

ex001 – Soma dos números pares

Explicação:

- I. **while True::**
 - a. **O que é?:** Loop infinito.
 - b. **Pra que serve?:** Faz o programa continuar rodando até que uma condição específica seja atendida.
 - c. **Como funciona?:** O programa fica repetindo o bloco de código dentro do while True até que você use break para sair do loop.
- II. **n = int(input("Digite um número entre 1 e 150: ")):**
 - a. **O que é?:** Entrada de dados.
 - b. **Pra que serve?:** Pede ao usuário para digitar um número.
 - c. **Como funciona?:** O input() exibe uma mensagem para o usuário digitar algo. O int() converte a entrada em um número inteiro.
- III. **if n > 150::**
 - a. **O que é?:** Condicional.
 - b. **Pra que serve?:** Verifica se o número é maior que 150.
 - c. **Como funciona?:** Se o número for maior que 150, o código dentro do if é executado.
- IV. **elif n < 1::**
 - a. **O que é?:** Condicional alternativa.
 - b. **Pra que serve?:** Verifica se o número é menor que 1.
 - c. **Como funciona?:** Se o número for menor que 1, o código dentro do elif é executado.
- V. **else::**
 - a. **O que é?:** Condicional.
 - b. **Pra que serve?:** Executa o código se nenhuma das condições anteriores for verdadeira.
 - c. **Como funciona?:** Se o número estiver entre 1 e 150, o código dentro do else é executado.
- VI. **break:**
 - a. **O que é?:** Comando de saída.
 - b. **Pra que serve?:** Sai do loop while True.
 - c. **Como funciona?:** Quando o break é executado, o programa sai do loop e continua a execução do código fora do loop.
- VII. **soma = 0:**
 - a. **O que é?:** Atribuição.
 - b. **Pra que serve?:** Inicializa a variável soma com 0.

- c. **Como funciona?**: Cria uma variável chamada soma e define seu valor inicial como 0.

VIII. **numeros_pares = []:**

- a. **O que é?**: Lista vazia.
- b. **Pra que serve?**: Armazena os números pares encontrados.
- c. **Como funciona?**: Cria uma lista chamada numeros_pares que começa vazia.

IX. **for i in range(2, n+1, 2):**

- a. **O que é?**: Loop.
- b. **Pra que serve?**: Itera sobre os números pares de 2 até n.
- c. **Como funciona?**: O range(2, n+1, 2) gera uma sequência de números pares de 2 até n, e o for percorre essa sequência.

X. **soma += i:**

- a. **O que é?**: Atribuição com adição.
- b. **Pra que serve?**: Adiciona o valor de i à variável soma.
- c. **Como funciona?**: O += é uma forma curta de escrever soma = soma + i.

XI. **numeros_pares.append(i):**

- a. **O que é?**: Método de lista.
- b. **Pra que serve?**: Adiciona o número i à lista numeros_pares.
- c. **Como funciona?**: O append(i) adiciona o valor de i ao final da lista numeros_pares.

XII. **if n % 2 == 0:**

- a. **O que é?**: Condicional.
- b. **Pra que serve?**: Verifica se n é par.
- c. **Como funciona?**: O $n \% 2 == 0$ verifica se o resto da divisão de n por 2 é 0, o que indica que n é par.

XIII. **print(f'Os números pares até {n} são: {numeros_pares}')**

- a. **O que é?**: Comando de saída.
- b. **Pra que serve?**: Mostra os números pares encontrados.
- c. **Como funciona?**: O print exibe a mensagem com os números pares armazenados na lista numeros_pares.

XIV. **print(f'A soma dos numeros pares até {n} é {soma}')**

- a. **O que é?**: Comando de saída.
- b. **Pra que serve?**: Mostra a soma dos números pares.
- c. **Como funciona?**: O print exibe a mensagem com a soma dos números pares.

Explicação:

- I. **print("Digite 'sair' a qualquer momento para encerrar o programa.\n"):**
 - a. **O que é:** Essa linha faz o computador escrever uma mensagem na tela.
 - b. **Como funciona:** Quando o programa começa, ele mostra a mensagem para informar ao usuário que ele pode digitar "sair" para parar o programa.
- II. **while True::**
 - a. **O que é:** Isso cria um loop infinito, ou seja, o código dentro desse loop vai se repetir para sempre até que o programa seja parado.
 - b. **Como funciona:** O while True faz com que o código dentro dele seja executado repetidamente. O True significa "verdadeiro", então o loop continua rodando enquanto a condição for verdadeira.
- III. **palavra = input("Digite uma palavra: "):**
 - a. **O que é:** Essa linha pede ao usuário para digitar algo e guarda o que foi digitado em uma "caixinha" chamada palavra.
 - b. **Como funciona:** O computador mostra a mensagem e espera que o usuário digite algo. O que o usuário digitar será armazenado na variável palavra.
- IV. **if palavra.lower() == 'sair'::**
 - a. **O que é:** Isso verifica se o que o usuário digitou é igual a "sair" (depois de converter tudo para minúsculas).
 - b. **Como funciona:** O palavra.lower() converte a palavra digitada para minúsculas para evitar problemas com maiúsculas/minúsculas. Se o usuário digitar "Sair" ou "SAIR", o programa vai entender que ele quer sair.
- V. **break:**
 - a. **O que é:** Isso faz o programa sair do loop infinito.
 - b. **Como funciona:** Quando o usuário digita "sair", o break faz com que o programa pare de repetir o código dentro do while True e encerra o programa.
- VI. **if palavra == palavra[::-1]::**
 - a. **O que é:** Isso verifica se a palavra é igual à sua versão invertida.
 - b. **Como funciona:** O palavra[::-1] inverte a palavra. Se a palavra for igual à sua versão invertida, o programa vai dizer que é um palíndromo.

Explicação:

I. try: e except:

- O que é?** São como "tentativa" e "pegue o erro" em inglês.
- Para que serve?** Servem para tentar fazer algo e, se der errado, pegar o erro e fazer outra coisa em vez de deixar o programa quebrar.
- Como funciona?** No código, o try: tenta transformar o que você digitou em um número inteiro (int(input(...))). Se você digitar algo que não é um número (como "abc"), o except ValueError: pega esse erro e mostra uma mensagem de erro.

II. int()

- O que é?** É uma função que transforma algo em um número inteiro.
- Para que serve?** Serve para garantir que o que você digitou é um número, e não uma palavra ou outra coisa.
- Como funciona?** Quando você digita algo, o int() tenta transformar isso em um número. Se não conseguir, dá um erro.

III. input()

- O que é?** É uma função que permite que você digite algo no computador.
- Para que serve?** Serve para que o programa possa receber informações do usuário.
- Como funciona?** Quando o programa chega na linha input("Digite um número: "), ele espera você digitar algo e pressionar "Enter". O que você digitou é então usado no programa.

IV. for i in range(1, 11):

- O que é?** É um "para cada" em inglês.
- Para que serve?** Serve para repetir algo várias vezes, mudando um pouco a cada vez.
- Como funciona?** No código, range(1, 11) cria uma lista de números de 1 a 10. O for faz o programa repetir o que está dentro do for para cada número dessa lista.

V. print()

- O que é?** É uma função que mostra algo na tela.
- Para que serve?** Serve para que o programa possa mostrar informações para o usuário.
- Como funciona?** Quando o programa chega na linha print(...), ele mostra o que está dentro dos parênteses na tela.

VI. exit()

- O que é?** É uma função que faz o programa parar.
- Para que serve?** Serve para encerrar o programa se algo der errado.
- Como funciona?** Se o programa entrar no except ValueError:, ele mostra a mensagem de erro e depois usa exit() para parar o programa

Explicação:**I. texto = input("Digite uma palavra: ")**

- O que é?:** Esta linha de código é uma função que solicita ao usuário que insira uma string (uma sequência de caracteres, como palavras ou frases).
- Pra que serve?:** Serve para capturar a entrada do usuário e armazená-la na variável texto.
- Como funciona?:** Quando o programa é executado, ele exibe a mensagem "Digite uma palavra: " na tela e espera que o usuário digite algo. O que o usuário digitar será armazenado na variável texto

II. vogais = "aeiouAEIOU"

- O que é?:** Esta linha de código cria uma string que contém todas as vogais, tanto minúsculas quanto maiúsculas.
- Pra que serve?:** Serve para definir quais caracteres serão considerados vogais no código.
- Como funciona?:** A string "aeiouAEIOU" é simplesmente uma lista de caracteres que o programa usará para verificar se um caractere é uma vogal ou não.

III. count = 0

- O que é?:** Esta linha de código cria uma variável chamada count e a inicializa com o valor 0.
- Pra que serve?:** Serve para contar quantas vogais existem na string digitada pelo usuário.
- Como funciona?:** A variável count começa com 0 e será incrementada cada vez que o programa encontrar uma vogal na string.

IV. for char in texto:

- O que é?:** Esta linha de código inicia um loop que percorre cada caractere da string texto.
- Pra que serve?:** Serve para analisar cada caractere da string, um por um.
- Como funciona?:** O loop começa no primeiro caractere da string e vai até o último. A cada iteração, o caractere atual é armazenado na variável char.

