

## Домашнє завдання №26\_1

Виконати домашнє завдання №25\_1 повторно за допомогою мови Java без використання готових бібліотечних реалізацій.

*\* акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

*\* коментар: це завдання тотожне до завдання №25\_1, №25\_2, №26\_2 та лабораторної роботи №5; таким чином можна порівняти різні засоби програмування; далі наводиться приклад повністю виконаного завдання; для компіляції і запуску можна використати <https://repl.it/languages/java> або [https://www.tutorialspoint.com/compile\\_java8\\_online.php](https://www.tutorialspoint.com/compile_java8_online.php).*

### Вибір варіанту

Варіант завдання відповідає варіанту домашнього завдання №25_1
--

### Приклад коду

Наведений зразок коду реалізовує завдання для даних, що подані в наступній таблиці:

Дані для прикладу	5, 7, 3, 4, 1, 9, 2, 8, 10, 6
Означення	
<b>final static</b> Integer[] <b>INPUT_DATA</b> = <b>new</b> Integer[]{5, 7, 3, 4, 1, 9, 2, 8, 10, 6};	

Лістинг

```

//// use default package // or use package: // package org.eom.acm;

import java.util.stream.*;
import java.util.*;

public class Main{
    final static Integer[] INPUT_DATA = new Integer[]{5, 7, 3, 4, 1, 9, 2, 8, 10, 6};

    public static List<Integer> quickSort(List<Integer> xs){
        if(xs.size() < 2){
            return xs;
        }

        int x = xs.get(0);

        List<Integer> lesserPart = xs.stream().skip(1).filter(i -> i <=
x).collect(Collectors.toList());

        List<Integer> greaterPart = xs.stream().skip(1).filter(i -> i >
x).collect(Collectors.toList());

        return Stream.concat(
            Stream.concat(quickSort(lesserPart).stream(),Stream.of(x)),

```

```
        quickSort(greaterPart).stream()).collect(Collectors.toList());
    }

    public static void print(List<Integer> xs){
        if(xs.size() == 0){
            return;
        }
        else{
            int x = xs.get(0);
            System.out.print(String.valueOf(x) + " ");
            print(xs.stream().skip(1)
                .collect(Collectors.toList())); // xs.remove(0); print(xs);
        }
    }

    public static void main(String[] args) {
        //System.out.println(Main.quickSort(
        //new ArrayList<Integer>(Arrays.asList(INPUT_DATA))));
        List<Integer> input = new ArrayList<Integer>(Arrays.asList(INPUT_DATA));
        List<Integer> output = Main.quickSort(input);
        System.out.println("input:");
        Main.print(input);
        System.out.println();
        System.out.println("output:");
        Main.print(output);
        System.out.println();
    }
}
```