

Extrato

Exercícios de Programação

Funções

1. Elaborar um programa na linguagem Python que permita gerenciar a aprovação ou não de um grupo de 30 alunos, com base nas notas e na frequência. O programa deve possuir uma função para determinar a média do aluno, dada por $mf = 0,4.ap1 + 0,4.ap2 + 0,2.ac$. Deve possuir também uma função que permita, caso o aluno seja reprovado, substituir a menor nota das ap pela nota da as (se a nota da as for superior a menor ap) e determinar a nova média final. Deve possuir também uma função para determinar a porcentagem de faltas de cada aluno. O programa deverá ler o número de aulas dadas; ler as notas de ap, ap2 e ac; ler a nota de as (se for necessário); e ler o número de faltas de cada aluno. Deverá escrever a média final de cada aluno, a porcentagem de faltas e a condição ('aprovado' ou 'reprovado'). O aluno estará aprovado se a média final for superior ou igual a 7.0 e tiver uma frequência superior a 75%. Deverá fazer a as se obtiver uma nota mf inferior a 7.0.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/funcoesAprovacao.py

2. Elaborar um programa na linguagem Python que possua um conjunto de funções para calcular a área de figuras geométricas. O programa deve permitir que o usuário digite o número de figuras cujos parâmetros serão lidos e cuja área será determinada. Deve ler, para cada figura, o seu tipo, ('t':triângulo; 'r':retângulo; 'c':círculo) e os respectivos parâmetros para cálculo de área, escrevendo a área da figura solicitada.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/funcoesGeometria.py

3. Elaborar um programa na linguagem Python que leia um número inteiro. Se o número for par não negativo, o programa deve executar uma função para calcular o valor da soma dos termos positivos de uma série aritmética de razão 1, cujo maior termo é N dado, ou seja, $soma = 1 + 2 + 3 + \dots + N$. Se o número for ímpar não negativo, o programa deve executar uma função para determinar a soma dos seus divisores. Se o número for negativo, o programa deve calcular a soma dos quadrados do número até 0.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/funcoesSeriesN.py

4. Elaborar um programa na linguagem Python que leia os dados de uma pessoa: nome, idade, peso, altura. Para a idade lida, o programa deverá recomendar exercícios do tipo A, para pessoas acima de 60 anos; do tipo B, para pessoas entre 40 e 60 anos; do tipo C, para pessoas entre 15 e 40 anos; e do tipo D, para crianças (idade inferior a 15 anos). Com base no peso e na altura, o programa deve calcular o IMC (dado por $\text{peso}/\text{altura}^2$), determinando as dietas: 'magra', IMC superior a 25; 'normal', IMC entre 19 e 25; e 'gorda', IMC inferior a 19.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/funcoesImcExercicio.py

5. Elaborar um programa na linguagem Python que permita realizar várias operações diferentes com um par de números digitados pelo usuário, A e B. As operações são as seguintes:

- soma dos quadrados, designada 'sq', que resulta na soma do quadrado de A com o quadrado de B;
- soma dos cubos, designada 'sc', que resulta na soma de A elevado à terceira potência com B elevado à terceira potência;
- quadrado da diferença, 'qdif', que resulta no quadrado da diferença de A por B;
- quadrado da soma, 'qsoma', que resulta no quadrado da soma de A e B;
- cubo da diferença, 'cdif', resulta no cubo da diferença de A por B;
- cubo da soma, 'csoma', que resulta no cubo da soma de A e B.