V. if char in vogais:

- O que é?:** Esta linha de código é uma condição que verifica se o caractere atual (char) está presente na string vogais.
- Pra que serve?:** Serve para determinar se o caractere atual é uma vogal.
- Como funciona?:** Se o caractere atual (char) estiver na lista de vogais (vogais), a condição é verdadeira e o código dentro do if será executado.

VI. count += 1

- O que é?:** Esta linha de código incrementa o valor da variável count em 1.
- Pra que serve?:** Serve para contar quantas vezes uma vogal foi encontrada na string.

- c. **Como funciona?:** Cada vez que o programa encontra uma vogal, ele adiciona 1 ao valor de count.

VII. `print(f"O que foi digitado, contém {count} vogais.")`

- a. **O que é?:** Esta linha de código exibe uma mensagem na tela.
- b. **Pra que serve?:** Serve para mostrar ao usuário quantas vogais foram encontradas na string que ele digitou.
- c. **Como funciona?:** A mensagem "A string contém {count} vogais." é exibida, onde {count} é substituído pelo valor atual da variável count.

ex005- Fatorial

O que é Fatorial?

O fatorial de um número é o produto de todos os números inteiros positivos menores ou iguais a ele. Por exemplo, o fatorial de 5 (escrito como 5!) é $5 * 4 * 3 * 2 * 1 = 120$. Esse código serve para calcular e mostrar o fatorial de um número que o usuário digitar

Explicação:

I. `n = int(input("Digite um número: "))`

- O que é?** Esta linha pede ao usuário para digitar um número.
- Pra que serve?** Para que o programa saiba de qual número calcular o fatorial.
- Como funciona?** `input("Digite um número: ")` exibe a mensagem "Digite um número:" na tela e espera que o usuário digite algo. O que o usuário digitar será convertido para um número inteiro (`int`) e armazenado na variável `n`.

II. `fatorial = 1`

- O que é?** Esta linha cria uma variável chamada `fatorial`.
- Pra que serve?** Para armazenar o resultado do cálculo do fatorial.
- Como funciona?** A variável `fatorial` começa com o valor 1, porque o fatorial de qualquer número começa multiplicando por 1.

III. `for i in range(1, n+1):`

- O que é?** Esta linha inicia um loop `for`.
- Pra que serve?** Para repetir uma ação várias vezes, multiplicando os números de 1 até `n`.
- Como funciona?** `range(1, n+1)` cria uma sequência de números de 1 até `n` (inclusive). O loop `for` vai passar por cada um desses números, um por um, e executar o código dentro do loop para cada número.

IV. `fatorial *= i`

- O que é?** Esta linha multiplica o valor atual de `fatorial` pelo valor de `i`.
- Pra que serve?** Para calcular o fatorial do número `n`.
- Como funciona?** `fatorial *= i` é uma forma abreviada de escrever `fatorial = fatorial * i`. A cada iteração do loop, o valor de `fatorial` é atualizado multiplicando-o pelo valor atual de `i`.

V. `print(f"O fatorial de {n} é {fatorial}")`

- O que é?** Esta linha exibe o resultado na tela.
- Pra que serve?** Para mostrar ao usuário o fatorial do número que ele digitou.
- Como funciona?** `print()` é uma função que exibe algo na tela. A string `f"O fatorial de {n} é {fatorial}"` é uma string formatada, onde `{n}` e `{fatorial}` são substituídos pelos valores das variáveis `n` e `fatorial`, respectivamente.

Explicação:**I. `n = int(input("Digite um número: "))`**

- O que é?** Esta linha pede ao usuário para digitar um número.
- Pra que serve?** Para que o programa saiba quantos números da sequência de Fibonacci deve exibir.
- Como funciona?** `input("Digite um número: ")` exibe a mensagem "Digite um número:" na tela e espera que o usuário digite algo. O que o usuário digitar será convertido para um número inteiro (`int`) e armazenado na variável `n`.

II. `a, b = 0, 1`

- O que é?** Esta linha cria duas variáveis, `a` e `b`.
- Pra que serve?** Para armazenar os dois primeiros números da sequência de Fibonacci.
- Como funciona?** `a` começa com o valor 0 e `b` com o valor 1. Esses são os dois primeiros números da sequência de Fibonacci.

III. `for _ in range(n):`

- O que é?** Esta linha inicia um loop `for`.
- Pra que serve?** Para repetir uma ação várias vezes, exibindo os números da sequência de Fibonacci.
- Como funciona?** `range(n)` cria uma sequência de números de 0 até `n-1`. O loop `for` vai passar por cada um desses números, um por um, e executar o código dentro do loop para cada número. O `_` é uma variável de loop que não será usada dentro do loop, então usamos `_` para indicar que não precisamos de um nome para essa variável.

