



Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina Fundamentos de Programação

AD1 – 2º semestre de 2020

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.
- 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 via atividade específica na Plataforma 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 (2,0 pontos)

Faça um programa que leia números inteiros da entrada padrão, até que um número menor ou igual a zero seja digitado, não sendo considerado para o computo da(s) saída(s). Caso o primeiro número lido seja menor ou igual a zero, escreva a mensagem “Nenhum número maior que zero foi lido!!!”. Caso contrário escreva:

- (1) Qual o maior número lido;
- (2) Caso algum número par tenha sido lido, quantos foram e qual a média dos números pares. Escrita com precisão de duas casas decimais;
- (3) Caso algum número ímpar tenha sido lido, quantos foram e qual a média dos números ímpares. Escrita com precisão de duas casas decimais;
- (4) Caso algum número primo tenha sido lido, quantos foram e qual a média dos números primos. Escrita com precisão de duas casas decimais.

Definição: Um número inteiro é primo se e somente se ele é maior que 1 e só é divisível por 1 e por ele mesmo.

Exemplo

Entrada	Saída
123 4 9 1890 5 -10	Quantidade de Pares: 2 com média: 947.00 Quantidade de Ímpares: 3 com média: 45.67 Quantidade de Primos: 1 com média: 5.00

2ª Questão (2,0 pontos)

Utilizando sub-programação, faça um programa que leia da entrada padrão uma única linha podendo conter zero ou mais números inteiros. Caso exista(m), escreva qual(is) o(s) número(s) par(es), Caso exista(m), escreva qual(is) o(s) número(s) primo(s). Mostre-o(s) um por linha. Caso a linha lida seja uma string vazia, escreva a mensagem: “Nenhum Número Foi Lido!!!”.

Definição: Um número inteiro é primo se e somente se ele é maior que 1 e só é divisível por 1 e por ele mesmo.

Exemplo

Entrada	Saída
100 15 13 8 4 2 1 -29 37 997 1024	Relação de Par(es): 100 8 4 2 1024 Fim da Relação de Par(es). Relação de Primo(s): 13 2 37 997 Fim da Relação de Primo(s).

3ª Questão (2,0 pontos)

Utilizando sub-programação, faça um programa que leia da entrada padrão strings, contendo linha(s) de número(s) de ponto flutuante de uma matriz bidimensional. Até que uma string vazia seja digitada. Escreva:

- (1) O conteúdo da matriz, onde cada linha a ser escrita possua apenas números de ponto flutuante, com duas casas decimais de precisão, sem vírgulas nem colchetes;
- (2) O menor e o maior número de cada linha;
- (3) O menor e o maior número de toda a matriz.

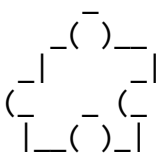
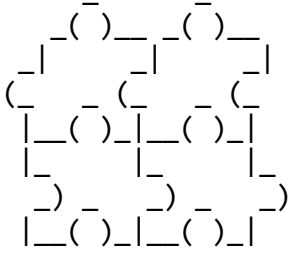
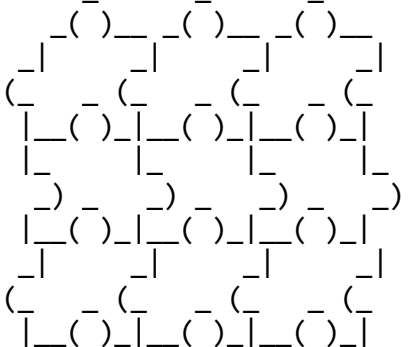
Exemplo

Entrada	Saída
3.44 100.88 12.5 77.66 1 2 3 4 87.65 121.13 9.55 44	Conteúdo da Matriz: 3.44 100.88 12.50 77.66 1.00 2.00 3.00 4.00 87.65 121.13 9.55 44.00 Menor e Maior da Linha 1: 3.44 e 100.88 Menor e Maior da Linha 2: 1.00 e 4.00 Menor e Maior da Linha 3: 9.55 e 121.13 Menor e Maior de Toda a Matriz: 1.00 e 121.13

4ª Questão (2,0 pontos)

Vamos descobrir o quão bom você é para reconhecer padrões simples e desenhá-los utilizando apenas caracteres?

Observe a associação entre os valores numéricos e os “desenhos” em cada padrão:

Padrão 1	Padrão 2	Padrão 3
1 #	1 #	1 
2 # ##	2 # # # #	2 
3 # ## ###	3 # # # # # #	3 

Nessa questão, seu programa deverá receber como entrada um número inteiro com valor entre 1 e 100, e desenhar as figuras correspondentes nos padrões 1, 2 e 3. O padrão não pode estar pré-armazenado no código-fonte do programa. Ou seja, ele deve ser desenhado “ao vivo”.

Entrada

A entrada é um valor inteiro N entre 1 e 100 informado via entrada padrão.

Saída

A impressão, na saída padrão, da figura correspondente no primeiro padrão, seguido por uma linha em branco, a impressão correspondente no segundo padrão, mais uma linha em branco e a impressão do terceiro padrão.

Exemplo

Entrada	Saída
3	<pre># ## ### # # # # # # # # _ () _ _ () _ _ () _ _ _ () _ _ () _ _ () _ (_ _ () _ _ () _ _ () _ _ () _ _ () _ _ () _ _) _ _) _ _) _ _) _ () _ _ () _ _ () _ _ _ _ _ _ _ _ _ _ (_ _ () _ _ () _ _ () _ _ () _ _ () _ _ () _ </pre>

Observações

Caso algum dos padrões esteja pré-armazenado no código fonte ou em arquivos extra então a nota atribuída para a questão será 0 (zero).

5ª Questão (2,0 pontos)

Essa questão envolve um jogo com barras de chocolate. O jogo começa quando eu lhe dou algumas barras de chocolate. Você pode então me devolver algumas delas, mas você precisa seguir algumas regras para isso. Nas regras descritas a seguir, N é a quantidade de barras que você tem:

Regra 1: Se N é par então você deve me devolver metade das barras.

Regra 2: Se N é divisível por 3 ou 4 sem gerar um valor fracionado então você deve multiplicar os dois últimos dígitos de N e me devolver essa quantidade de barras.

Regra 3: Se N é divisível por 5 sem gerar um valor fracionado então você deve me devolver exatamente 42 barras.

Nesse jogo, as regras são aplicadas toda vez que a quantidade N de barras mudar. O jogo encerra quando não for mais possível aplicar as regras ou quando você ficar com exatamente 42 barras, sendo esse o objetivo do jogo para declará-lo vencedor.

Seu programa deve incluir uma única função recursiva que recebe como entrada apenas a quantidade N e retorna `True` caso o resultado final seja 42 e `False` caso contrário. Após o retorno, o resultado deverá ser impresso na saída padrão conforme definido a seguir.

Entrada: O valor inteiro e não negativo N inicial, informado via entrada padrão.

Saída: Seu programa deverá imprimir “Venci” na saída padrão caso a primeira chamada da função recursiva retorna `True` ou “Perdi” caso contrário.

Exemplos

Entrada	Saída
41	Perdi

Entrada	Saída
42	Venci

Entrada	Saída
53	Perdi

Entrada	Saída
84	Venci

Entrada	Saída
250	Venci

Observações

Caso o jogo não seja implementado com recursão conforme a especificação então a nota atribuída para a questão será 0 (zero).