

DOCUMENTAÇÃO DO CÓDIGO

1) INTRODUÇÃO

O presente documento tem por objetivo apresentar para os recrutadores a lógica aplicada durante o desenvolvimento do presente algoritmo. Ao decorrer da literatura, trarei as funções criadas para a realização da tarefa e explicarei suas funcionalidades e minha lógica para criá-las.

2) FUNÇÕES CRIADAS

2.1) void limpar_tela () :

C/C++

```
void limpar_tela () {  
    //Verifica qual é o S.O. para limpar o terminal com o comando  
    compatível  
    #ifdef _WIN32  
        system("cls"); // Windows  
    #else  
        system("clear"); // Linux && mac  
    #endif  
}
```

Função usada meramente por estética e organização do terminal. Sua funcionalidade é apagar o texto apresentado durante a execução do programa. Por conta de sua funcionalidade simples, ela não recebe nenhum parâmetro e também não retorna nenhum valor, somente usa a função do sistema “cls” (Windows) ou “clear” (Linux ou MacOS) para fazer a limpeza do output.

2.2) void apresentar_palitos (int quantidade) :

```
C/C++
void apresentar_palitos(int quantidade) {
    for(int i = 0; i < quantidade; i++) {
        cout << "| ";
    }
    cout << endl;
    cout << "Numero de palitos restantes: " << quantidade << endl;
}
```

Função responsável por cumprir o requisito de apresentar a quantidade de pauzinhos após uma jogada ou após o input da quantidade inicial. A função recebe como parâmetro a quantidade atual de pauzinhos disponíveis para serem retirados, e os apresenta por meio de um *for* que vai de 0 à quantidade disponível - 1, sendo que a cada iteração, esse *for* apresenta no terminal um símbolo “|”. Ao final desse laço de repetição, é apresentado numericamente a quantidade restante.

2.3) int jogada_robo (int quantidade) :

```
C/C++
int jogada_robo (int quantidade) {
    int jogada = (quantidade % 4);

    //Caso não tenha a possibilidade, retira 1 palito.
    if (jogada == 0) {
        return 1;
    }

    return jogada;
}
```

Essa função implementa a estratégia do robô no jogo dos pauzinhos. A lógica aplicada aqui é a seguinte: o robô tenta sempre retirar uma quantidade de palitos que deixe o jogador com um número múltiplo de 4. Para garantir que isso ocorra, a função recebe como parâmetro o número atual de pauzinhos e calcula o valor da jogada com base no resto da divisão desse número por 4. Se o valor calculado for 0 (ou seja, já é um múltiplo de 4), o robô retira apenas 1 palito, mas caso sobre algum outro valor entre 1 ou 3, esse é o valor retirado. A função retorna a quantidade de palitos que o robô deve retirar.

2.4) jogada_humano (int retirada) :

```
C/C++
int jogada_humano (int retirada) {
    cout << "Digite quantos palitos pegar (1, 2 ou 3): ";
    cin >> retirada;
    cout << endl;

    // Verifica se o valor digitado pelo usuário está entre os valores
    permitidos
    while(retirada < MIN_RETIRADA || retirada > MAX_RETIRADA) {
        cout << "Valor invalido! Valem somente as retiradas de valores entre
" << MIN_RETIRADA << " e " << MAX_RETIRADA << " palitos" << endl;
        cout << "> ";
        cin >> retirada;
    }

    return retirada;
}
```

Essa função é responsável por receber e verificar a jogada do jogador humano. O jogador deve escolher quantos palitos ele deseja retirar (entre 1, 2 ou 3), e a função verifica se a jogada é válida. Se o valor inserido estiver fora do intervalo permitido (menor que 1 ou maior que 3), o programa pede novamente uma entrada válida. Quando uma jogada correta é recebida, a função retorna o número de palitos retirados pelo jogador.

2.5) jogo_dos_palitos (int quantidade_total) :

```
C/C++
void jogo_dos_palitos (int quantidade_total) {
    int retirada = 0;

    // Looping enquanto ainda houver palitos
    do {
        apresentar_palitos(quantidade_total);

        retirada = jogada_humano(quantidade_total);
        quantidade_total -= retirada; // Subtrai os palitos retirados pelo
jogador humano

        // Caso o humano vença, o looping é interrompido e retorna para a
função main
    }
```

```

    if(quantidade_total == 0) {
        cout << "Fim do jogo -> Voce venceu!" << endl;
        return;
    }

    apresentar_palitos(quantidade_total);

    // Jogada do bot e retirada dos palitos por ele feita
    retirada = jogada_robo(quantidade_total);
    quantidade_total -= retirada;

    // Apresenta jogada do robô
    cout << "O computador pegou " << retirada << " palitos" << endl;
    cout << endl;
} while(quantidade_total > 0);

cout << "Fim do jogo -> O computador venceu" << endl;
}

```

Essa função pode ser considerada o coração do jogo, pois é onde ocorre a interação entre o jogador e o robô. Ela começa chamando a `apresentar_palitos()` para exibir a quantidade inicial de palitos. Em seguida, recebe a jogada do jogador humano através da `jogada_humano()` e atualiza a quantidade de palitos restantes.

Se o jogador humano acabar com todos os palitos, o jogo declara sua vitória e encerra a função. Caso contrário, é a vez do robô jogar, com base na função `jogada_robo()`. Após a jogada do robô, o jogo verifica novamente se ele venceu, e o processo continua até que não sobrem mais palitos, assim, declarando um dos jogadores como vencedor.

2.6) int main () :

```
C/C++
int main () {
    int quantidade_total = 0;

    cout << "Digite a quantidade inicial de palitos: ";
    cin >> quantidade_total;

    limpar_tela();

    // Inicia o jogo e declara o ganhador
    jogo_dos_palitos(quantidade_total);

    return 0;
}
```

A função main é responsável pelo fluxo principal do jogo, é responsável por receber a quantidade inicial de palitos no jogo, após isso, chama a função `limpar_tela()` para apagar do terminal o input dessa variável. Por fim, chama a função `jogos_dos_palitos()` para iniciar a disputa entre o computador e o jogador.