

LÓGICA DE PROGRAMAÇÃO

LISTA DE EXERCÍCIOS 2

ESTES EXERCÍCIOS SERÃO POSTADOS NO SAVA E CONTAM PONTOS PARA APROVAÇÃO. ESTA LISTA VALE 4 PONTOS.

1. Escreva um programa Python que peça ao usuário para informar um número inteiro n positivo e simule n lançamentos de um dado não viciado. Ao final dos n lançamentos deve ser apresentado um resultado como o seguinte:

Face 1: 16667094 vezes (16.67%)
Face 2: 16667358 vezes (16.67%)
Face 3: 16667390 vezes (16.67%)
Face 4: 16670879 vezes (16.67%)
Face 5: 16661376 vezes (16.66%)
Face 6: 16665903 vezes (16.67%)

Execute este programa para mil, 1 milão e 100 milhões de lançamentos.

Dica: você vai precisar de import random. E de face = random.randint(1, 6)

2. A sequência de Fibonacci é uma progressão numérica em que cada termo é a soma dos dois anteriores: 1, 1, 2, 3, 5, 8, 13, 21, 34, ... Matematicamente, é definida por: F(1) = 1, F(2) = 1, e F(n) = F(n-1) + F(n-2) para n > 2.

Na natureza, a sequência aparece em diversos contextos, como na disposição das pétalas de flores, no padrão de sementes de girassóis, nas conchas do tipo nautilus e na ramificação de galhos e folhas — fenômenos associados à eficiência do crescimento e à proporção áurea.

Em estudos sociais, a sequência de Fibonacci tem sido usada para modelar crescimento populacional, difusão de informações e dinâmicas de redes sociais, já que descreve processos de expansão cumulativa e auto-organização semelhantes aos observados em interações humanas e comportamentos coletivos.

Faça um programa que gere a série de valores da sequência de Fibonacci até que o valor seja maior que 500.

3. A sequência de Collatz é criada a partir de uma regra simples:

- Se o número é par, divida-o por 2.
- Se o número é ímpar, multiplique-o por 3 e some 1.
- Repita esse processo até chegar ao número 1.

Por exemplo, começando com n = 6 teremos a sequência 6 -> 3 -> 10 -> 5 -> 16 -> 8 -> 4 -> 2 -> 1

Etapa	Valor de n	Par/ímpar	Regra aplicada	Resultado
1 (valor inicial)	6	Par	$6 \div 2$	3

Etapa	Valor de n	Par/Ímpar	Regra aplicada	Resultado
2	3	Ímpar	$3 \times 3 + 1$	10
3	10	Par	$10 \div 2$	5
4	5	Ímpar	$5 \times 3 + 1$	16
5	16	Par	$16 \div 2$	8
6	8	Par	$8 \div 2$	4
7	4	Par	$4 \div 2$	2
8	2	Par	$2 \div 2$	1

Embora a regra seja simples, ninguém conseguiu provar matematicamente que todos os números inteiros positivos chegam a 1 — isso é conhecido como a Conjectura de Collatz, um dos enigmas mais curiosos da matemática moderna.

Na programação, essa sequência é útil para exercitar laços de repetição, condicionais e tratamento de números inteiros.

Escreva um programa em Python que peça ao usuário para digitar um número inteiro positivo n maior que 1 e exiba a sequência de Collatz. Conte e mostre quantos passos foram necessários para chegar até 1.

4. Os números triangulares representam quantidades que podem ser dispostas em forma de triângulo. Eles são obtidos somando os números naturais sucessivos:

$$\begin{aligned} T_1 &= 1 = 1 \\ T_2 &= 3 = 1 + 2 \\ T_3 &= 6 = 1 + 2 + 3 \\ T_4 &= 10 = 1 + 2 + 3 + 4 \\ T_5 &= 15 = 1 + 2 + 3 + 4 + 5 \end{aligned}$$

Esses números aparecem em contextos como arranjos de objetos (bolas, pinos de boliche, pontos luminosos), organização de grupos (por exemplo, número de apertos de mão possíveis entre pessoas), padrões geométricos e séries aritméticas.

Escreva um programa Python que solicite ao usuário um número inteiro positivo n e calcule e exiba os n primeiros números triangulares, um por linha.

5. O fatorial de um número inteiro positivo n , representado por $n!$, é o produto de todos os inteiros positivos de 1 até n .

Em outras palavras:

$$\begin{aligned} n! &= n \times (n-1) \times (n-2) \times \dots \times 2 \times 1 \\ n! &= n \times (n-1)! \end{aligned}$$

Por convenção, definimos que $0!=1$ e $0!=1$.

Escreva um programa Python que solicite ao usuário um número inteiro positivo n maior que 20 e calcule $n!$.

6. Escreva um programa Python para verificar a nota do aluno em uma prova com 10 questões objetivas. O programa deve perguntar ao usuário um aluno e a resposta para cada questão da prova. Considere o seguinte gabarito.

Questão 1 – A

Questão 2 – B

Questão 3 – C

Questão 4 – D

Questão 5 – E

Questão 6 – E

Questão 7 – D

Questão 8 – C

Questão 9 – B

Questão 10 – A

7. Em uma competição de salto em distância cada atleta tem direito a cinco saltos. No final da série de saltos de cada atleta, o melhor e o pior resultados são eliminados. O seu resultado final é dado pela média dos três valores restantes.]

