



Introdução à Programação e Resolução de Problemas

2016/2017

Exame Recurso

26 de janeiro de 2017 - 9h00, Duração: 2h30m

Nome:

Número:

AVISO

Este exame especial tem a duração de 2h30m. O exame é composto por quatro perguntas. As perguntas devem ser resolvidas sem recurso ao computador. As respostas às perguntas devem ser dadas no espaço reservado para o efeito. Como material de consulta apenas pode usar o manual de referência do Python. A violação desta última regra implica a anulação da prova, reprovação à cadeira e eventual processo disciplinar. Caso tenha alguma dúvida pergunte. Boa sorte!

Pergunta	Pontos	Resultado
1)	10	
2)	15	
3)	20	
4)	25	
5)	30	
	100	

Visto por:

Nome:

Número:

Pergunta 1 **10 pontos**

Considere o programa da listagem abaixo.

```
1 def xpto(lista):  
2     for i in range(len(lista)):  
3         m = max(lista[i:])  
4         ind = lista.index(m)  
5         lista[i], lista[ind] = lista[ind], lista[i]  
6     return lista
```

Diga para que serve, como funciona e se, eventualmente, tem algum erro. Neste último caso identifique-o e diga como o podia corrigir face ao objetivo que identificou.

A sua resposta:

A sua resposta:

Nome:

Número:

Pergunta 2 15 pontos

Escreva um programa que recorrendo ao módulo **Turtle** permite desenhar imagens como as ilustradas na figura.

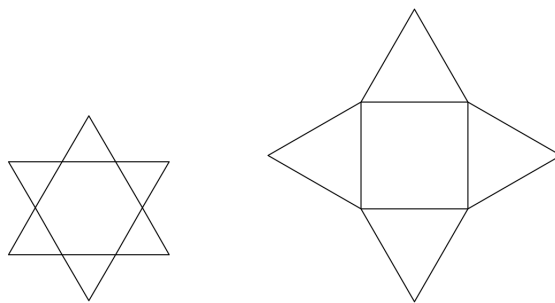


Figura 1: Triângulos às voltas.

Notar que deve ser possível parametrizar o número de triângulos equiláteros e a dimensão do respectivo lado.

A sua resposta:

A sua resposta:

Nome:

Número:

Pergunta 3 20 pontos

Escreva um programa que recebe uma lista de tuplos em que cada tuplo tem o formato (caractere, posição, repetições). Cada tuplo da lista indica para um dado caractere a posição inicial em que aparece e o número de vezes consecutivas que é repetido. A partir desta lista de tuplos, deverá gerar a cadeia de caracteres correspondente. Note que a lista de tuplos poderá não estar ordenada pela posição em que os caracteres aparecem. Por exemplo:

```
1 >>> lista = [('2', 1, 1), ('a', 10, 0), ('x', 0, 0), ('d', 3, 2), (' ',  
    6, 0), ('c', 7, 0), ('b', 8, 1)]  
2 >>> gera_cadeia(lista)  
3 'x22ddd cbba'
```

A sua resposta:

A sua resposta:

Nome:

Número:

Pergunta 4 **25 pontos**

Considere uma rede social como o twitter. Podemos representar a rede de seguidores através de um dicionário em que a chave é o nome de um utilizador, e o valor é a lista dos utilizadores que o primeiro segue. Note que neste tipo de rede social, A seguir B, não implica que B siga A. Pretendemos que o sistema permita fazer recomendações de novos utilizadores a seguir. Escreva um programa que, recebendo o nome de um utilizador u , um dicionário com toda a rede, e um valor inteiro n , devolva a lista de sugestões para esse utilizador. Devemos sugerir a u todos os utilizadores que ainda não sejam seguidos por si, e que pelo menos n dos seus seguidos sigam. Por exemplo:

```
1 >>> rede = {'tiago': ['rita', 'francisco', 'joao', 'filipa'], 'joao': ['  
    tiago', 'ricardo'], 'ricardo': ['rita', 'francisco'], 'rita': ['tiago'  
    , 'filipa', 'ricardo'], 'filipa': [], 'francisco': ['tiago']}
```

```
2 >>> sugestoes('joao', rede, 2))  
3 ['francisco', 'rita']
```

A sua resposta:

A sua resposta:

Nome:

Número:

Pergunta 5 **30 pontos**

Considere que tem um ficheiro com os resultados da prova de Triatlo Olímpico Masculino do Rio2016. O ficheiro contém o nome do atleta, o país e o tempo total das 3 provas (natação, ciclismo e corrida) no formato **h:m:s** como se indica no exemplo abaixo. Pretendemos criar um novo ficheiro com os atletas que ficaram nos 5 primeiros lugares. Cada linha deste ficheiro é formada pela classificação do atleta, o nome do atleta, e a diferença em segundos relativamente ao 1º classificado.

```
1 Richard Inglaterra 1:45:50
2 Thomas USA 1:46:09
3 Jonathan Inglaterra 1:45:07
4 Alistair Inglaterra 1:45:01
5 João Portugal 1:45:52
6 Pierre France 1:47:03
7 Konstantin Austria 1:48:54
8 Mário Portugal 1:48:29
9 Schoeman AfricaSul 1:45:53
10 Murray USA 1:46:34
```

A sua resposta:

A sua resposta: