



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 2º 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)

Faça um programa, contendo pelo menos um subprograma, que leia inicialmente da entrada padrão as informações de um círculo, ou seja, as coordenadas inteiras, x_C e y_C , do seu centro e o seu raio, r_C , também inteiro. Todas estas três informações em uma única linha. As linhas seguintes deverão conter dois inteiros em cada linha, representando coordenadas de pontos, x_P e y_P , até que uma linha vazia seja digitada, delimitando o fim das entradas. Para cada ponto que estiver dentro da área do círculo, incluindo as bordas, seu programa deve escrever a mensagem: (x_P, y_P) está dentro do círculo (x_C, y_C, r_C) . Ao final, seu programa deve escrever o total de pontos lidos, o total de pontos dentro do círculo e o percentual de pontos dentro do círculo, com precisão de uma casa decimal. Veja o teste a seguir e siga rigidamente seu formato.

Definição: a distância entre dois pontos (x_A, y_A) e (x_B, y_B) é dada pela raiz quadrada da soma dos quadrados das diferenças $(x_B - x_A)$ e $(y_B - y_A)$.

Obs.: Para realizar esta questão, só se pode utilizar estruturas nativas de Python, sem bibliotecas de terceiros.

Teste:

Entradas:	Saídas Correspondentes:
10 50 20 10 10 0 50 100 40 40 60 10 100 20 45 15 150 30 -100 20 20	(0, 50) está dentro do círculo (10, 50, 20) (20, 45) está dentro do círculo (10, 50, 20) Quantidade de Pontos Processados: 9 Quantidade de Pontos Dentro de Círculo: 2 Percentual de Pontos Dentro: 22.2 %

Questão 2 – (Valor 1.5 pontos)

Faça um programa, contendo subprogramas, que leia inicialmente da entrada padrão a quantidade de linhas de um texto que deverá ser lido nas linhas seguintes, contendo zero ou mais palavras por linha. Identifique qual(is) a(s) palavra(s) mais frequente(s) e liste esta(s) palavra(s). Ao final, escreva a quantidade de ocorrência(s) desta(s) palavra(s). Converta todas as palavras lidas para letras maiúsculas. Caso nenhuma palavra tenha sido lida, escreva mensagem informando isto.

Obs.: Para realizar esta questão, só se pode utilizar estruturas nativas de Python, sem bibliotecas de terceiros.

Teste:

Entradas :	Saídas Correspondentes :
4 No meio do caminho tinha uma pedra Tinha uma pedra no meio do caminho Tinha uma pedra No meio do caminho tinha uma pedra	TINHA UMA PEDRA Ocorreu(ram) 4 vez(es)

Questão 3 – (Valor 2.0 pontos)

O número de plantas numa horta cresce de modo que a cada ano esse número é obtido pelo dobro do número que havia no ano anterior. Além disso, para nossa tristeza, morre a metade do número de plantas que havia há dois anos atrás. Faça um programa para cada item abaixo, que retorne a quantidade de plantas que há na horta no ano n , de modo recebemos na entrada um inteiro n , e dois valores do tipo float que sejam a quantidade de plantas que há nos dois primeiros anos. Considere a saída com duas casas decimais.

(a) Implemente usando recursividade.

(b) Implemente sem usar recursividade.

Obs.: Para realizar esta questão, só se pode utilizar estruturas nativas de Python, sem bibliotecas de terceiros.

Teste:

Entradas :	Saídas Correspondentes :
7 10.234 34	No ano 1 há 10.234 flores e no ano 2, 34.0. Assim, no ano 7 há 542.55 flores.

Questão 4 – (Valor 3.5 pontos)

Faça um programa, utilizando subprogramação, que receba uma sequência de inteiros distintos de entrada, e:

. Ordene a sequência, utilizando a seguinte estratégia recursiva: divida a sequência ao meio, ordene cada metade recursivamente, una cada metade comparando elementos de cada sequência ordenada e obtendo o menor dentre as duas sequências para a solução.

. Retorne a sequência ordenada, após isso, receba um inteiro x como entrada, e faça um programa que verifique se o elemento x pertence a sequência ordenada. Para tal, implemente o algoritmo de busca binária.