Escreva um programa Python receba o nome e as cinco distâncias alcançadas pelo atleta em seus saltos apresente os resultados conforme mostrado abaixo:

Atleta: Sandra Helena

Primeiro Salto: 6.1 m

Segundo Salto: 6.1 m

Terceiro Salto: 6.4 m

Quarto Salto : 6.0 m

Quinto Salto : 5.9 m

Melhor salto: 6.4 m

Pior salto : 5.9 m

Média considerada = (6.1 + 6.1 + 6.0) / 3 = 6,067 m

8. A média (ou média aritmética) é a soma de todos os valores dividida pelo número de elementos. Exemplo: Para os valores [5, 7, 9], a média é $(5 + 7 + 9) / 3 = 7$.

A **mediana** é o valor central quando os dados estão **ordenados**. Se houver n ímpar elementos, é o valor do meio. Se houver n par, é a média dos dois valores centrais. Por exemplo, para [3, 7, 9], a mediana é 7; para [2, 4, 8, 10], a mediana é $(4 + 8) / 2 = 6$.

A **moda** é o valor que mais se repete em um conjunto de dados. É, portanto, a medida de tendência central baseada na frequência.

Exemplo: Conjunto: [2, 4, 4, 5, 7, 7, 7, 8] » O número que mais aparece é 7. Logo, a moda = 7. Uma amostra pode ser unimodal, bimodal, multimodal ou amodal.

Tipo	Descrição	Exemplo	Moda(s)
Unimodal (modal)	Um valor mais frequente	[3, 4, 4, 5, 6]	4
Bimodal	Dois valores mais frequentes	[2, 2, 4, 4, 5]	2 e 4

Descrição	Exemplo	Moda(s)
Multimodal	[1, 1, 2, 2, 3, 3]	1, 2, 3
Amodal	[1, 2, 3, 4, 5]	—

O **desvio padrão** mostra o quanto os valores se afastam da média. Quanto maior o desvio, mais espalhados estão os dados.

$$\sigma = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$$

Escreva um programa Python que leia N valores numéricos e calcule a média, a mediana, a moda e o desvio-padrão.

9. O cardápio de uma Lanchonete do Manuel é o seguinte:

Especificação	Código	Preço
Cachorro Quente	100	R\$ 11,20
Bauru Simples	101	R\$ 11,00
Bauru com ovo	102	R\$ 14,00
Hambúrguer	103	R\$ 12,90
Cheeseburguer	104	R\$ 13,90
Refrigerante	105	R\$ 10,00

Escreva um programa Python que imprima os valores correspondentes aos pedidos de um cliente, conforme o exemplo abaixo:

```
+-----+
|                               RESUMO DA CONTA
+-----+-----+-----+-----+
| Especificação | Código | Preço Unitário(R$) | Quantidade | Total (R$) |
+-----+-----+-----+-----+
| Cheeseburger | 104    | 13,90             | 2          | 27.80      |
| Refrigerante | 105    | 10,00             | 2          | 20.00      |
+-----+-----+-----+-----+
| Total Geral:                                47.80 |
+-----+-----+-----+-----+
```

10. O algoritmo de validação do CPF calcula o primeiro dígito verificador a partir dos 9 primeiros dígitos do CPF, e em seguida, calcula o segundo dígito verificador a partir dos 9 (nove) primeiros dígitos do CPF, mais o primeiro dígito, obtido na primeira parte.

Tomes como exemplo o CPF fictício : 111.444.777-05.

a - Cálculo do primeiro dígito

O primeiro passo é calcular o primeiro dígito verificador, e para isso, separamos os primeiros 9 dígitos do CPF (111.444.777) e multiplicamos cada um dos números, da

direita para a esquerda por números crescentes a partir do número 2, como no exemplo abaixo:

1	1	1	4	4	4	7	7	7
10	9	8	7	6	5	4	3	2
10	9	8	28	24	20	28	21	14

Multiplicamos cada dígito do CPF pelo respectivo número e somamos cada um dos resultados : $10+9+8+28+24+20+28+21+14 = 162$

Pegamos o resultado obtido 162 e dividimos por 11. Consideraremos como quociente apenas o valor inteiro.

$$162 / 11 = 14 \text{ com resto } 8$$

- Se o resto da divisão for menor que 2, então o dígito é igual a 0 (Zero).
- Se o resto da divisão for maior ou igual a 2, então o dígito verificador é igual a 11 menos o resto da divisão ($11 - \text{resto}$).

No nosso exemplo temos que o resto é 8 então faremos $11-8 = 3$

Logo o primeiro dígito verificador é 3. Então podemos escrever o CPF com os dois dígitos calculados : 111.444.777-3X

b - Cálculo do segundo dígito

Para calcular o segundo dígito vamos usar o primeiro dígito já calculado. Vamos montar a mesma tabela de multiplicação usada no cálculo do primeiro dígito. Só que desta vez usaremos na segunda linha os valores 11,10,9,8,7,6,5,4,3,2 já que estamos incluindo mais um dígito no cálculo(o primeiro dígito calculado):

1	1	1	4	4	4	7	7	7	3
11	10	9	8	7	6	5	4	3	2
11	10	9	32	28	24	35	28	21	6

Novamente, efetuamos somamos o resultado da multiplicação : $11 + 10 + 9 + 32 + 28 + 24 + 35 + 28 + 21 + 6 = 204$

Dividimos o total do somatório por 11 e consideramos o resto da divisão.

$$204 / 11 = 18 \text{ e resto } 6$$

Após obter o resto da divisão, precisamos aplicar a mesma regra que utilizamos para obter o primeiro dígito:

- Se o resto da divisão for menor que 2, então o dígito é igual a 0 (Zero).
- Se o resto da divisão for maior ou igual a 2, então o dígito é igual a 11 menos o resto da divisão ($11 - \text{resto}$).

$11-6= 5$ logo 5 é o nosso segundo dígito verificador.

Logo o nosso CPF fictício será igual a : 111.444.777-35.

Escreva um algoritmo para validar um CPF.

