
Lista de Exercícios – Estruturas de Controle de Repetição

As questões a seguir envolvem o uso das estruturas de decisão while e for (explicadas em sala) juntamente com problemas simples e do dia-a-dia, para os quais será necessário o desenvolvimento de algoritmos em Python. Em cada caso, procure antes resolver de fato o problema para só então começar a escrever código, lembrando sempre que, para cada problema, existem mais de uma solução aplicável.

1. Escreva um programa em Python que receba do aluno: nome, matrícula e as notas referente as três unidades e retorne a média dessas notas, atente para o fato de as notas estarem dentro os limites convencionais ($\text{nota} > 0$ & $\text{nota} < 10$).
2. Modifique o programa anterior para que além de retornar a média, informa a situação do aluno quando a estar: reprovado, aprovado ou em recuperação.
Convenção da média: 1 – 3: reprovado. 4 – 6: recuperação. 7 – 10: aprovado.
3. Escreva um programa em Python que receba como entrada um número n e retorne sua tabuada.
Ex.: $n = 5$
 $5 * 1 = 5$
 $5 * 2 = 10$
...
 $5 * 10 = 50$
4. Modifique o programa anterior para que exiba somente a partir de onde um usuário quiser.
Ex.: $n = 5$ & $p = 5$
 $5 * 5 = 25$
 $5 * 6 = 30$
...
 $5 * 10 = 50$
5. Escreva um programa em Python que receba um determinado número n e verifique se o mesmo é primo ou não, caso seja, imprima, caso não, peça outro número e refaça o teste.
6. Escreva um programa em Python que calcule o fatorial de um número n e exiba, atente para o fato de números negativos não tem fatorial, só permita a fatoraçaõ caso o número indicado seja positivo.
7. Escreva um programa em Python que escreva na tela a série Fibonacci até um determinado n . **Observação:** A série Fibonacci consiste em uma sucessão de números que aparece em muitos fenômenos da natureza. Descrita por Leonardo Fibonacci, ela é infinita e começa

com 0 e 1. Os números seguintes são sempre a soma dos dois números anteriores. Portanto, depois de 0 e 1, vem 1 (0+1), 2 (1+1), 3 (2+1), 5 (3+2) ...

8. Escreva um programa em Python que receba um tal n e exiba o somatório dos próximos números digitados, até que o 0 (zero) seja digitado, encerrando o programa e ao final exibindo o somatório de todos os números dados como entrada em tempo de execução.
9. Escreva um programa em Python onde o usuário tente adivinhar o número gerado randomicamente, dentro de um intervalo escolhido pelo usuário e que só finalize quando o usuário acertar o número gerado, contabilizando o número de tentativas.
10. Seu Manoel tem uma loja de artigos de R\$ 1,69 e gostaria de montar uma tabela que contenha os preços das quantidades de 1 a 50, que é o número máximo de itens que um cliente pode pegar, mas como ele é preguiçoso, resolveu pagar a você, jovem programador, para fazer essa tabela.
Modelo da tabela: Loja do Seu Manoel
1 – R\$ 1,69
2 – R\$ 3,38
...
50 – R\$ 84,50
11. Lembra da questão 9? Então, desenvolva um algoritmo em Python que receba como parâmetro um determinado valor n e que execute o seguinte bloco de instruções n vezes. Script: O algoritmo deve sortear um número aleatório e guarda-lo em uma variável, o usuário terá direito a (n) tentativas de acertar o número sorteado. O algoritmo de imprimir uma mensagem de sucesso ou derrota. O zero dever condição de parada em caso de desistência.
12. Em 2010 João começou a trabalhar na Pilli Tecnologia, empresa de desenvolvimento de software de sua cidade. Quando ele chegou, o percentual de aumento salarial era de 1,5% ao ano. Preveja quanto está o salário de João, já em 2018, levando em consideração que o percentual de aumento salarial dobra a cada ano. Escreva um programa em Python que imprima o valor do salário, o percentual do aumento e o valor acrescido ao salário de João a cada ano.
13. Modifique o programa anterior para que seja dado como entrada: o ano inicial e final, o percentual inicial e o salário inicial do funcionário.
14. Escreva um algoritmo em Python que retorne o MDC de dois números.