. Utilizando somente o algoritmo de busca binária feito anteriormente, caso o elemento x se encontre na lista, pergunte ao usuário se ele deseja remover da lista. Em caso afirmativo, remova-o, retornando a lista sem o elemento x . Em caso negativo, conclua o programa exibindo a sequência. Caso o elemento x não se encontre na lista, pergunte ao usuário se ele deseja inserir o elemento na lista. Em

caso afirmativo, insira o elemento, retornando a sequência com o elemento x. Em caso negativo, conclua o programa exibindo a sequência.

. No item anterior, caso a resposta da consulta sobre inserir (ou remover) um elemento seja diferente de S, s, N ou n, retorne a mensagem “Opção não encontrada” e execute novamente a consulta até receber respostas devidas.

Obs.: Ao utilizar métodos de lista, só é permitido o comando append(). Qualquer outro método é proibido.

Obs.: Para os algoritmos de ordenação e de busca, só serão consideradas as respostas que seguirem as estratégias pedidas.

Obs.: Para realizar esta questão, só se pode utilizar estruturas nativas de Python, sem bibliotecas de terceiros.

Teste:

Entradas:	Saídas Correspondentes:
4 5 2 19 29 28 17 s	A lista ordenada é: [2, 4, 5, 19, 28, 29] 17 O elemento 17 não consta na lista e, caso deseje, ele será inserido na posição 4 Você deseja inserir o elemento 17? Insira S (s) ou N (n) s [2, 4, 5, 17, 19, 28, 29]
4 5 2 19 29 28 20 n	A lista ordenada é: [2, 4, 5, 19, 28, 29] 20 O elemento 20 não consta na lista e, caso deseje, ele será inserido na posição 5 Você deseja inserir o elemento 20? Insira S (s) ou N (n) n [2, 4, 5, 19, 28, 29]
4 5 2 19 29 28 15 s	A lista ordenada é: [2, 4, 5, 19, 28, 29] 15 O elemento 15 não consta na lista e, caso deseje, ele será inserido na posição 4 Você deseja inserir o elemento 15? Insira S (s) ou N (n) s [2, 4, 5, 15, 19, 28, 29]
4 5 2 19 29 28 19 s	A lista ordenada é: [2, 4, 5, 19, 28, 29] 19 O elemento 19 está na posição 4 Você deseja remover o elemento 19? Insira S (s) ou N (n) s [2, 4, 5, 28, 29]
4 5 2 19 29 28 5 n	A lista ordenada é: [2, 4, 5, 19, 28, 29] 5 O elemento 5 está na posição 3 Você deseja remover o elemento 5? Insira S (s) ou N (n) n [2, 4, 5, 19, 28, 29]
4 5 2 19 29 28	A lista ordenada é: [2, 4, 5, 19, 28, 29]

4	4
a	O elemento 4 está na posição 2
m	Você deseja remover o elemento 4? Insira S (s) ou N (n)
n	a
	Opção não encontrada
	Você deseja remover o elemento 4? Insira S (s) ou N (n)
	m
	Opção não encontrada
	Você deseja remover o elemento 4? Insira S (s) ou N (n)
	n
	[2, 4, 5, 19, 28, 29]

Questão 5 – (Valor 1.5 pontos)

Faça um programa que receba uma string e execute os seguintes passos:

- . Verifique se a entrada x pertence ao alfabeto 0, 1.
- . Em caso negativo, retorne informando que a entrada x não está codificada em binário e conclua o programa.
- . Em caso positivo, faça a soma $x+1$, retornando cada passo dessa operação.

Obs.: Para realizar esta questão, só se pode utilizar estruturas nativas de Python, sem bibliotecas de terceiros.

Teste:

Entradas :	Saídas Correspondentes:
10101111	a soma de 10101111 por 1 é: 10101110 10101100 10101000 10100000 10110000
000000	a soma de 000000 por 1 é: 000001
10000	a soma de 10000 por 1 é: 10001
1111	a soma de 1111 por 1 é: 1110 1100 1000 0000 10000
01010101200	01010101200 não está codificado em binário

Boa Avaliação!