



Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação  
Disciplina Fundamentos de Programação**

**AD1 – 2º semestre de 2019**

---

**IMPORTANTE**

- As respostas (programas) deverão ser entregues pela plataforma em um arquivo ZIP contendo todos os arquivos de código fonte (extensão “.py”) necessários para que os programas sejam testados. Respostas entregues fora do formato especificado, por exemplo, em arquivos com extensão “.pdf”, “.doc” ou outras, não serão corrigidas.
- Serão aceitos apenas soluções escritas na linguagem Python 3. Programas com erro de interpretação não serão corrigidos. Evite problemas utilizando tanto a versão da linguagem de programação (Python 3.X) quanto a IDE (PyCharm) indicadas na Aula 1.
- Quando o enunciado de uma questão inclui especificação de formato de entrada e saída, tal especificação deve ser seguida à risca pelo programa entregue. Atender ao enunciado faz parte da avaliação e da composição da nota final.
- Faça uso de boas práticas de programação, em especial, na escolha de identificadores de variáveis, subprogramas e comentários no código.
- As respostas deverão ser entregues pela atividade "Entrega de AD1" antes da data final de entrega estabelecida no calendário de entrega de ADs. Não serão aceitas entregas tardias ou substituição de respostas após término do prazo.
- As ADs são um mecanismo de avaliação individual. As soluções podem ser buscadas por grupos de alunos, mas a redação final de cada prova tem que ser individual. Respostas plagiadas não serão corrigidas.

---

**Boa Avaliação!**

### 1ª Questão (1,5 pontos)

Faça um programa que leia strings da entrada padrão até que a string vazia ("" ) seja digitada. Suponha que cada string não vazia seja convertível em um número de ponto flutuante.

Escreva:

- (1) A quantidade de strings não vazias lidas;
- (2) Caso exista: a média dos valores, das strings numéricas lidas;
- (3) Caso exista: o valor do maior número lido.

### 2ª Questão (1,5 pontos)

Utilizando subprogramação, faça um programa que leia da entrada padrão uma linha podendo conter vários números de ponto flutuante. Caso exista, escreva qual o número que mais ocorreu e a quantidade de vezes que ocorreu. Caso haja empate escreva um deles. Caso a linha lida seja uma string vazia, escreva a mensagem: "nenhum número foi lido!!!"

### 3ª Questão (2,0 pontos)

Utilizando subprogramação, faça um programa que leia da entrada padrão, em uma única string, dois números inteiros separados por um espaço em branco, representando as duas dimensões de uma matriz bidimensional. Construa uma matriz bidimensional com as dimensões lidas e com valores gerados aleatoriamente, via função `randint` importada do módulo `random`, no intervalo 100 a 999.

Escreva:

- (1) O conteúdo da matriz, onde cada linha a ser escrita possua apenas números inteiros, sem vírgulas nem colchetes;
- (2) A posição (linha, coluna) do maior valor sorteado e o seu respectivo valor;
- (3) A soma de cada linha, de todas as linhas da matriz;
- (4) A soma de cada coluna, de todas as colunas da matriz.

### 4ª Questão (1,5 pontos)

A conversão de números inteiros para diferentes bases é uma operação realizada frequentemente em computação. Por exemplo, no dia a dia estamos habituados a trabalhar com números na base decimal. Entretanto, o computador opera na base binária. Enquanto, eventualmente, a inspeção visual do conteúdo da memória do computador é feita na base octal ou hexadecimal.

Faça um programa que, dados valores inteiros na binária, escreva na saída padrão cada valor convertido para as bases 3 a 10.

Seu programa deve conter um subprograma que respeite o seguinte protótipo:

```
def converte(numBinario, base):  
    ...  
    return numConvertido
```

### Entrada

A entrada é composta por várias linhas, cada uma contando um valor inteiro binário não negativo com até 8 dígitos. A última linha contém o valor -1, que não deve ser processado.

### Saída

Uma linha deve ser emitida na saída padrão para cada valor dado como entrada. Essa linha deve conter as oito representações do número, uma para cada base, separados por um espaço em branco. As conversões devem ser apresentadas em ordem crescente de base.

### Restrições

Não é permitido o uso de rotinas de conversão nativas do Python.

### Exemplo

Entrada	Saída
1101	111 31 23 21 16 15 14 13
1100010	10122 1202 343 242 200 142 118 98
0	0 0 0 0 0 0 0 0
-1	

### **5ª Questão (1,5 pontos)**

Considere a existência de uma linha vertical que representa o intervalo  $[0, 2^n]$ . Agora desenhe traços verticais apoiados nos pontos  $1, 2, 3, \dots, 2^n - 1$  da linha vertical imaginária. O traço vertical no ponto médio do intervalo deve ter altura  $n$ , os traços nos pontos médios dos subintervalos esquerdo e direito têm altura  $n-1$ , e assim por diante. Diremos que isso é uma régua de ordem  $n$  no intervalo  $[0, 2^n]$ .

Dado o valor de  $n$ , escreva um programa recursivo que escreva tal régua na saída padrão.

### Entrada

A entrada é composta por um único valor  $n$  inteiro não negativo.

### Saída

A régua impressa da saída padrão com o formato conforme apresentado no exemplo.

### Exemplo

Entrada	Saída
4	01 - 02 -- 03 - 04 --- 05 - 06 -- 07 - 08 ---- 09 - 10 -- 11 - 12 --- 13 - 14 -- 15 -

### **6ª Questão** (2,0 pontos)

Faça um programa que peça ao usuário as dimensões de uma matriz bidimensional, chamada de `valores`, de números inteiros a ser gerada em aleatoriamente no intervalo 10 a 99. Via subprogramação:

- Mostre a matriz gerada;
- Ordene as linhas da matriz em ordem crescente da soma de seus valores;
- Mostre a matriz ordenada.

Nesse programa, seu programa DEVE implementar e utilizar o método da partição.

### Entrada

A entrada é composta por uma primeira linha, que define a quantidade `L` de linhas e a quantidade `C` de colunas da matriz a ser gerada.

### Saída

`L` linhas, onde cada linha possui `C` valores inteiros no intervalo 10 a 99 que correspondem à matriz original sorteada; seguida(s) de uma linha em branco; e por `L` linhas, onde cada linha possui `C` valores que correspondem à matriz ordenada.

### Exemplo

Entrada	Saída
5 6	13 28 45 50 26 10 27 24 22 33 88 11 90 25 85 23 76 55 77 15 31 29 13 14 66 41 50 20 47 11  13 28 45 50 26 10 77 15 31 29 13 14 27 24 22 33 88 11 66 41 50 20 47 11 90 25 85 23 76 55