**import** java.util.Scanner;

**public** **class** Trabalho1

{

**public** **static** **void** main(String[] args)

{

Scanner sc = **new** Scanner(System.***in***);

// variaveis

**int** N; // numero de discos

**int** m\_min; // numero de movimentos minimo

**int** c; // contagem dos movimentos

**int** m; // movimentos possiveis entre 1 a 6

**boolean** acabar\_jogo = **false**; //quando o jogador pretender carregar em Y e acabar o jogo

**boolean** ganhar = **false**; //chegar ao final do jogo

// Inicio do jogo

// SE HOUVER TEMPO METER ASTERISCOS

System.***out***.println("TORRE DE HANOI");

System.***out***.println("O objetivo do jogo e passar todos os discos do pino mais a esquerda para o pino mais a direita, usando o pino intermedio como auxiliar.");

System.***out***.println("Regra do jogo: \nNao pode colocar um disco maior sobre um disco mais pequeno.");

System.***out***.print("Numero de discos (entre 3 e 10): ");

**if** ( sc.hasNextInt() )

{

N = sc.nextInt();

**if** ( N >= 3 && N <= 10 )

{

m\_min = (**int**)Math.*pow*(2,N) - 1;

System.***out***.printf("%nO numero minimo de jogadas para %d rodas e %d.%n", N, m\_min);

// Criar matrizes das torres

// Linhas = N ( discos ) + 3 ( barra de cima + cardinais + letra da torre )

// Colunas = 3 ( disco mais pequeno ) + N \* 2 ( adicionar asteriscos de lado ate chegar a base de cardinais )

String[][] torre\_A = **new** String[N + 3][2\*N + 3]; // NO FINAL TENTAR CRIAR ARRAYS NUM LOOP MAIS COMPACTO

String[][] torre\_B = **new** String[N + 3][2\*N + 3]; // torre auxiliar

String[][] torre\_C = **new** String[N + 3][2\*N + 3];

// Torre de comparacao: vamos igualar a torre A inicial a C final

// Se forem iguais, o jogador ganhou

String[][] torre\_comp = **new** String[N + 3][2\*N + 3];

// construir torre A (preenchida com discos)

**for** (**int** i = 0; i < torre\_A.length; i++) // correr linhas

{

**for** (**int** j = 0; j < torre\_A[0].length; j++) // correr colunas

{

// Vamos comecar pelas linhas simples ( 1a, penultima e ultima)

**if** ( i == 0 ) // 1a linha da matriz

{

**if** ( j == torre\_A[0].length/2 )

{

torre\_A[i][j] = "|"; // barra da torre ao meio

}

**else**

{

torre\_A[i][j] = " ";

}

}

**else** **if** ( i == torre\_A.length - 2 ) // penultima linha - base de cardinais

{

torre\_A[i][j] = "#";

}

**else** **if** ( i == torre\_A.length - 1 ) // ultima linha

{

**if** ( j == torre\_A[0].length/2 )

{

torre\_A[i][j] = "A"; // nome da torre no meio

}

**else**

{

torre\_A[i][j] = " ";

}

}

**else** // restantes colunas ( colunas em que estao os discos: \*)

{

// vamos comecar com o disco mais pequeno e descer

**if** (j > N-i && j < torre\_A[0].length - 1 -(N-i))

{

torre\_A[i][j] = "\*";

}

**else**

{

torre\_A[i][j] = " ";

}

}

}

}

// construir torre B e C

**for** (**int** i = 0; i < torre\_B.length; i++) // tamanho das matrizes são iguais

{

**for** (**int** j = 0; j < torre\_B[0].length; j++)

{

**if** (i != N+1 && i != N+2)

{

**if** (j == torre\_B[0].length/2)

{

torre\_B[i][j] = "|";

torre\_C[i][j] = "|";

}

**else**

{

torre\_B[i][j] = " ";

torre\_C[i][j] = " ";

}

}

**else** **if** (i == torre\_B.length-2)

{

torre\_B[i][j] = "#";

torre\_C[i][j] = "#";

}

**else** // i==N+2

{

**if** (j == torre\_B[0].length/2)

{

torre\_B[i][j] = "B";

torre\_C[i][j] = "C";

}

**else**

{

torre\_B[i][j] = " ";

torre\_C[i][j] = " ";

}

}

}

}

**for**(**int** i = 0; i < torre\_A.length; i++)

{

**for**(**int** j = 0; j < torre\_A[i].length; ++j)

{

System.***out***.print(torre\_A[i][j]);

}

**for**(**int** j = 0; j < torre\_B[i].length; ++j)

{

System.***out***.print(torre\_B[i][j]);

}

**for**(**int** j = 0; j < torre\_C[i].length; ++j)

{

System.***out***.print(torre\_C[i][j]);

}

System.***out***.println();

}

System.***out***.println("Se, em alguma jogada, quiser acabar o jogo mais cedo, selecione 'Y' ou 'y'.");

String yy = sc.next();

sc.nextLine();

**if** (yy.equals("Y") || yy.equals("y"))

{

acabar\_jogo = **true**;

}

}

**else**

{

System.***out***.println("ERRO: numero de discos invalido. Reinicie o jogo.");

}

}

**else**

{

System.***out***.println("ERRO: valor introduzido nao e inteiro. Reinicie o jogo. ");

}

}

}

// "Se quiser acabar o jogo mais cedo, selecione 'Y' ou 'y'.\nSe pretende continuar selecione um tecla qualquer: "