O programa deve executar sucessivas vezes as operações. Na primeira repetição, o usuário deve digitar os valores de A e B, seguidos da operação desejada. Nas repetições seguintes, o usuário deve digitar o valor de um novo número, seguido da nova operação desejada. A nova operação será aplicada sobre o novo número e o resultado da repetição anterior. O programa será interrompido quando for digitada a operação 'nop' - nenhuma operação - e deverá escrever o último valor resultante da operação.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/funcoesNotacaoPolonesa.py

6. Uma forma de calcular a raiz quadrada de um número é utilizar um algoritmo recursivo que é alimentado por um valor inicial dado. A fórmula da recursão é dada por $sn = 1/2.[sn-1 + A/sn-1]$, onde sn é o valor da raiz na interação n e sn-1 é o valor na interação anterior. Elaborar uma função na linguagem Python que leia um número para o qual se deseja obter a raiz quadrada, o valor inicial da raiz do número dado e o número de interações. A função deverá retornar a raiz quadrada.

Listas

7. Elaborar um programa na linguagem Python que leia e armazene os dados de um grupo de 30 pessoas: nome, idade, peso, altura. Para a idade de cada pessoa lida, o programa deverá recomendar exercícios do tipo A, para pessoas acima de 60 anos; do tipo B, para pessoas entre 40 e 60 anos; do tipo C, para pessoas entre 15 e 40 anos; e do tipo D, para crianças (idade inferior a 15 anos). Com base no peso e na altura, o programa deve calcular o IMC (dado por $\text{peso}/\text{altura}^2$), determinando as dietas: 'magra', IMC superior a 25; 'normal', IMC entre 19 e 25; e 'gorda', IMC inferior a 19. O programa deverá armazenar a dieta de cada pessoa e o respectivo IMC.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/listaDieta.py

8. Elaborar um programa na linguagem Python que leia e armazene as matrículas e notas de um grupo de 20 alunos. O programa deverá calcular a média das notas lidas e o desvio padrão (usando a fórmula disponível em <<https://www.todamateria.com.br/desvio-padrao/>>). Deverá escrever somente as matrículas dos alunos com nota inferior à nota média.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/listaDevPad.py

9. Elaborar um programa na linguagem Python que leia uma sequência de vários números inteiros. O programa será interrompido quando for lido um número igual a 0. Para cada

número lido o programa deverá determinar: (i) determinar a soma da série $\{0,1,2, \dots, N\}$; (ii) determinar a soma da série $\{1/N, 2/N-1, \dots, N/1\}$. O programa deverá escrever todos os números superiores ao primeiro número lido. Deverá escrever também todos os números inferiores ao último número lido antes do 0.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/listaSerie.py

10. Elaborar um programa na linguagem Python para cadastrar os códigos de um grupo de clientes de uma loja e o valor das respectivas compras realizadas em um determinado mês. O código 0 significa que não será cadastrado novo cliente. Após os clientes serem cadastrados, o usuário poderá fazer uma consulta ao sistema, digitando um valor qualquer. O sistema deverá escrever os códigos dos clientes que realizaram compras acima desse valor.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/listaCliente.py

11. Elaborar um programa na linguagem Python para registrar os lançamentos contábeis de uma empresa ao longo de um ano. O programa deverá ler o CNPJ da empresa no início do ano juntamente com o saldo de caixa do mês anterior. Deverá cadastrar, mês a mês, a receita e a despesa. Ao final do ano, o programa deverá apresentar o extrato do balanço contábil, contendo o mês, receita, despesa, saldo parcial.

12. Elaborar um programa em Python que leia e cadastre em uma lista um conjunto inicial de N itens armazenados em uma despensa. O programa deve permitir que, após cadastrar os itens, o usuário: (i) inclua um novo item; (ii) remova um item pelo nome; (iii) apague todos os itens da lista. Sugestão: consulte o link <https://docs.python.org/pt-br/3/tutorial/datastructures.html?highlight=m%C3%A9todos%20lista>

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/listaDespensa.py

13. Elaborar um programa em Python que leia e cadastre os nomes dos alunos e as respectivas notas obtidas em uma avaliação de uma turma de aula com 20 alunos. O programa deverá, após os dados serem cadastrados, ler o nome de um aluno e escrever a nota obtida por ele. Sugestão: consulte o link <https://docs.python.org/pt-br/3/tutorial/datastructures.html?highlight=m%C3%A9todos%20lista>

