

Programação I 2016/2017

Departamento de Informática, Universidade de Évora

Trabalho prático

27 de dezembro de 2016 (v1.0)

– Sopa de Letras –

1 Introdução

O jogo *Sopa de Letras* tem como objetivo localizar numa grelha, uma lista de palavras previamente fornecida. Normalmente as palavras podem aparecer escritas na grelha, em todas as direcções (vertical, horizontal e diagonal) e sentidos (norte↔sul, este↔oeste, noroeste↔sudeste e nordeste↔sudoeste).

Consideremos, por exemplo, a Sopa de Letras de tamanho 9x9 representada na figura seguinte e a lista de palavras: branco, preto, azul, verde, rosa, amarelo, vermelho, cinza e lilas.

	a	b	c	d	e	f	g	h	i
1	O	H	L	E	M	R	E	V	U
2	B	B	R	A	N	C	O	Z	A
3	S	R	A	M	S	U	P	A	O
4	A	A	B	A	P	O	T	Z	Z
5	L	N	Z	R	O	E	R	N	U
6	I	U	I	E	P	D	O	I	I
7	L	O	L	L	O	R	A	C	A
8	I	T	M	O	T	E	R	P	P
9	L	I	E	A	Z	V	Y	U	U

A solução do jogo indica a localização daquelas palavras na matriz. Para o exemplo, a localização pode ser indicada da seguinte forma:

- branco: b2-g2, este
- preto: h8-d8, oeste
- azul: d4-a7, sudoeste
- verde: f9-f5, norte
- rosa: g5-d2, noroeste
- amarelo: d2-d8, sul
- vermelho: h1-a1, oeste
- cinza: h7-h3, norte
- lilas: a7-a3, norte

2 Descrição do trabalho

Considere a existência de 2 aplicações distintas:

1. **localizar** as palavras numa sopa de letras;
2. **gerar** uma sopa de letras;

A primeira aplicação é de desenvolvimento obrigatório e tem a cotação máxima de 15 valores; a segunda é optativa e tem a cotação de 5 valores. Apenas deverá desenvolver a segunda aplicação após a conclusão com sucesso da primeira.

2.1 Localizar palavras

Esta aplicação solicita ao utilizador o nome do ficheiro de texto que contém a lista de palavras e a grelha e mostra no ecrã a lista de palavras (uma por linha) e a sua localização na grelha. Esta informação deve seguir o formato apresentado na secção anterior.

O ficheiro de entrada contém a seguinte informação:

- na primeira linha é indicado o nº de palavras existentes na grelha;
- seguem-se as palavras, uma em cada linha;
- depois é fornecida a grelha de letras

2.2 Gerar sopa de letras

Esta aplicação tem como entrada um ficheiro de texto e como saída um outro ficheiro de texto. Os nomes dos ficheiros devem ser solicitados ao utilizador.

O ficheiro de entrada contém a seguinte informação:

- na primeira linha é indicado o tamanho da grelha (nº de linhas, nº de colunas);
- seguem-se as palavras a incluir na grelha, uma em cada linha

O ficheiro de saída deve ter o formato compatível para ser utilizado pela aplicação de localização:

- na primeira linha é indicado o nº de palavras existentes na grelha;
- seguem-se as palavras, uma em cada linha;
- depois é indicada a sopa de letras gerada

3 Desenvolvimento

O desenvolvimento do trabalho inclui duas fases a serem entregues em instantes de tempo diferentes:

Fase 1 Desenho da aplicação. Compreende a sua estruturação em funções com especificação clara dos argumentos e valor de retorno de cada função.

Fase 2 Implementação da aplicação. Inclui a implementação e teste de cada uma das funções, começando pelas mais simples e que não chamam outras funções.

O programa deverá ser desenvolvido em **Python** por grupos de **três alunos**. Qualquer uma das fases inclui a entrega de um relatório em PDF.

3.1 Fase 1

O desenho da aplicação inclui:

1. Perceber o problema, identificando inequivocamente o input e output do programa;
2. Pensar numa solução, que inclui dividir o problema noutros mais simples, encontrar a solução para estes e, a partir destas, contruir a solução geral

Como resultado desta fase, deverá obter:

1. o programa estruturado através de um conjunto de funções, com especificação clara dos argumentos e valor de retorno de cada função;
2. um fluxograma que descreva os principais passos executados pelo programa e cada uma das funções.

3.2 Fase 2

Esta fase compreende a codificação e teste de cada uma das funções desenhadas na fase anterior, bem como do programa principal. Nesta fase é essencial **comentar o código**. Para tal utilize o seguinte formato:

```
#####
# area - Esta funcao calcula a area de um retangulo
#
# Argumentos:
# l - largura, inteiro
# c - comprimento, inteiro
# Valor de retorno:
# area calculada, inteiro
#####
def area(l, c):
    return l*c
```

4 Relatório

O texto do relatório deve ser escrito de forma impessoal, isto é, usando voz passiva, focando nos aspectos técnicos (e não nos autores). Exemplos:

- **Certo:** Foi encontrado um erro...
- **Errado:** Encontrámos um erro...
- **Certo:** Nos testes efectuados verificou-se que o programa não funciona correctamente se...
- **Errado:** Reparámos que o programa não funciona correctamente se...
- **Certo:** A função xpto foi usada para...
- **Errado:** Usámos a função xpto para...

A organização do relatório e a qualidade do texto também são avaliadas. Sugere-se a escrita sua escrita em L^AT_EX e a utilização dos editores online Sharelatex¹ ou Overleaf².

¹<https://www.sharelatex.com/>

²<https://www.overleaf.com/>

4.1 Fase 1

O relatório deverá indicar se o trabalho inclui a implementação de ambas as aplicações (localizar palavras + gerar sopa de letras) ou apenas a aplicação para localizar as palavras.

Deve depois fazer a descrição da solução proposta e deve incluir toda a informação relativa ao resultado desta fase: a lista de funções a serem implementadas e os fluxogramas respetivos (exeto para as mais simples).

Para cada função proposta deverão ser especificados os argumentos recebidos, o valor retornado, bem como uma descrição detalhada da funcionalidade a implementar. Para ter uma ideia do que se pretende, veja as *man pages* em Linux:

```
$ man puts
```

4.2 Fase 2

Para além de outra informação que for considerada relevante, este relatório deve incluir:

- informação sobre as alterações (se existirem) ao desenho proposto na fase anterior;
- a listagem atualizada de todas as funções implementadas, especificando os argumentos recebidos, o valor retornado e uma descrição detalhada da funcionalidade implementada.

5 Prazos e Entrega

Como já referido, o trabalho deve ser desenvolvido por grupos de **três** alunos, em duas fases e entregues em duas datas distintas (definidas no moodle):

Fase 1 – Tarefas:

- Desenhar a solução, com a identificação das funções necessárias ao desenvolvimento da solução;
- Escrever o relatório

Fase 2 – Tarefas:

- Implementar a solução proposta, com alterações se necessário;
- Completar o relatório

No moodle é submetido apenas um ficheiro com nome definido pelos números dos alunos (por exemplo `12345-67890-45787.tar.gz`). Na Fase 2 o ficheiro deve incluir o código implementado e o relatório. Os ficheiros podem ser comprimidos com o comando `tar`. Um exemplo da sua utilização pode ser:

```
$ tar cvfz 12345-67890-45787.tar.gz sopadeletras.py relatorio.pdf
```

Bom trabalho!