

# **trabalho de programação**

## **sudoku**

Tiago Bernieri Dal Belo

Primeiramente, foi definido um tabuleiro de 9x9 na variável tabela. Esse tabuleiro é constituído por uma lista com outras 9 listas dentro dela.

Após o tabuleiro ser definido, foi feita uma função chamada mostrar, que tem como objetivo imprimir o tabuleiro de Sudoku com as divisões de quadrantes, linhas e colunas para o usuário. A função imprime, através de um loop, uma linha divisória quando o índice das linhas for divisível por 3. Para as colunas, é seguida uma lógica parecida: a cada 3 números é impressa uma barra para dividir os quadrantes. A função ainda coloca um ponto no lugar de todo 0 presente na tabela, para deixá-la mais bonita e legível.

A função `checa_vitória` verifica se ainda existem 0 no tabuleiro; caso ainda exista, o programa continuará. Caso contrário, será mostrada no terminal uma mensagem de vitória.

A função `checa_linha` pega a tabela (lista) e o índice (listas dentro da lista) e verifica se existe um número na mesma lista/linha. Caso exista, ela retorna `True` (verdadeiro); senão, retorna `False` (falso).

A função `checa_coluna` tem um loop `for` que percorre todas as linhas dessa tabela (9 linhas) e, então, pega a variável coluna como índice (posição do número dentro da lista). Caso exista um número em alguma linha na posição do índice, a função retorna `True`; caso contrário, `False`.

A função `checa_quadrante` pega as variáveis linha e coluna e as divide por 3, resultando em um número inteiro, e depois pega esse resultado e o multiplica por 3 para descobrir em qual quadrante a linha e a coluna se iniciam. Após a divisão, ela percorre as 3 linhas e colunas do quadrante e verifica se o número está dentro dele.

O código possui uma randomização de caracteres para a tabela utilizando o `import random`. Ele randomiza a coluna, a linha e o número que será inserido na posição aleatória e, então, faz a modificação do número na posição da linha e coluna que foram aleatorizadas.

A lógica principal utiliza um loop infinito `while True` para o funcionamento do jogo. Primeiramente, são impressas 3 opções: uma que inicia o jogo, outra que imprime as regras e outra que sai do loop infinito, encerrando o código. No jogo, é pedido ao usuário a linha, a coluna e o número que será inserido nessa posição descrita pelo jogador. Após o usuário digitar o que foi pedido, é checado se os números inseridos por ele são válidos. Caso o número seja válido, os valores entram em outro loop `while` para realizar as checagens das funções. Caso alguma checagem das funções retorne verdadeiro, é mostrada uma mensagem de que o número não pode ser inserido no local designado e, então, o loop é quebrado e volta para o início do loop principal. Caso nenhuma das funções retorne verdadeiro, o número digitado é inserido na coluna e linha descritas pelo usuário. Após esse loop, a função de checagem de vitória é chamada; caso a função retorne verdadeiro, é impressa uma mensagem de vitória e o loop é encerrado. Senão, o loop continua.

As fontes buscadas foram principalmente Stack Overflow e a utilização do ChatGPT para a ajuda nas funções de `checa_quadrante`, vitória e de mostrar a tabela, além de correções no código.

Os problemas encontrados foram as checagens das linhas, colunas e quadrantes, além da checagem da vitória e a impressão da tabela organizada. A solução encontrada foi a criação de uma função para cada um dos problemas com suas respectivas soluções. A solução para a checagem de linhas foi verificar se o número está presente na lista na qual o usuário digitou no terminal. A solução para a checagem das colunas foi a realização de um loop for que percorre cada lista dentro da lista principal e verifica a posição do número dentro da lista digitado pelo usuário em todas as 9 listas. Para a checagem do quadrante, foi realizada uma conta que descobre em qual linha e coluna se inicia o quadrante, de acordo com o que o usuário digitou no terminal, e, então, foram realizados loops for que percorrem as 3 linhas e colunas do quadrante para ver se o número do usuário já está nele. Para a checagem de vitória, utilizamos a função `all`, que faz várias checagens de uma vez. Ela retorna verdadeiro se todas as checagens dentro dela forem verdadeiras, porém, se apenas uma for falsa, ela retorna falso. Dentro da função, existe um for que percorre todas as linhas e verifica se algum número dentro da linha é zero; caso o número zero seja encontrado em qualquer linha, ela retorna `False`. Senão, retorna `True`, indicando que o usuário terminou o Sudoku. A impressão da tabela é feita através de um loop que coloca uma linha divisória quando o índice das linhas for divisível por 3. Para as colunas, é seguida uma lógica parecida: a cada 3 números é impressa uma barra para dividir os quadrantes. A função ainda coloca um ponto no lugar de todo 0 presente na tabela, para deixá-la mais bonita e legível.