a) Criar uma matriz 4x4 de inteiros sendo que cada elemento da matriz será preenchido com o dobro do elemento anterior (o elemento [0][0] será o 2):

```
import java.util.Scanner;

public class ExercicioA {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int[][] matrizA = new int[4][4];
        matrizA[0][0] = 2;

        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                if (i == 0 && j == 0) continue;
                matrizA[i][j] = matrizA[i - 1][j] * 2;
            }
        }

        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                System.out.print(matrizA[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

b) Criar uma matriz de 3x3 de inteiros, que deverá ser preenchida aleatoriamente. Fazer uma busca que indique o maior e o menor valor da matriz e qual a posição em que ele está (qual a sua linha e coluna)

```
import java.util.Scanner;
import java.util.Random;

public class ExercicioB {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      Random rand = new Random();

      int[][] matrizB = new int[3][3];

      for (int i = 0; i < 3; i++) {
         for (int j = 0; j < 3; j++) {
            matrizB[i][j] = rand.nextInt(100);
         }
      }

      System.out.println("Matriz B:");
      for (int i = 0; i < 3; i++) {
         for (int j = 0; j < 3; j++) {
            System.out.print(matrizB[i][j] + " ");
         }
         System.out.println();
      }

      int menorValor = matrizB[0][0];
      int maiorValor = matrizB[0][0];
      int linhaMenor = 0, colunaMenor = 0, linhaMaior = 0, colunaMaior = 0;

      for (int i = 0; i < 3; i++) {
         for (int j = 0; j < 3; j++) {
            if (matrizB[i][j] < menorValor) {
               menorValor = matrizB[i][j];
               linhaMenor = i;
               colunaMenor = j;
            }
            if (matrizB[i][j] > maiorValor) {
               maiorValor = matrizB[i][j];
               linhaMaior = i;
               colunaMaior = j;
            }
         }
      }

      System.out.println("Menor valor: " + menorValor + " na posição [" + linhaMenor
+ "][" + colunaMenor + "]");
      System.out.println("Maior valor: " + maiorValor + " na posição [" + linhaMaior
```

```
+ "][" + colunaMaior + "]");
    }
}
```

c) Criar uma matriz 3x4 sendo que na última coluna deverá ter a soma de cada linha

```
import java.util.Scanner;

public class ExercicioC {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int[][] matrizC = new int[3][4];

        for (int i = 0; i < 3; i++) {
            int somaLinha = 0;
            for (int j = 0; j < 3; j++) {
                matrizC[i][j] = scanner.nextInt();
                somaLinha += matrizC[i][j];
            }
            matrizC[i][3] = somaLinha;
        }

        System.out.println("Matriz C com a soma das linhas na última coluna:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 4; j++) {
                System.out.print(matrizC[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

d) Criar uma cartela de bingo aleatoriamente com 16 posições (4x4), indo de 01 a 75. Ao cantar os números informar se a cartela tem ou não os números cantados. O jogador ganha quando acerta as 16 dezenas, nesse caso, informa a frase BINGO e diz quantas rodadas ele demorou para acertar.

```
import java.util.Random;

public class ExercicioD {
    public static void main(String[] args) {
        int[][] cartelaBingo = gerarCartela();
      exibirCartela(cartelaBingo);


        for (int i = 1; i <= 16; i++) {
            System.out.println("Número chamado: " + i);
            if (verificarBingo(cartelaBingo, i)) {
                System.out.println("BINGO! O jogador acertou todas as dezenas.");
                System.out.println("Rodada: " + i);
                break;
            }
        }
    }

    // Método para gerar uma cartela aleatória
    static int[][] gerarCartela() {
        int[][] cartela = new int[4][4];
        Random random = new Random();

        // Preenche a cartela com números sem repetição
        int[] numeros = new int[16];
        for (int i = 0; i < 16; i++) {
            numeros[i] = i + 1;
        }

        // Embaralha os números
        for (int i = 15; i > 0; i--) {
            int indiceAleatorio = random.nextInt(i + 1);
            int temp = numeros[i];
            numeros[i] = numeros[indiceAleatorio];
            numeros[indiceAleatorio] = temp;
        }

        // Preenche a cartela com os números embaralhados
        int k = 0;
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
```

```
                cartela[i][j] = numeros[k++];
            }
        }

        return cartela;
    }

    // Método para exibir a cartela
    static void exibirCartela(int[][] cartela) {
        System.out.println("Cartela de Bingo:");
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                System.out.printf("%2d ", cartela[i][j]);
            }
            System.out.println();
        }
    }

    // Método para verificar se há BINGO
    static boolean verificarBingo(int[][] cartela, int numeroChamado) {
        // Marcar o número na cartela (substituir por 0)
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                if (cartela[i][j] == numeroChamado) {
                    cartela[i][j] = 0;
                }
            }
        }

        // Verificar linhas
        for (int i = 0; i < 4; i++) {
            if (cartela[i][0] == 0 && cartela[i][1] == 0 && cartela[i][2] == 0 &&
cartela[i][3] == 0) {
                return true;
            }
        }

        // Verificar colunas
        for (int j = 0; j < 4; j++) {
            if (cartela[0][j] == 0 && cartela[1][j] == 0 && cartela[2][j] == 0 &&
cartela[3][j] == 0) {
                return true;
            }
        }

        // Verificar diagonais
        if ((cartela[0][0] == 0 && cartela[1][1] == 0 && cartela[2][2] == 0 &&
cartela[3][3] == 0) ||
            (cartela[0][3] == 0 && cartela[1][2] == 0 && cartela[2][1] == 0 &&
cartela[3][0] == 0)) {
```

```
            return true;
        }

        return false;
    }
}
```