a última alínea realizada (não necessita manter a interface das alíneas anteriores).

Não deve assumir nenhum valor máximo para o número de palavras nem o tamanho das palavras. Pode no entanto assumir que um nome não tem mais de 255 caracteres, mas não deve alocar mais memória do que o necessário. Deve alocar memória dinamicamente, dependente do ficheiro e dados introduzidos pelo utilizador. Caso não deixe em aberto o número de palavras e seu tamanho, utilize um valor máximo de 10.000 palavras, e para cada nome utilize 80 caracteres, sendo no entanto aplicável uma penalização de 50% nas alíneas C e D.

Deve entregar um relatório, descrevendo o trabalho realizado, os testes que fez, as opções que tomou, o qual não deve ultrapassar as 4 páginas nem conter o código fonte. Se realizou apenas parcialmente uma dada alínea, descreva o que fez nessa alínea no relatório. Não apresente um programa que não compile no TCC.

Alínea A) Contar palavras (1 valor)

Nesta alínea pretende-se que receba o nome de um ficheiro, e indique o número de linhas que o ficheiro tem. Pode assumir que nenhuma linha terá mais que 250 caracteres.

Na primeira execução de exemplo não foi fornecido o ficheiro com as palavras, sendo indicado o modo de utilização. Na segunda execução o ficheiro foi lido e indicado o número de linhas.

Execuções de exemplo:

```
C:\...>111111
```

```
Utilizacao: 111111 <ficheiro de palavras>  
C:\...>111111 pt-BR-antigo.dic
```

```
Palavras: 40889
```

¹ Dicionário de palavras utilizado, de <https://addons.mozilla.org/pt-br/firefox/addon/ortografia-br/>

Alínea B) Palavra aleatória (1 valor)

Pretende-se nesta alínea que imprima uma palavra aleatória do ficheiro, que não tenha nem letras acentuadas, nem letras maiúsculas, ou seja, deve conter apenas letras minúsculas. No entanto, se a única letra maiúscula for a primeira, a palavra deve ser considerada, convertendo essa letra para minúscula. Mostre ainda quantas palavras existem nestas condições. A semente aleatória deve ser inicializada com o tempo, de modo a poder-se ter execuções distintas. O ficheiro de palavras fornecido, termina a palavra com o carácter '/', pelo que este carácter e os seguintes, não devem ser considerados.

Nas duas execuções de exemplo pode-se ver o mesmo número de palavras válidas lido, e duas palavras distintas selecionadas, "adoecer" e "rocha", ambas constituídas por apenas palavras com letras minúsculas.

Execuções de exemplo:

```
C:\...>111111 pt-BR-antigo.dic
```

```
Palavras validas: 29155
```

```
Palavra: adoecer
```

```
C:\...>111111 pt-BR-antigo.dic
```

```
Palavras validas: 29155
```

```
Palavra: rocha
```

Alínea C) Maior prefixo comum (1 valor)

Pretende-se nesta alínea a apresentação da palavra aleatória com uma parte oculta. Assuma que as palavras estão no ficheiro por ordem alfabética. Para a palavra aleatória, determinar o maior prefixo comum, por comparação da palavra anterior e seguinte. Deve-se mostrar a palavra com o prefixo comum visível e o sufixo distinto oculto. Se o dicionário for completo, a parte oculta deverá ser passível de ser descoberta por alguém que conheça todas as palavras. Para efeitos de testes deve também apresentar a palavra, bem como a palavra anterior e a palavra seguinte que serviram de base à determinação da parte oculta. Caso o resultado não inclua pelo menos dois caracteres ocultos, deve-se repetir o processo gerando aleatoriamente outra palavra, até que essa situação se verifique.

Nas duas execuções de exemplo pode-se ver no primeiro caso, a palavra "ajustar", que tem em comum com a palavra anterior, "aju", e com a palavra seguinte "ajust", pelo que fica oculto apenas "ar". No segundo caso, a palavra "condigno" tem em comum com as restantes palavras "condi", pelo que os três últimos caracteres estão ocultos.

Execução de exemplo:

```
C:\...>111111 pt-BR-antigo.dic
```

```
Palavra: ajust**
```

```
Palavras: ajuntar ajustar ajuste
```

```
C:\...>111111 pt-BR-antigo.dic
```

```
Palavra: condi***
```

```
Palavras: condicionar condigno condimentar
```

Alínea D) Gerador de perguntas (1 valor)

Nesta alínea pretende-se que faça um gerador de perguntas, que gere K pares de pergunta resposta cada vez que é executado, com um número W de caracteres mínimos ocultos. Para construir os pares de pergunta resposta, deve-se respeitar o seguinte:

- Obter uma palavra aleatória com o maior prefixo comum (e sufixo oculto), de acordo com a alínea C – palavra A;
- Obter uma palavra B cujo prefixo seja igual ao sufixo oculto da palavra A, e cujo o seu maior prefixo comum (condições da alínea C), seja igual ou superior ao número de caracteres do sufixo oculto da palavra A.
- Caso não exista uma palavra B para a palavra A gerada, deve-se repetir o processo gerando outra palavra A.
- As palavras A e B devem ser mostradas iniciando-se com a parte visível da palavra A, parte oculta comum de ambas as palavras, e parte visível da palavra B.
- Deve aceitar apenas partes comuns com pelo menos W caracteres ocultos.

Para obter a palavra B, as palavras devem ser testadas por ordem aleatória, de modo a que quaisquer dois pares de palavras A e B possam ter a mesma probabilidade de ocorrer. Tanto o número de perguntas (K), como o número de caracteres ocultos (W), deve ser especificados pelo utilizador nos parâmetros do programa.

Na execução de exemplo, pode-se ver a primeira execução com 10 perguntas, com o mínimo de 3 caracteres ocultos. Por exemplo, a palavra dilema e emmanuel, partilham “ema”. Na segunda execução é solicitado 4 perguntas, com no mínimo 4 caracteres ocultos, e na última é solicitado 4 perguntas com 6 ou mais caracteres ocultos.

Execuções de exemplo:

```
C:\...>111111 pt-BR-antigo.dic 10 3
```

```
Pergunta: dil__nuela   Resposta: dilema emmanuel
Pergunta: ria__quiteiro Resposta: riamos mosquito
Pergunta: serif__giar  Resposta: serifada adagiar
Pergunta: quadr__bela  Resposta: quadrilha ilhabela
Pergunta: ak__nar      Resposta: akamai amainar
Pergunta: cons__lerante Resposta: consinto intolerante
Pergunta: rus__ibaldo  Resposta: rusgar garibaldi
Pergunta: ai__polho    Resposta: aipim pimpolho
Pergunta: advenha__quito Resposta: advenhamos mosquito
Pergunta: adr__tecer   Resposta: adrenal enaltecer
```

```
C:\...>111111 pt-BR-antigo.dic 4 4
```

```
Pergunta: frag__lado  Resposta: fragmento mentolado
Pergunta: anti__izar   Resposta: anticonstitucional constitucionalizar
Pergunta: dia__rian    Resposta: diamond mondrian
Pergunta: mane__icamente Resposta: manequim quimicamente
```

```
C:\...>111111 pt-BR-antigo.dic 4 6
```

```
Pergunta: poli__glicol   Resposta: polietileno etilenoglicol
Pergunta: hiper__logicamente Resposta: hipermercado mercadologicamente
Pergunta: poli__glicol   Resposta: polietileno etilenoglicol
Pergunta: peri__hadora   Resposta: perimetral metralhadora
```