

**LISTA DE EXERCÍCIOS 4- PROGRAMAÇÃO 1**  
**PROF. FERNANDO CASTOR**  
**SISTEMAS DE INFORMAÇÃO – CIN – UFPE 2017.1**

**Questão 1:**

Crie um programa que armazene uma lista de inteiros de tamanho máximo 25. Os inteiros devem ser adicionados à lista individualmente, o programa deve parar de receber inteiros quando a lista atingir tamanho 25 ou o usuário não digitar nada. Depois o programa deve retornar a soma de todos os inteiros, a média de todos, o maior inteiro e o menor inteiro.

**Exemplo 1:**

Formato de entrada:

12  
34  
52  
13  
65  
3452  
23

*<<Entrada em branco para finalizar a lista*

Formato de saída:

Soma dos itens da lista: 3651  
Média dos itens da lista: 521.5714  
Maior item da lista: 3452  
Menor item da lista: 12

**Exemplo 2:**

Formato de entrada:

131  
235  
768  
12  
12  
-1  
-435  
123  
124  
456  
321

<<Entrada em branco para finalizar a lista

### Formato de Saída:

Soma dos itens da lista: 1746

Média dos itens da lista: 158.7273

Maior item da lista: 768

Menor item da lista: -435

*Obs: a média só precisa ter 4 casas decimais. Não é permitido o uso de `min()` , `max()` , `sort()` ou `sum()`.*

### Questão 2:

Bem-vindo(a) ao Python Land! Nos capítulos anteriores os aventureiros desse mundo maravilhoso registraram com sucesso o nome, classe e status do seus heróis, porém Python Land é um mundo fantástico e perigoso! Sair por aí sem conhecimento algum pode ser problemático; imaginem um novato se deparando com um Beholder no seu primeiro encontro!? Preocupado com isso, o mestre de jogo deseja que vocês, aventureiros, criem um bestiário (um livro de registro de todas as criaturas, nele existem vários dados úteis para que um aventureiro de primeira viagem conheça os monstros que poderá enfrentar).

O bestiário deve ser uma lista de monstros onde cada monstro possui uma lista com seu nome, uma lista com seus status, uma lista contendo suas características e outra lista contendo suas habilidades.

**Nome:** todo monstro deve possuir um nome. Esse nome não deve ser composto nem vazio (ou seja, o nome não pode ser “”).

**Status:** força, agilidade, inteligência e sabedoria. Todo monstro deve possuir esses status e ele deve ser um número  $\geq 0$ .

**Características:** todo monstro possui características e pelo menos uma, e quantas características forem necessárias ou seja um N indefinido

**Habilidades:** deve haver pelo menos 1 item para habilidade na lista, se essa habilidade não existir, a string ‘nenhum’ deve ser inserida no lugar.

- Todos os campos deverão estar preenchidos.
- Apenas os stats devem estar separados por vírgula.
  - Cada monstro pode ter mais de um tipo.
  - Cada monstro pode ter mais de uma habilidade.
- Se o tipo ou a habilidade for um nome composto sem hífen, as palavras que o compõem devem ser separadas por vírgula e sem espaços como no exemplo abaixo.

### **Exemplo 1:**

#### Formato de entrada:

medusa 0,0,0,0 monstro petrificar

#### Formato de saída:

Criatura: medusa

Stats: Força: 0 Habilidade: 0 Inteligencia: 0 Sabedoria: 0

Características: monstro

Habilidades: petrificar

### **Exemplo 2:**

#### Formato de entrada:

medusa 1,2,3,4 mitologico petrificar minotauro 200,1,0,0 besta,humanoide chifrada,mugido

#### Formato de saída:

Criatura: medusa

Status: Força: 1 Agilidade: 2 Inteligência: 3 Sabedoria: 4

Características: mitologico

Habilidades: petrificar

Criatura: minotauro

Status: Força: 200 Agilidade: 1 Inteligência: 0 Sabedoria: 0

Características: besta humanoide

Habilidades: chifrada mugido