



**Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância**

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina Fundamentos de Programação**

**Professores: Dante Corbucci Filho e Luís Felipe Ignácio Cunha**

**APX1 1º semestre de 2021**

---

### **IMPORTANTE**

**MUITA ATENÇÃO:** Nesta APX há, na próxima página, o TERMO de CONDUTA que todo aluno deve estar ciente e assinar devidamente o documento. Você pode redigi-lo e assinar manuscrito ou assinar digitalmente, e deve fazer upload deste termo juntamente com o arquivo ZIP das suas respostas. Caso o TERMO de CONDUTA não seja devidamente entregue e assinado, a APX1 não será corrigida e a nota da APX1 será zero.

- 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.
  - Os exemplos fornecidos nos enunciados das questões correspondem a casos específicos apontados para fins de ilustração e não correspondem ao universo completo de entradas possíveis especificado no enunciado. Os programas entregues devem ser elaborados considerando qualquer caso que siga a especificação e não apenas os exemplos dados. Essa é a prática adotada tanto na elaboração das listas exercícios desta disciplina quanto no mercado de trabalho.
  - 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 específica na Plataforma antes da data final de entrega estabelecida. Não serão aceitas entregas tardias ou substituição de respostas após término do prazo.
  - As APXs 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.
-



## TERMO DE CONDUTA

Declaro assumir o compromisso de confidencialidade e de sigilo escrito, fotográfico e verbal sobre as questões do exame ou avaliação pessoal que me serão apresentadas, durante o curso desta disciplina. Comprometo-me a não revelar, reproduzir, utilizar ou dar conhecimento, em hipótese alguma, a terceiros, e a não utilizar tais informações para gerar benefício próprio ou de terceiros. Reitero minha ciência de que não poderei fazer cópia manuscrita, registro fotográfico, filmar ou mesmo gravar os enunciados que me são apresentados. Declaro, ainda, estar ciente de que o não cumprimento de tais normas caracteriza infração ética, podendo acarretar punição de acordo com as regras da minha universidade.

Ciente,

Assinatura do Aluno

**Questão 1 – (Valor 1.5 pontos)**

Uma pessoa deseja subir uma escada. Suponha que cada passo a ser dado possa cobrir ou um único degrau ou dois degraus de uma vez. Faça um programa que leia da entrada um número  $n$  de degraus que há na escada, onde  $n > 0$ , e escreva na saída o número de maneiras diferentes que essa pessoa pode subir os  $n$  degraus dessa escada respeitando as restrições apresentadas. Implemente de forma que seja possível obter (em poucos segundos) respostas para valores de  $n$  suficientemente grandes, por volta de 1000.

**Teste:**

Entradas:	Saídas Correspondentes:
1000	Posso subir a escada de 1000 degraus de 7033036771142281582183525487718354977018 1269836358732742604905087154537118196933 5797422494945626117334877504492417659910 8818636326545022364710601205337412127386 7339111198139373125598767690091902245245 323403501 formas

**Questão 2 – (Valor 1.5 pontos)**

Utilizando subprogramação, faça um programa que leia da entrada padrão nomes completos de pessoas, até que um nome vazio seja lido. Escreva na saída padrão todos os nomes lidos com abreviações dos nomes intermediários. Onde cada nome intermediário fica com a primeira letra, seguido de ponto ".". Além disso, nomes intermediários tais como "e", "de", "da", "das", "do", e "dos" devem ser suprimidos.

**Teste:**

Entradas:	Saídas Correspondentes:
Edson Arantes do Nascimento Ayrton Senna do Brasil Juca de Oliveira Joaquim José da Silva Xavier Pedro de Alcântara de Bragança e Bourbon	Edson A. Nascimento Ayrton S. Brasil Juca Oliveira Joaquim J. S. Xavier Pedro A. B. Bourbon

**Questão 3 – (Valor 3.5 pontos)**

Faça um programa que leia da entrada padrão um alfabeto, cujas letras são distintas e separadas por espaço. Considere  $n$  como o número de letras do alfabeto. Escreva na saída todas as palavras que podem ser escritas por este alfabeto de tamanho 1 até  $n$ . Ou seja, você deve gerar todas as palavras de tamanho 1, todas as palavras de tamanho 2, e assim sucessivamente, até gerar todas as palavras de tamanho  $n$ .

**Teste:**

Entradas:	Saídas Correspondentes:
a b c	As palavras do alfabeto a b c são ['a', 'b', 'c', 'aa', 'ab', 'ac', 'ba', 'bb',

	'bc', 'ca', 'cb', 'cc', 'aaa', 'aab', 'aac', 'aba', 'abb', 'abc', 'aca', 'acb', 'acc', 'baa', 'bab', 'bac', 'bba', 'bbb', 'bbc', 'bca', 'bcb', 'bcc', 'caa', 'cab', 'cac', 'cba', 'cbb', 'cbc', 'cca', 'ccb', 'ccc']
--	---

#### Questão 4 – (Valor 3.5 pontos)

Utilizando subprogramação, faça um programa que processe pontos tridimensionais (x,y,z). Seu programa deve ler da entrada padrão linhas até que uma linha vazia seja digitada. Cada linha lida, exceto a vazia, deve conter três números de ponto flutuante, separados por espaços em branco. Em sequência, leia da entrada padrão um ponto de referência (xRef,yRef,zRef) que se pretende um deslocamento de todos os demais pontos já lidos. Liste os pontos lidos e respectiva soma de distâncias ao ponto de referência, conforme teste apresentado a seguir. Em seguida, leia um inteiro representando a quantidade de ciclos de deslocamento, bem como um número de ponto flutuante representando o diferencial de deslocamento de cada coordenada de cada ponto a sua respectiva coordenada no ponto de referência. Processe os pontos e após o deslocamento em cada ciclo pedido mostre as coordenadas de todos pontos e sua respectiva soma de distâncias ao ponto de referência.

#### Teste:

Entradas:	Saídas Correspondentes:
10 10.3 10 20 20.2 20 30 30.1 30 50 50.4 10 10 50.1 50 5 5 50 3 1.75	Pontos Originais: (10.00, 10.30, 10.00) (20.00, 20.20, 20.00) (30.00, 30.10, 30.00) (50.00, 50.40, 10.00) (10.00, 50.10, 50.00) Soma das Distâncias para o Ponto (5.00, 5.00, 50.00): 238.95  Listagem de Pontos no Ciclo 1, com Delta de Deslocamento 1.75: (8.25, 8.55, 11.75) (18.25, 18.45, 21.75) (28.25, 28.35, 31.75) (48.25, 48.65, 11.75) (8.25, 48.35, 50.00) Soma das Distâncias para o Ponto (5.00, 5.00, 50.00): 226.05  Listagem de Pontos no Ciclo 2, com Delta de Deslocamento 1.75: (6.50, 6.80, 13.50) (16.50, 16.70, 23.50) (26.50, 26.60, 33.50) (46.50, 46.90, 13.50) (6.50, 46.60, 50.00) Soma das Distâncias para o Ponto (5.00, 5.00, 50.00): 213.38  Listagem de Pontos no Ciclo 3, com Delta de Deslocamento 1.75: (4.75, 5.05, 15.25) (14.75, 14.95, 25.25) (24.75, 24.85, 35.25)

	(44.75, 45.15, 15.25) (4.75, 44.85, 50.00) Soma das Distâncias para o Ponto (5.00, 5.00, 50.00): 200.98
--	--

**Boa Avaliação!**