```
package jp.co.acroquest.lambda;

import java.util.function.BiFunction;
import java.util.function.Consumer;
import java.util.function.Function;
import java.util.function.Predicate;
import java.util.function.Supplier;

/**
 * Answer following questions.
 *
 * @see <a href="https://www.baeldung.com/java-8-functional-interfaces">Baeldung.com - Functional Interfaces in Java 8</a>
 */
public class LabmdaQuestion
{

    protected static final RuntimeException RuntimeException = null;

    /**
     * Question 1:
     * <pre>
     *      Return function to convert strings to upper case by using a labmda expression.
     * </pre>
     *
     * @return
     */
    public Function<String, String> mapToUpperCase()
    {
        return (String changeWord) -> changeWord.toUpperCase();
    }

    /**
     * Question 2:
```

```
     * <pre>
     *      Return function to create FullName by given string.
     *      Example: "John Do" -> FullName(firstName: "John", lastName: "Do")
     * </pre>
     *
     * @return
     */
    public Function<String, FullName> convertToFullName()
    {
        return (String changeWord) -> {
            FullName Name = new FullName();
            String[] Array = changeWord.split(" ", 2);
            Name.setFirstName(Array[0]);
            Name.setLastName(Array[1]);
            return Name;
        };
    }

    /**
     * Question 3:
     * <pre>
     *      Return function to get max int value.
     * </pre>
     *
     * @return
     */
    public BiFunction<Integer, Integer, Integer> max()
    {
        return (Integer firstInt, Integer secInt) -> {
            if (firstInt > secInt)
                return firstInt;
            else return secInt;
        };
    }

    /**
```

```
 * Question 4:
 * <pre>
 *
 * </pre>
 *
 * @return
 */
public Supplier<RuntimeException> runtimeExceptionSupplier()
{
	Supplier<RuntimeException> exception() -> {
		return new RuntimeException();
	};
	return exception;
}

/**
 * Question 5:
 * <pre>
 *		Return function to print given strings. (Use System.out.println())
 * </pre>
 *
 * @return
 */
public Consumer<String> printConsumer()
{
	return (String printingName) -> System.out.println(printingName);
}

/**
 * Question 6:
 * <pre>
 *	Return function to check given int value is greater than 30.
 * </pre>
 *
 * @return
 */
```

```
    public Predicate<Integer> checkOver30()
    {
        return (Integer checkingInt) -> {
            if (checkingInt > 30)
                return true;
            else return false;
        };
    }

}
```