IV. `print(a, end=" ")`

- O que é?** Esta linha exibe o valor de `a` na tela.
- Pra que serve?** Para mostrar o número atual da sequência de Fibonacci.
- Como funciona?** `print(a)` exibe o valor de `a`. O parâmetro `end=" "` faz com que o próximo `print` seja na mesma linha, separado por um espaço em vez de uma nova linha.

V. `a, b = b, a + b`

- O que é?** Esta linha atualiza os valores de `a` e `b`.
- Pra que serve?** Para calcular o próximo número da sequência de Fibonacci.
- Como funciona?** `a, b = b, a + b` é uma atribuição múltipla. Ela faz duas coisas ao mesmo tempo:
 - `a` recebe o valor de `b` (o próximo número da sequência).
 - `b` recebe o valor de `a + b` (a soma dos dois números anteriores, que será o próximo número da sequência).

ex007 - Ordenação de lista

Explicação:

I. `def selection_sort(lista):`

- a. **O que é?** É uma definição de função em Python.
- b. **Para que serve?** Serve para criar uma função que pode ser chamada posteriormente para realizar uma tarefa específica.
- c. **Como funciona?** Aqui, estamos definindo uma função chamada `selection_sort` que recebe uma lista como argumento e a ordena.

II. `for i in range(len(lista)):`

- a. **O que é?** É um loop "para cada" em inglês.
- b. **Para que serve?** Serve para repetir algo várias vezes, mudando um pouco a cada vez.
- c. **Como funciona?** Aqui, o loop percorre cada elemento da lista, onde `i` é o índice do elemento atual.

III. `min_idx = i`

- a. **O que é?** É uma atribuição de valor.
- b. **Para que serve?** Serve para guardar o índice do menor valor encontrado até o momento.
- c. **Como funciona?** Inicialmente, consideramos que o menor valor está no índice `i`.

IV. `for j in range(i+1, len(lista)):`

- a. **O que é?** Outro loop "para cada" em inglês.
- b. **Para que serve?** Serve para comparar o elemento atual com os elementos restantes na lista.
- c. **Como funciona?** Aqui, o loop percorre os elementos da lista a partir do índice `i+1` até o final da lista.

V. `if lista[j] < lista[min_idx]:`

- a. **O que é?** É uma condição "se" em inglês.
- b. **Para que serve?** Serve para verificar se uma condição é verdadeira.
- c. **Como funciona?** Aqui, verificamos se o elemento no índice `j` é menor que o elemento no índice `min_idx`.

VI. `min_idx = j`

- a. **O que é?** É uma atribuição de valor.
- b. **Para que serve?** Serve para atualizar o índice do menor valor encontrado.
- c. **Como funciona?** Se a condição anterior for verdadeira, atualizamos `min_idx` para `j`.

VII. `lista[i], lista[min_idx] = lista[min_idx], lista[i]`

- a. **O que é?** É uma troca de valores.
- b. **Para que serve?** Serve para trocar os valores de dois elementos na lista.

- c. **Como funciona?** Aqui, trocamos o elemento no índice `i` com o elemento no índice `min_idx`.

VIII. `return lista`

- a. **O que é?** É uma instrução de retorno.
- b. **Para que serve?** Serve para retornar um valor da função.
- c. **Como funciona?** Aqui, retornamos a lista ordenada.

IX. `entrada = input("Digite os números separados por espaços: ")`

- a. **O que é?** É uma função que permite que você digite algo no computador.
- b. **Para que serve?** Serve para que o programa possa receber informações do usuário.
- c. **Como funciona?** Quando o programa chega nessa linha, ele espera você digitar algo e pressionar "Enter". O que você digitou é então usado no programa.

X. `lista = list(map(int, entrada.split()))`

- a. **O que é?** É uma combinação de funções que transforma a entrada em uma lista de inteiros.
- b. **Para que serve?** Serve para converter a entrada do usuário em uma lista de números inteiros.
- c. **Como funciona?** `entrada.split()` divide a entrada em uma lista de strings com base nos espaços. `map(int, ...)` converte cada string em um número inteiro. `list(...)` transforma o resultado em uma lista.

XI. `lista_ordenada = selection_sort(lista)`

- a. **O que é?** É uma chamada de função.
- b. **Para que serve?** Serve para executar a função `selection_sort` com a lista como argumento.
- c. **Como funciona?** Aqui, chamamos a função `selection_sort` com a lista que criamos e armazenamos o resultado na variável `lista_ordenada`.

