1. Create a functional interface called `MathOperation` which has an abstract method `operation`. `Operation` receives two arguments of `int` and returns an `int`. Write four lambda expressions that implement the `MathOperation` interface: addition, subtraction, multiplication and division. These lambda expressions perform operation on the two `int` arguments as denoted by their names.
   Here is a sample output of the program:

```
10 + 5 = 15
10 - 5 = 5
10 * 5 = 50
10 / 5 = 2
```

2. `java.util.function.Predicate` is a functional interface that can be used as assignment target for lambda expression. It represents an operation that takes a single input and returns a boolean value. The interface has an abstract method called `test` which evaluates the predicate on the given argument. Write an `evaluate` method which receives two arguments: a `List` of integers and a predicate. It then evaluates each element in the `List` against the argument given to the predicate and prints the element if the evaluation returns true. With an array of 10 integers, use this method to print:
   a. All the elements
   b. All the odd elements
   c. All the even elements
   d. All the elements that are greater than 5

   Here is a sample output of the program:

```
Print all numbers:
1
2
3
4
5
6
7
8
9
10
Print odd numbers:
1
3
5
7
9
Print even numbers:
2
4
6
8
10
Print numbers greater than 5:
6
7
8
9
10
```

3. Write a Java program that implements a lambda expression to check if a given number is a perfect square.

4. Write a Java program that implements a lambda expression to check if a given string is a palindrome.