Teste 1 (versão 1)

Wellington Silva
Tiago Barradas
Escola de Matemática Aplicada, Fundação Getulio Vargas, Brasil.

April 1, 2023

Problema 1: Dado um natural n, seu objetivo aqui é construir uma função que irá printar uma pirâmide 2D de altura n, usando '#'. Depois, ela irá printar essa mesma pirâmide de cabeça pra baixo, formando um losango. Como nos exemplos:

```
>>> 3
  ##
 ####
######
 ####
  ##

>>> 4
    ##
   ####
  ######
########
  ######
   ####
    ##
```

Problema 2: Faça uma função "is_prime" que dado um número n menor que 100000, retorna 1 se n é primo e 0 se n não é primo. Siga o exemplo:

```
>>> 3
1

>>> 8
0

>>> 11
1
```

Agora com a função do item anterior faça uma função recursiva que contará todos os primos entre 1 e o m dado, completando o seguinte código:

```
#include <iostream>
using namespace std;

bool is_prime(int n) {
```

```
    // Função do item anterior
}

int count_primes(int n) {
if (n == 1) {
        return ... ; // Complete aqui
    } else {
        return ... ; // Complete aqui
    };
}

int main() {
    int n;

    cin >> n;

    cout << count_primes(n) << endl;
}
```

Problema 3: 'Connect 4' é um jogo composto por um "tabuleiro" vertical dividido em 6 linhas e 7 colunas, onde os jogadores alternam, colocando suas peças no topo de uma coluna escolhida. Como o tabuleiro é vertical, a peça cai até a linha mais embaixo possível naquela coluna. O objetivo do jogador é conectar quatro peças da sua cor em uma linha reta, que pode ser vertical, horizontal, ou diagonal.

Implemente uma versão funcional desse jogo através de uma array de chars, onde 'O' representa um espaço vazio, 'L' e 'X' representam as duas cores de peças. O tabuleiro começa inteiramente vazio, e as peças deverão ser adicionadas através de uma função "jogada", que recebe a array do tabuleiro, um char representando o tipo de peça a ser jogado, e uma int, representando o número da coluna onde a peça será inserida (de 1 a 7).

Em cada jogada, deverá ser checado se o movimento é válido (ou seja, se está sendo feito em uma coluna existente e que não está cheia), e também deverá ser checado se o movimento levará a vitória do jogador. Independentemente dessas checagens, a função também irá printar o estado atual do tabuleiro após o processamento da jogada. Segue o exemplo:

```
>>>jogada(tabuleiro, "X", 7)
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O X

>>>jogada(tabuleiro, "L", 8)
Jogada inválida!
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O X
```