

```

public static String helloWorld(String name) {
    // 1
    if ((name == null) || (name == "")) {
        return "Hello World";
    } else
        return "Hello " + name;
}

public static String[] removeNullElements(String[] array) {
    // 2
    List<String> memory = new ArrayList<>();
    for (int i = 0; i < array.length; i++) {
        if (array[i] != null) {
            memory.add(array[i]);
        }
    }
    String[] result = new String[memory.size()];
    result = memory.toArray(result);
    return result;
}

public static int[] addElementToBeginning(int[] array, int element) {
    // 3
    int[] result = new int[(array.length + 1)];

    result[0] = element;

    for (int i = 0; i < array.length; i++) {
        result[i + 1] = array[i];
    }
    return result;
}

public static int[] allElementsExceptFirstThree(int[] array) {
    // 4
    List<Integer> memory = new ArrayList<>();
    for (int i = 3; i < array.length; i++) {
        memory.add(array[i]);
    }
    int[] result = new int[memory.size()];

    for (int i = 0; i < result.length; i++) {
        result[i] = memory.get(i);
    }
    return result;
}

public static String getFirstHalf(String word) {
    // 5
    int halfWordLength = Math.round((float) word.length() / 2);

    String halfString = word.substring(0, halfWordLength);

    return halfString;
}

public static String[] selectElementsStartingWithA(String[] array) {
    // 6
    List<String> memory = new ArrayList<>();
    for (int i = 0; i < array.length; i++) {

```

```

        char first = array[i].charAt(0);
        if (first == 'a') {
            memory.add(array[i]);
        }
    }
    String[] listStartWithA = new String[memory.size()];
    listStartWithA = memory.toArray(listStartWithA);

    return listStartWithA;
}

public static String[] selectElementsStartingWithVowel(String[] array) {
    // 7
    List<String> list = new ArrayList<>();
    for (int i = 0; i < array.length; i++) {

        if (array[i] != null) {
            char first = array[i].charAt(0);
            if (first == 'a' || first == 'e' || first == 'i' || first == 'o' || first
== 'u') {
                list.add(array[i]);
            }
        }
    }
    String[] listStartWithA = new String[list.size()];
    listStartWithA = list.toArray(listStartWithA);

    return listStartWithA;
}

public static String[] reverseOrderInArray(String[] array) {
    // 8
    String[] reverse = new String[array.length];
    //Array to reverse
    int j = array.length - 1;
    for (String element : array) {
        reverse[j] = element;
        j--;
    }
    return reverse;
}

public static int[] insertElementInTheMiddleOfAnArray(int[] array, int element) {
    // 9
    // Array to arrayList

    List<Integer> listClone = new ArrayList<>();
    for (int num : array) {
        listClone.add(num);
    }
    //Add element in the middle of the arrayList

    int middleArray = Math.round(array.length / 2);
    listClone.add(middleArray, element);

    //ArrayList to array
    int[] newArray = new int[listClone.size()];

    for (int i = 0; i < newArray.length; i++) {
        newArray[i] = listClone.get(i);
    }
    return newArray;
}

```

```

}

public static String shortestWord(String text) {
    // 10
    String[] textArray = text.split(" ");
    int length = 100;
    String shortestWord = "";
    for (String str : textArray) {

        if (str.length() < length) {
            length = str.length();
            shortestWord = str;
        }
    }
    return shortestWord;
}

public static String removeCapitals(String text) {
    // 11
    char[] stringToChar = text.toCharArray();
    List<Character> withoutCap = new ArrayList<>();
    for (char character : stringToChar) {
        if (!Character.isUpperCase(character)) {
            withoutCap.add(character);
        }
    }
    String result = new String();
    for (char character : withoutCap) {
        result = result + character;
    }
    return result;
}

public static long addingTwoNumbers(long number1, long number2) {
    // 12
    long result = number1 + number2;
    return result;
}

public static long addingThreeNumbers(long number1, long number2, long number3) {
    // 13
    long result = number1 + number2 + number3;
    return result;
}

public static long addingSeveralNumbers(final Integer... numbers) {
    // 14
    int sum = 0;
    for (int number : numbers) {
        sum = sum + number;
    }
    return sum;
}

public static float makeNegative(float number) {
    // 15
    float negative;
    if (number > 0) {
        negative = 0 - number;
    } else {
        negative = number;
    }
}

```

```

        return negative;
    }

    public static boolean checkForSpecialCharacters(String string) {
        // 16
        Pattern special = Pattern.compile("[@#%$&*()_+=|<>?{}\\[\\]\\~ -]");
        Matcher hasSpecial = special.matcher(string);

        return hasSpecial.find();
    }

    public static boolean checkIfStringStartsWithConsonant(String word) {
        // 17
        String firstChar = String.valueOf(word.charAt(0));

        Pattern consonant = Pattern.compile("[zrtpqsd fghjklmwx cvbn]", Pattern.CASE_INSENSITIVE);
        Matcher hasconsonant = consonant.matcher(firstChar);

        return hasconsonant.find();
    }

    public static String getDomainName(String email) {
        // 18

        String[] split = email.split(".com");
        split = split[0].split("@");

        return split[1];
    }

    public static int[] letterPosition(String name) {
        // 19
        String lowerCaseName = name.toLowerCase();
        char[] alphabet = { ' ', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z' };
        int[] letterPosition = new int[lowerCaseName.length()];
        for (int i = 0; i < lowerCaseName.length(); i++) {
            char target = lowerCaseName.charAt(i);
            int index = Chars.indexOf(alphabet, target);
            letterPosition[i] = index;
        }
        return letterPosition;
    }

    public static boolean isPeer(int number) {
        // 20
        float halfNumber = (float)number / 2;
        if (halfNumber == Math.floor(halfNumber)) {
            return true;
        } else {
            return false;
        }
    }
}

```