

Documentação

Desafio de Programação

Warren Tech Academy

Desenvolvedor: Tiago Stasaitis

A resposta para o desafio foi desenvolvida em Python utilizando o Visual Studio Code como IDE. Não existe nenhuma dependência adicional além de ter instalado o Python para executar os scripts.

Para executar os scripts, recomendo realizar o seguinte passo a passo:

1. Abra a pasta do projeto no Visual Studio Code pressionando Ctrl + O ou Ctrl + K e em seguida selecionando a pasta do projeto **/Desafio Warren**.
2. Em seguida, clique com o botão direito sobre algum dos arquivos referentes aos desafios e em seguida em "Run Python File in Terminal".

Feito isso, basta interagir com o script via terminal. As instruções serão apresentadas em tela em todos scripts desenvolvidos..

Descrição das Soluções Apresentadas

Desafio 01:

Este é o único dos algoritmos em que não será solicitada nenhuma entrada do usuário. Ao ser executado o script segue os seguintes passos:

- Criará uma lista inicial com todos os números de 1 a 1 milhão que atendem aos requisitos: soma com seu inverso ser igual a um número ímpar, não iniciar com zero, seu inverso não começar com zero.
- Criará uma nova lista em que, para cada elemento da lista gerada no passo anterior, verificará se o somatório do número em questão com o seu inverso resulta em um valor menor que 1 milhão.
- Por fim, para cada número da lista anterior, irá adicionar ele a uma string se, e somente se, o seu inverso já não tiver sido adicionado (foi necessário para não acontecer por exemplo de se adicionar o número 123 e 321, visto que aqui correspondem ao mesmo número).

Assim, com a string de resultado finalizada, ela é mostrada em tela.

Observação: Tanto a criação de duas listas separadamente como o tratamento do resultado como uma string e não uma lista foram escolhas feitas por motivos de ganho de performance.

Desafio 02:

O algoritmo irá solicitar um número máximo de alunos atrasados aceitável e em seguida iniciará um loop para inserção das informações de chegada aluno por aluno.

Quando informado que não existem mais alunos para inserir o horário de chegada. O algoritmo compara o número de alunos atrasados com o número máximo de alunos atrasados inserido no início da operação. Em seguida é apresentado o resultado da operação.

Desafio 03:

Observação: Este foi o único desafio em que foi necessária a importação de uma biblioteca externa (método “**combinations_with_replacement**” do pacote “**itertools**”) para execução dele, mais especificamente para gerar a lista de combinações possíveis a ser citada mais a frente neste documento.

Durante seu funcionamento, o algoritmo irá solicitar o número que se buscará atingir o somatório. Em seguida, iniciará em loop solicitando número a número os itens da lista de elementos a serem utilizados no somatório. Após finalizada a inserção de todos os números, o algoritmo irá seguir os seguintes passos:

- Gerar uma lista com todas as possibilidades de combinações possíveis dos elementos inseridos na lista.
- Para cada item da lista gerada, analisará qual deles atende o requisito de estar mais próximo ou ser igual ao número alvo.
- Para cada item da lista de itens gerados na etapa anterior, seleciona os que utilizam a menor quantidade de elementos possível.

Desta forma, todas as validações foram finalizadas e é exibido em tela o resultado da operação.