-
15. Escreva um algoritmo em Python que retorne o MMC de dois números.
 16. Implemente uma urna eletrônica com 5 candidatos e simule uma eleição com os mesmos, ao final da execução, será necessário exibir a média de votação de cada candidato juntamente no número rodadas de votação que ocorreram durante a execução, considere 0 (zero) para finalizar o algoritmo.
 17. Escreva um algoritmo em Python que receba um número n e exiba todos os números ímpares e depois todos os números pares até o n , considerando que o 0 não é par e nem ímpar.
 18. Escreva um programa em Python que simule um cronômetro regressivo onde a entrada seja as horas.
Observação: Para ajudar na fidelidade (contagem em segundos) do algoritmo, importe a biblioteca `time`, para o uso do método `sleep`, que recebe um inteiro com parâmetro, contando esse parâmetro como segundo.
 19. Na matemática, existe o conceito de número perfeito, que diz o seguinte: “tendo um n qualquer, diz-se que ele é perfeito se o seu dobro for igual a soma de todos os seus divisores.” Escreva um algoritmo em Python que indique se um número capturado do teclado é perfeito ou não. Exemplo: 6 -> Divisores (1, 2, 3, 6) Soma dos divisores (1+2+3+6=12) Dobro de 6 (12). 6 é um número perfeito, segundo a afirmação.
 20. Modifique o programa anterior para que seja dado pelo usuário um intervalo, sendo exibidos todos os números perfeitos dentro do mesmo.
 21. Escreva um algoritmo em Python que simule a função “***” do interpretador, ou seja, dados dois números n e k retorne para o usuário, nk .
 22. Escreva um programa em Python que receba uma palavra do teclado e depois imprima suas letras.
Dica: Ao receber uma palavra no “`input()`”, a variável é tratada como um vetor de caracteres, para acessar uma posição do vetor, se escreve “vetor[posição]” será necessária uma função que capture o tamanho do vetor ou uma condição de parada para um laço de repetição.
 23. Em programação de uma maneira geral, é comum medir quanto tempo demora uma determinada função, atividade ou mesmo um trecho de código, escolha pelo menos dois algoritmos já implementados (ou outros dois de sua escolha), conte quanto tempo cada um demorou para executar sua função e imprima esse tempo na tela.
Observação: Para ajudar na fidelidade (contagem em segundos) do algoritmo, importe a biblioteca `time`.

-
24. Escreva um programa em Python que receba dois números e exiba um menu de opções que represente as 4 operações básicas, e deixe o usuário escolher o que fazer, após o resultado da operação, o menu deve reaparecer para que o usuário escolha uma nova ação, detalhe, a opção de atualizar os números e sair do programa também devem estar disponíveis.
 25. Escreva um programa em Python que receba dois valores que representem um intervalo de n até k onde $(n < k)$ e receba também um p que represente um passo, o programa deve exibir os valores de n até k saltando de p em p .
 26. Escreva um algoritmo em Python em que o usuário fique digitando número inteiros positivos até que o 0 seja indicado, resultando no fim da execução, ao final o algoritmo ainda deve dizer ao usuário quantos números pares e ímpares ele digitou durante a execução.
 27. Escreva um programa em Python onde seja solicitado ao usuário o sexo M ou F, respectivamente, masculino e feminino, e continue a pedir caso o sexo passado não seja válido.
 28. Relembrando a questão 26, escreva um programa em Python onde o usuário continue a passar número inteiros positivos até que o 999 seja digitado, finalizando a aplicação, após isso, o programa deve retornar ao usuário, o maior e menor número digitado bem como a média de todos os números digitados.
 29. Escreva um programa em Python que escreva os n primeiro número de uma PA (progressão aritmética) dados: o primeiro termo, a razão e quantidade n de termos.