

1-

```
int[][] board = new int[8][8];
for (int i = 0; i < 8; i++) {
    for (int j = 0; j < 8; j++) {
        if ((i + j) % 2 == 0) {
            board[i][j] = 0; // posição par recebe zero
        } else {
            board[i][j] = 1; // posição impar recebe um
        }
    }
}
```

2-

```
String[][] board = new String[3][3];
board[0][0] = "x";
board[0][1] = "";
board[0][2] = "o";
board[1][0] = "";
board[1][1] = "o";
board[1][2] = "";
board[2][0] = "o";
board[2][1] = "";
board[2][2] = "x";
```

5- O conteúdo de names é ["ann","cal"] porque o comando names.remove(1) remove o elemento "bob".

6-

Aqui tem um erro de sintaxe esta a faltar " " na palavra bob.

8-

```
import java.util.Random;

int[] numeros = new int[10];

Random random = new Random();

for (int i = 0; i < numeros.length; i++) {
    numeros[i] = random.nextInt(100) + 1;
}
```

TPC2 PARTE 1

2-

a) Soma dos números pares entre 2 e 100

```
int soma = 0;
for (int i = 2; i <= 100; i += 2) {
```

```

        soma += i;
    }
    System.out.println(" Soma dos números pares entre 2 e 100: " + soma);

```

b) Soma dos quadrados entre 1 e 100

```

    int soma = 0;
    for (int i = 1; i <= 100; i++) {
        soma += i * i;
    }
    System.out.println(" Soma dos quadrados entre 1 e 100: " + soma);

```

d) Soma dos dígitos ímpares de n

```

    System.out.print(" Digite um número inteiro positivo: ");
    int n = scanner.nextInt();
    int soma = 0;
    while (n > 0) {
        int digito = n % 10;
        if (digito % 2 != 0) {
            soma += digito;
        }
        n /= 10;
    }
    System.out.println("Soma dos dígitos ímpares de n: " + soma);
}
}

```

3-

- a) 2 3 4 5 6 7 8 9
- b) 3 4 5 6 7 8 9
- c) 9 8 7 6 5 4 3 2
- d) 1 2 3 4 5 6 7 8 9
- e) 2 4 6 8
- f) 2 4 6 8

4-

- a)10
- b)10
- c)10
- d)21
- e)número infinito de interações
- F)6
- g)3

5-

```

int s=0;
int i=1;
while (i <= 10) {
    s = s + i;
    i++;
}

```

6-

```
int n = in.nextInt();
double x = 0;
double s = 1.0 / (1 + n * n);
while (s > 0.01) {
    n++;
    x += s;
    s = 1.0 / (1 + n * n);
}
```

7-

- A) 3 4 5 6
- B) 1 4 9
- C) 15 6

8-

- A) 5