XII. `print("Lista ordenada:", lista_ordenada)`

- a. **O que é?** É uma função que mostra algo na tela.
- b. **Para que serve?** Serve para que o programa possa mostrar informações para o usuário.
- c. **Como funciona?** Quando o programa chega nessa linha, ele mostra a mensagem "Lista ordenada:" seguida da lista ordenada na tela

Explicação:**1. def e_primo(n):**

- O que é?** É uma definição de função em Python.
- Para que serve?** Serve para criar uma função que pode ser chamada posteriormente para realizar uma tarefa específica.
- Como funciona?** Aqui, estamos definindo uma função chamada e_primo que recebe um número n como argumento e verifica se ele é primo.

2. if n <= 1:

- O que é?** É uma condição "se" em inglês.
- Para que serve?** Serve para verificar se uma condição é verdadeira.
- Como funciona?** Aqui, verificamos se o número n é menor ou igual a 1. Se for, o número não é primo.

3. return False

- O que é?** É uma instrução de retorno.
- Para que serve?** Serve para retornar um valor da função.
- Como funciona?** Se a condição anterior for verdadeira, retornamos False, indicando que o número não é primo.

4. for i in range(2, int(n0.5) + 1):**

- O que é?** É um loop "para cada" em inglês.
- Para que serve?** Serve para repetir algo várias vezes, mudando um pouco a cada vez.
- Como funciona?** Aqui, o loop percorre os números de 2 até a raiz quadrada de n (arredondada para baixo).

5. if n % i == 0:

- O que é?** É uma condição "se" em inglês.
- Para que serve?** Serve para verificar se uma condição é verdadeira.
- Como funciona?** Aqui, verificamos se n é divisível por i (ou seja, se o resto da divisão de n por i é zero).

6. return False

- O que é?** É uma instrução de retorno.
- Para que serve?** Serve para retornar um valor da função.
- Como funciona?** Se a condição anterior for verdadeira, retornamos False, indicando que o número não é primo.

7. return True

- O que é?** É uma instrução de retorno.
- Para que serve?** Serve para retornar um valor da função.

- c. **Como funciona?** Se o loop terminar sem encontrar divisores, retornamos True, indicando que o número é primo.

8. n = int(input("Digite um número: "))

- a. **O que é?** É uma função que permite que você digite algo no computador.
- b. **Para que serve?** Serve para que o programa possa receber informações do usuário.
- c. **Como funciona?** Quando o programa chega nessa linha, ele espera você digitar algo e pressionar "Enter". O que você digitou é então convertido em um número inteiro e armazenado na variável n.

9. if e_primo(n):

- a. **O que é?** É uma condição "se" em inglês.
- b. **Para que serve?** Serve para verificar se uma condição é verdadeira.
- c. **Como funciona?** Aqui, verificamos se o número n é primo chamando a função e_primo(n).

10. print(f'{n} é primo.')

- a. **O que é?** É uma função que mostra algo na tela.
- b. **Para que serve?** Serve para que o programa possa mostrar informações para o usuário.
- c. **Como funciona?** Se o número for primo, mostramos uma mensagem informando isso.

11. else:

- a. **O que é?** É uma condição "senão" em inglês.
- b. **Para que serve?** Serve para executar algo se a condição anterior não for verdadeira.
- c. **Como funciona?** Aqui, se o número não for primo, executamos o código dentro do else.

12. print(f'{n} não é primo.')

- a. **O que é?** É uma função que mostra algo na tela.
- b. **Para que serve?** Serve para que o programa possa mostrar informações para o usuário.
- c. **Como funciona?** Se o número não for primo, mostramos uma mensagem informando isso.

Explicação:

I. def busca_binaria(lista, alvo):

- a. **O que é?** É uma definição de função em Python.
- b. **Para que serve?** Serve para criar uma função que pode ser chamada posteriormente para realizar uma tarefa específica.
- c. **Como funciona?** Aqui, estamos definindo uma função chamada busca_binaria que recebe uma lista e um alvo como argumentos e busca o alvo na lista.

II. esquerda, direita = 0, len(lista) - 1

- a. **O que é?** É uma atribuição de valores.
- b. **Para que serve?** Serve para definir os limites iniciais da busca.
- c. **Como funciona?** esquerda começa no índice 0 (início da lista) e direita começa no último índice da lista (len(lista) - 1).

III. while esquerda <= direita:

- a. **O que é?** É um loop "enquanto" em inglês.
- b. **Para que serve?** Serve para repetir algo enquanto uma condição for verdadeira.
- c. **Como funciona?** Aqui, o loop continua enquanto esquerda for menor ou igual a direita.

