CS209 – LAB4

Key Content

- Understand the concept that passing code to methods with behavior parameterization.
- Understand the necessity of method reference and lambda.
- Learn how to use important functional interface.

Before Exercise

Here is a question from Assignment2 of CS102A 2018Fall.

Sort

Description:

Given an array, you should output the indices of elements according their values from small to large. For example, here is an array [2, 1, 3, 4], element 1 is the smallest one, its index is 1, so the first output is 1, element 2 is the second smallest, its index is 0, so the second output is 0, and so on. The output of array [2,1,3,4] is [1,0,2,3]. If some elements have the same value, according their indices from small to large. For example, the output of array [2,2,2,2] is [0,1,2,3]. When you print the result, you should follow the format according to the **Output** description.

Input:

The first line will be an integer T (1<=T<=20), which is the number of test cases.

For each test case, the first line will be the size of the array N (1 <= N<= 1000).

The second line contains N integers, each integer a_i is in range: [-2^30,2^30].

Output:

The indices in output should be separated by a space.

Sample Input:

1

9

5 3 66 22 9 1 77 88 99

Sample Output:

5 1 0 4 3 2 6 7 8

And here is a reference code:

```java
import java.util.Scanner;

public class Sort {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int total = in.nextInt();

        while (total-- > 0) {
            int n = in.nextInt();
            int array[] = new int[n];
            int index[] = new int[n];
            for (int i = 0; i < n; i++) {
                array[i] = in.nextInt();
                index[i] = i;
            }
            //Bubble sort the index according the value
            int temp;
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n - i - 1; j++) {

                    if (array[index[j]] > array[index[j+1]]){
                        temp = index[j + 1];
                        index[j + 1] = index[j];
                        index[j] = temp;

                    }
                }
            }

            for (int i = 0; i < n; i++) {
                System.out.printf("%d", index[i]);
                if (i != n - 1)
                    System.out.print(" ");
            }
```

```
                System.out.println();
            }
            in.close();
        }
}
```

1. Rewrite above code using Collections.sort().
2. If we want to output the indices of elements according their values from large to small, what we should do?

```
Sample Input:
1
9
5 3 66 22 9 1 77 88 99


Sample Output:
8 7 6 2 3 4 0 1 5
```

3. If add an input argument, 0 means to arrange the element from small to large, 1 means to arrange the element from large to small, what we should do?

```
Sample Input:
2
0 9
5 3 66 22 9 1 77 88 99
1 9
5 3 66 22 9 1 77 88 99


Sample Output:
5 1 0 4 3 2 6 7 8
8 7 6 2 3 4 0 1 5
```

Reference code:

1. Sort1.java
2. Sort2.java
3. Sort3.java

Exercise

1. Given an integer array, please do following tasks:

(1) output all even numbers in the array.

```
Sample Input:
5
5 6 7 8 11
Sample Output:
6 8
```

(2) Output all odd numbers in the array.

```
Sample Input:
5
5 6 7 8 11
Sample Output:
5 7 11
```

(3) Output all prime numbers in the array.

```
Sample Input:
5
5 6 7 8 11
Sample Output:
5 7 11
```

(4) Output all prime numbers and bigger than 5 in the array.

```
Sample Input:
5
5 6 7 8 11
Sample Output:
7 11
```

You should try to use Predicate<T> interface.

2. Give a sequence of integer numbers, output their quadratic sum.

Sample Input:

```
3
1 2 3
```

Sample Output:

```
14
```

You should try to use UnaryOperator<T>, Function<T, R> or BinaryOperator<T>.