Strings

14. Elaborar um script em Python que permita ler uma palavra digitada pelo usuário. O programa deverá escrever todas as letras da palavra diferentes da letra 'a'.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/StringExercicio1.py

15. Elaborar um script em Python que permita ler um texto digitado pelo usuário. O programa deverá calcular o número de vogais presentes no texto digitado.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/StringExercicio2.py

ou

https://github.com/claytonjasilva/prog_exemplos/blob/main/StringExercicio2-1.py

16. Elaborar um script em Python que leia uma palavra digitada pelo usuário. O programa deverá escrever a palavra digitada com a posição das letras invertidas.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/StringExercicio3.py

ou

https://github.com/claytonjasilva/prog_exemplos/blob/main/StringExercicio3-1.py

17. Elaborar um script em Python que leia o nome completo de uma pessoa e a função em uma empresa. O programa escreve o endereço de e-mail formado da seguinte maneira: primeiro nome + ponto + função + arroba + 'empresa.com.br'. O endereço de e-mail deve ser formado por letras minúsculas.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/StringExercicio4.py

18. Elaborar um script em Python que leia uma relação de N nomes de alunos digitada pelo usuário. Após cadastrar a relação o usuário poderá digitar uma letra e o programa retornará todos os nomes iniciados com a letra digitada.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/StringExercicio5.py

19. Elaborar um script em Python que leia um texto digitado pelo usuário. Após digitar o texto, o usuário poderá digitar uma palavra e o programa determinará o número de vezes que a palavra está presente no texto digitado.

Sugestão: ver < [Tipos embutidos — documentação Python 3.10.4](#) >

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/StringExercicio6.py

20. Elaborar um script em Python que leia um texto digitado pelo usuário. O texto poderá ter no máximo 50 caracteres. Enquanto o texto não for válido o programa deverá solicitar que o usuário digite um texto dentro do limite. Caso o usuário tente três vezes o programa interromperá a execução, informando que o texto não foi cadastrado.

Tuplas

21. Elaborar um programa na linguagem Python que armazene as vogais em uma tupla. Leia também uma string, verifique e escreva em uma lista quantas vezes cada vogal está presente na string lida.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/tuplaExemplo1.py

22. Elaborar um programa em Python que leia a matrícula de um grupo de N alunos. Como a matrícula identifica de modo inequívoco cada aluno, após cadastrar a matrícula dos alunos, o programa não admitirá mudar o número os dados cadastrados.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/tuplaExemplo2.py

23. Elaborar um programa em Python que leia a matrícula de um grupo de N alunos. Leia também as notas do grupo (AP1, AP2 e AC). O programa deverá escrever a matrícula e a média de cada aluno.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/tuplaExemplo3.py

24. Elaborar um programa em Python que leia e armazene o cpf e o saldo bancário de um grupo de 10 pessoas ao longo de 30 dias. O programa deverá escrever o cpf das pessoas seguido do saldo médio mensal.

Solução em

https://github.com/claytonjasilva/prog_exemplos/blob/main/tuplaExemplo4.py

25. Elaborar um programa na linguagem Python que leia o nome e o cpf de um grupo de 30 pessoas. O programa deve fornecer a possibilidade de 'cadastrar a habilitação' para cada cpf; ou 'alterar a data da habilitação' se já tiver sido cadastrada. Quando cadastrar a habilitação, o usuário deve informar o número e a data. O número não pode ser alterado, no entanto a data pode ser alterada.

26. Elaborar um programa na linguagem Python que leia a identificação de um grupo de componentes mecânicos. O usuário deixará de cadastrar os componentes quando for inserido o número de identificação for igual a 0. O identificador do componente, que não pode ser alterado, sempre é cadastrado juntamente com sua versão de fabricação, que pode ser alterada. Após cadastrados os componentes, o programa deve permitir localizar um componente a partir de seu identificador, apresentando a sua versão de fabricação.