IV. meio = (esquerda + direita) // 2

- a. **O que é?** É uma atribuição de valor.
- b. **Para que serve?** Serve para encontrar o índice do meio da lista.
- c. **Como funciona?** Aqui, calculamos o índice do meio da lista usando a fórmula (esquerda + direita) // 2.

V. if lista[meio] == alvo:

- a. **O que é?** É uma condição "se" em inglês.
- b. **Para que serve?** Serve para verificar se uma condição é verdadeira.
- c. **Como funciona?** Aqui, verificamos se o elemento no índice meio é igual ao alvo.

VI. return meio

- a. **O que é?** É uma instrução de retorno.
- b. **Para que serve?** Serve para retornar um valor da função.
- c. **Como funciona?** Se a condição anterior for verdadeira, retornamos o índice meio.

VII. elif lista[meio] < alvo:

- a. **O que é?** É uma condição "senão se" em inglês.
- b. **Para que serve?** Serve para verificar outra condição se a primeira não for verdadeira.

- c. **Como funciona?** Aqui, verificamos se o elemento no índice meio é menor que o alvo.

VIII. esquerda = meio + 1

- a. **O que é?** É uma atribuição de valor.
- b. **Para que serve?** Serve para ajustar o limite esquerdo da busca.
- c. **Como funciona?** Se o elemento no índice meio for menor que o alvo, ajustamos
esquerda para meio + 1.

IX. else:

- a. **O que é?** É uma condição "senão" em inglês.
- b. **Para que serve?** Serve para executar algo se nenhuma das condições anteriores for verdadeira.
- c. **Como funciona?** Aqui, se o elemento no índice meio for maior que o alvo, executamos
executamos o código dentro do else.

X. direita = meio - 1

- a. **O que é?** É uma atribuição de valor.
- b. **Para que serve?** Serve para ajustar o limite direito da busca.
- c. **Como funciona?** Se o elemento no índice meio for maior que o alvo, ajustamos
direita para meio - 1.

XI. return -1

- a. **O que é?** É uma instrução de retorno.
- b. **Para que serve?** Serve para retornar um valor da função.
- c. **Como funciona?** Se o alvo não for encontrado, retornamos -1.

XII. lista = [1, 3, 5, 7, 9, 11, 13]

- a. **O que é?** É uma lista de números.
- b. **Para que serve?** Serve para armazenar os números que serão pesquisados.
- c. **Como funciona?** Aqui, definimos uma lista de números que será usada na busca.

XIII. alvo = 7

- a. **O que é?** É uma variável que armazena o valor que estamos procurando.
- b. **Para que serve?** Serve para definir o número que queremos encontrar na lista.
- c. **Como funciona?** Aqui, definimos que estamos procurando o número 7 na lista.

XIV. indice = busca_binaria(lista, alvo)

- a. **O que é?** É uma chamada de função.
- b. **Para que serve?** Serve para executar a função busca_binaria com a lista e o alvo como argumentos.
- c. **Como funciona?** Aqui, chamamos a função busca_binaria com a lista e o alvo, e armazenamos
armazenamos o resultado na variável indice.

XV. if indice != -1:

- a. **O que é?** É uma condição "se" em inglês.

- b. **Para que serve?** Serve para verificar se uma condição é verdadeira.
- c. **Como funciona?** Aqui, verificamos se o índice retornado pela função não é -1, o que significa que o alvo foi encontrado.

ex010 – Calculo MDC

Explicação:

1. `def mdc(a, b):`

- O que é?** É uma definição de função em Python.
- Para que serve?** Serve para criar uma função que pode ser chamada posteriormente para realizar uma tarefa específica.
- Como funciona?** Aqui, estamos definindo uma função chamada `mdc` que recebe dois números `a` e `b` como argumentos e calcula o Máximo Divisor Comum (MDC) entre eles.

2. `while b != 0:`

- O que é?** É um loop "enquanto" em inglês.
- Para que serve?** Serve para repetir algo enquanto uma condição for verdadeira.
- Como funciona?** Aqui, o loop continua enquanto `b` for diferente de 0.

3. `a, b = b, a % b`

- O que é?** É uma atribuição de valores.
- Para que serve?** Serve para atualizar os valores de `a` e `b` durante o cálculo do MDC.
- Como funciona?** Aqui, `a` recebe o valor de `b` e `b` recebe o resto da divisão de `a` por `b` (`a % b`).

