

Programação I 2017/2018

Departamento de Informática, Universidade de Évora

Trabalho 1

20 de dezembro de 2017 (v1.0)

– Pigs and Bulls –

1 Introdução

Pigs and Bulls é um jogo para 2 jogadores cujo objetivo é adivinhar o código pensado por um dos jogadores. Na versão original o código é composto por 4 algarismos escolhidos entre 0 e 9, sem repetição.

Um dos jogadores pensa num código; o outro jogador propõe um código e o primeiro indica o número de acertos. Existem duas indicações distintas para acertos: se os algarismos estão nas posições certas, são "touro"; se estiverem em posições diferentes, são "porcos". Por exemplo, se o código for **4271** e a tentativa for **1234**, a resposta deve ser **1 touro; 2 porcos** (o touro é "2", os porcos são "4" e "1").

2 Descrição do trabalho

O trabalho consiste em desenvolver uma aplicação para jogar *Pigs and Bulls* com o computador: o computador deve gerar o código e o jogador vai fazendo tentativas até acertar. No final o computador deve apresentar número total de tentativas bem como a sequência de tentativas (e respetivos acertos).

Utilize o seguinte exemplo de interação como guia:

```
>> Código? 1234
1T
>> Código? 1567
3P
>> Código? 5276
3P
>> Código? 7635
2T 1P
>> Código? 7654
Acertou!!!

As suas tentativas foram:
T1: 1234, 1T
T2: 1567, 3P
T3: 5276, 3P
T4: 7635, 2T 1P
T5: 7654, 4T
```

3 Desenvolvimento

O programa deverá ser desenvolvido em **Python** e ser acompanhado por um relatório **PDF**.

Para gerar um número aleatório deve utilizar a função `random()` do módulo `random`. Esta função gera um número aleatório no intervalo $[0, 1[$.

É importante dividir o trabalho em duas fases:

Fase 1 Desenho da aplicação. Compreende a 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).

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 o programa estruturado através de um conjunto de funções, com especificação clara dos argumentos e valor de retorno de cada função.

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

A organização do relatório e a qualidade do texto também são avaliadas. 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...

Deve fazer a descrição da solução proposta e incluir toda a informação relativa ao seu desenvolvimento, incluindo as variáveis utilizadas (e respetivos tipos) e a lista de funções implementadas especificando os argumentos recebidos, o valor devolvido e uma descrição detalhada da funcionalidade implementada. Para ter uma ideia do que se pretende, veja as *man pages* em Linux usando o comando `man`; experimente, por exemplo, `man puts`.

5 Prazos e Entrega

O trabalho deve ser desenvolvido **individualmente** e submetido no moodle até ao final do dia **2 de janeiro**. A **apresentação** do trabalho será feita na aula prática de janeiro.

A submissão deve incluir o ficheiro de código bem como o relatório. Para tal deverá comprimir os ficheiros para um de nome definido pelo número do aluno. Os ficheiros podem ser comprimidos com o comando `tar`. Um exemplo da sua utilização pode ser:

```
$ tar cvfz 12345.tar.gz pigsandbulls.py relatorio.pdf
```

Bom trabalho!