



/*

* Escreva um algoritmo que mostre na tela uma escada de tamanho n utilizando o
* caractere * e espaços. A base e altura da escada devem ser iguais ao valor de
* n. A última linha não deve conter nenhum espaço.
*/

```
public static void main(String[] args) {

    Scanner leia = new Scanner(System.in);
    List<String> degraus = new ArrayList<>();

    System.out.println("Olá, seja Bem Vinde!! ");
    System.out.println("Vamos fazer uma escada de * ?? ");
    System.out.printf("Me diga quantos degraus ela terá: ");
    int qtdDegraus = leia.nextInt();

    for (int i = 0; i < qtdDegraus; i++) {
        degraus.add(" ".repeat(qtdDegraus - i) + "*".repeat(i + 1));
    }

    for (String d : degraus) {
        System.out.println(d);
    }

    System.out.print("Obrigado por participar!! ");
}
```



desafio 2

- * O Desafio: Débora digitou uma string aleatória no campo de senha, porém ela
 - * não tem certeza se é uma senha forte. Para ajudar Débora, construa um
 - * algoritmo que informe qual é o número mínimo de caracteres que devem ser
 - * adicionados para uma string qualquer ser considerada segura.

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    boolean senhaInvalida = true;

    do {

        boolean digitouNumero = false;
        boolean digitouLetra = false;
        boolean digitouEspecial = false;

        System.out.printf("\n");
        System.out.println("Sua senha deve conter:");
        System.out.printf("\n");
        System.out.println("Possuir no mínimo 6 caracteres.");
        System.out.println("Contém no mínimo 1 dígito.");
        System.out.println("Contém no mínimo 1 letra em minúsculo.");
        System.out.println("Contém no mínimo 1 letra em maiúsculo.");
        System.out.println("Contém no mínimo 1 caractere especial. Os
caracteres especiais são: !@#$%^&*()-+");
        System.out.printf("\n");

        System.out.printf("Digite uma senha:");
        String usuarioSenha = scanner.nextLine();

        if (usuarioSenha.length() >= 6) {
            char[] charSenha = usuarioSenha.toCharArray();
            for (char chTest : charSenha) {
                if (Character.isDigit(chTest)) {
                    digitouNumero = true;

                } else if (Character.isLetter(chTest)) {
```

```

        digitouLetra = true;

        } else {
            digitouEspecial = true;

        }

    }

    if (digitouNumero && digitouLetra && digitouEspecial) {
        System.out.println("Parabéns, sua senha é uma senha
segura!");
        break;
    } else {
        System.out.printf("\n");
        System.out.println("Desculpe, sua senha não é segura!");
        System.out.println("Siga as instruções para criar um sanha
forte!");
    }

} while (senhaInvalida);

}

```



desafio 3

* Dada uma string qualquer, desenvolva um algoritmo que encontre o número de
* pares de substrings que são anagramas.
*/

```
public static void main(String[] args) {

    System.out.printf("\nOlá, vamos para mais um desafio?");
    System.out.printf("\nDigite uma palavra e descubra quantos anagramas de
pares ela tem!");
    System.out.printf("\nDigite a palavra:");
    Scanner leia = new Scanner(System.in);
    String anagrama = leia.next();
    separandoAnagrama(anagrama);
}

public static void separandoAnagrama(String texto) {
    HashMap<String, Integer> map = new HashMap<>();
    for (int i = 0; i < texto.length(); i++) {
        for (int a = i; a < texto.length(); a++) {
            char[] ch = texto.substring(i, a + 1).toCharArray();
            Arrays.sort(ch);
            String letra = new String(ch);
            if (map.containsKey(letra))
                map.put(letra, map.get(letra) + 1);
            else
                map.put(letra, 1);
        }
    }
    int somaAna = 0;
    for (String key : map.keySet()) {
        int n = map.get(key);
        somaAna += (n * (n - 1)) / 2;
    }
    System.out.println();
    System.out.println("O número de pares de anagramas da palavra que você
digiou é " + somaAna);
}
```