Problem 1. Work with this lambda expression: (String s1, String s2) -> s1.compareTo(s2)

```
public class Problem1 {

        // name and type of lambda goes here




        // representing lambda as a method reference
        // Hint: To define the method reference, make use of a helper method.




        //representing lambda as a static nested class









        //evaluate with String inputs: “Hello”, “Allo”

        public void evaluator() {




        }

        public static void main(String[] args) {

                Problem1 p = new Problem1();

                p.evaluator();

        }

}
```

Problem 2. Use Lambdas and Streams to do the followings:

1) Print out only the first names of all the employees.
2) Count the number of last names that begin with the letter 'B'. Print out this number.
3) Print out all the Employee objects whose last name begins with the letter 'B'.
4) Print out All of the employee objects, but if the last name begins with the letter 'I', then capitalize all the letters in the last name.
5) Create a List<Employee> the Employee objects whose last name begins with the letter 'G' and their first name and last name to be All capital letters.
6) Print out all the Employee objects' last names, whose last name begins with the letter 'I', but First name does not begin with the letter 'I'. Print out only the last names.
7) Create an infinite stream of even numbers (0, 2, 4, …) and then, eventually print out only the first 20 even numbers from this stream.
8) Display Employees with salaries in the range $4000-$6000.
9) Display Employees in IT department.