4. `return a`

- O que é?** É uma instrução de retorno.
- Para que serve?** Serve para retornar um valor da função.
- Como funciona?** Quando o loop termina (quando `b` se torna 0), retornamos o valor de `a`, que é o MDC.

5. `print(f'O MDC de {a} e {b} é {mdc(a, b)}')`

- O que é?** É uma função que mostra algo na tela.
- Para que serve?** Serve para que o programa possa mostrar informações para o usuário.
- Como funciona?** Quando o programa chega nessa linha, ele mostra a mensagem "O MDC de {a} e {b} é {mdc(a, b)}", onde {a} e {b} são os números que você digitou e {mdc(a, b)} é o resultado da função `mdc`.

ex011 - Conversão de base

Explicação:

1. `def para_base(n, base):`

- O que é?** É uma definição de função em Python.
- Para que serve?** Serve para criar uma função que pode ser chamada posteriormente para realizar uma tarefa específica.
- Como funciona?** Aqui, estamos definindo uma função chamada `para_base` que recebe um número `n` e uma base `base` como argumentos e converte o número para essa base.

2. `if n == 0:`

- O que é?** É uma condição "se" em inglês.
- Para que serve?** Serve para verificar se uma condição é verdadeira.
- Como funciona?** Aqui, verificamos se o número `n` é igual a 0. Se for, retornamos "0".

3. `return "0"`

- O que é?** É uma instrução de retorno.
- Para que serve?** Serve para retornar um valor da função.
- Como funciona?** Se a condição anterior for verdadeira, retornamos a string "0".

4. `digitos = "0123456789ABCDEF"`

- O que é?** É uma atribuição de valor.
- Para que serve?** Serve para definir uma string com todos os dígitos possíveis para bases de 2 a 16.
- Como funciona?** Aqui, criamos uma string que contém todos os dígitos de 0 a 9 e as letras de A a F, que são usados em bases maiores que 10.

5. `resultado = ""`

- O que é?** É uma atribuição de valor.
- Para que serve?** Serve para inicializar uma string vazia que será usada para armazenar o resultado da conversão.
- Como funciona?** Aqui, criamos uma string vazia chamada `resultado`.

6. `while n > 0:`

- O que é?** É um loop "enquanto" em inglês.
- Para que serve?** Serve para repetir algo enquanto uma condição for verdadeira.
- Como funciona?** Aqui, o loop continua enquanto `n` for maior que 0.

7. `resultado = digitos[n % base] + resultado`

- O que é?** É uma atribuição de valor.
- Para que serve?** Serve para adicionar o dígito correspondente ao resto da divisão de `n` pela base ao início da string `resultado`.
- Como funciona?** Aqui, pegamos o resto da divisão de `n` pela base (`n % base`) e usamos esse valor como índice na string `digitos` para obter o dígito correspondente. Esse dígito é adicionado ao início da string `resultado`.

8. `n //= base`

- a. **O que é?** É uma atribuição de valor.
- b. **Para que serve?** Serve para atualizar o valor de `n` para a divisão inteira de `n` pela base.
- c. **Como funciona?** Aqui, dividimos `n` pela base e atualizamos `n` com o resultado da divisão inteira (`n //= base`).

9. `return resultado`

- a. **O que é?** É uma instrução de retorno.
- b. **Para que serve?** Serve para retornar um valor da função.
- c. **Como funciona?** Quando o loop termina, retornamos a string resultado, que contém o número convertido para a base especificada.

10. `n = int(input("Digite um número: "))`

- a. **O que é?** É uma função que permite que você digite algo no computador.
- b. **Para que serve?** Serve para que o programa possa receber informações do usuário.
- c. **Como funciona?** Quando o programa chega nessa linha, ele espera você digitar algo e pressionar "Enter". O que você digitou é então convertido em um número inteiro e armazenado na variável `n`.

11. `base = int(input("Digite a base (2-16): "))`

- a. **O que é?** É uma função que permite que você digite algo no computador.
- b. **Para que serve?** Serve para que o programa possa receber informações do usuário.
- c. **Como funciona?** Quando o programa chega nessa linha, ele espera você digitar algo e pressionar "Enter". O que você digitou é então convertido em um número inteiro e armazenado na variável `base`.

12. `print(f'O número {n} na base {base} é {para_base(n, base)}')`

- a. **O que é?** É uma função que mostra algo na tela.
- b. **Para que serve?** Serve para que o programa possa mostrar informações para o usuário.
- c. **Como funciona?** Quando o programa chega nessa linha, ele mostra a mensagem "O número {n} na base {base} é {para_base(n, base)}", onde {n} é o número que você digitou, {base} é a base que você escolheu e {para_base(n, base)} é o resultado da conversão.

Explicação:

1. def mochila(capacidade, pesos, valores, n):

- a. **O que é?** É uma definição de função em Python.
- b. **Para que serve?** Serve para criar uma função que pode ser chamada posteriormente para realizar uma tarefa específica.
- c. **Como funciona?** Aqui, estamos definindo uma função chamada mochila que recebe a capacidade da mochila, uma lista de pesos, uma lista de valores e o número de itens como argumentos. A função calcula o valor máximo que pode ser colocado na mochila.

2. dp = [[0 for _ in range(capacidade + 1)] for _ in range(n + 1)]

- a. **O que é?** É uma criação de uma matriz (lista de listas) preenchida com zeros.
- b. **Para que serve?** Serve para armazenar os resultados intermediários da solução do problema da mochila.
- c. **Como funciona?** Aqui, criamos uma matriz dp com n + 1 linhas e capacidade + 1 colunas, todas preenchidas com zeros.

3. for i in range(1, n + 1):

- a. **O que é?** É um loop "para cada" em inglês.
- b. **Para que serve?** Serve para repetir algo várias vezes, mudando um pouco a cada vez.
- c. **Como funciona?** Aqui, o loop percorre os itens de 1 até n.

4. for w in range(1, capacidade + 1):

- a. **O que é?** É outro loop "para cada" em inglês.
- b. **Para que serve?** Serve para repetir algo várias vezes, mudando um pouco a cada vez.
- c. **Como funciona?** Aqui, o loop percorre as capacidades da mochila de 1 até capacidade.

5. if pesos[i-1] <= w:

- a. **O que é?** É uma condição "se" em inglês.
- b. **Para que serve?** Serve para verificar se uma condição é verdadeira.
- c. **Como funciona?** Aqui, verificamos se o peso do item i-1 é menor ou igual à capacidade atual w.

6. dp[i][w] = max(valores[i-1] + dp[i-1][w-pesos[i-1]], dp[i-1][w])

- a. **O que é?** É uma atribuição de valor.
- b. **Para que serve?** Serve para calcular o valor máximo que pode ser obtido com o item i e a capacidade w.
- c. **Como funciona?** Aqui, calculamos o máximo entre o valor do item i-1 mais o valor máximo que pode ser obtido com a capacidade restante (w - pesos[i-1]) e o

valor máximo que pode ser obtido sem o item i-1.

7. else:

- O que é?** É uma condição "senão" em inglês.
- Para que serve?** Serve para executar algo se a condição anterior não for verdadeira.
- Como funciona?** Aqui, se o peso do item i-1 for maior que a capacidade atual w, executamos o código dentro do else.

8. `dp[i][w] = dp[i-1][w]`

- O que é?** É uma atribuição de valor.
- Para que serve?** Serve para copiar o valor máximo que pode ser obtido sem o item i-1.
- Como funciona?** Aqui, copiamos o valor máximo que pode ser obtido com a capacidade w sem o item i-1.

9. `return dp[n][capacidade]`

- O que é?** É uma instrução de retorno.
- Para que serve?** Serve para retornar um valor da função.
- Como funciona?** Quando o loop termina, retornamos o valor máximo que pode ser obtido com todos os itens e a capacidade total da mochila.

10. `valores = [60, 100, 120]`

- O que é?** É uma lista de números.
- Para que serve?** Serve para armazenar os valores dos itens.
- Como funciona?** Aqui, definimos uma lista de valores que representam os itens que podem ser colocados na mochila.

11. `pesos = [10, 20, 30]`

- O que é?** É uma lista de números.
- Para que serve?** Serve para armazenar os pesos dos itens.
- Como funciona?** Aqui, definimos uma lista de pesos que representam os itens que podem ser colocados na mochila.

12. `capacidade = 50`

- O que é?** É uma variável que armazena um número.
- Para que serve?** Serve para definir a capacidade máxima da mochila.
- Como funciona?** Aqui, definimos que a capacidade máxima da mochila é 50.

13. `n = len(valores)`

- O que é?** É uma função que retorna o tamanho de uma lista.
- Para que serve?** Serve para obter o número de itens.
- Como funciona?** Aqui, usamos `len(valores)` para obter o número de itens na lista de valores.

14. print("O valor máximo que pode ser colocado na mochila é:", mochila(capacidade, pesos, valores, n))

- a. **O que é?** É uma função que mostra algo na tela.
- b. **Para que serve?** Serve para que o programa possa mostrar informações para o usuário.
- c. **Como funciona?** Quando o programa chega nessa linha, ele mostra a mensagem "O valor máximo que pode ser colocado na mochila é:" seguida do resultado da função mochila.