

Demo - pipeline

```
List<Integer> list = Arrays.asList(3, 4, 5, 10, 7);

Stream<Integer> st1 = list.stream().map(x -> x * 10);
System.out.println(Arrays.toString(st1.toArray()));

int sum = list.stream().reduce(0, (x, y) -> x + y);
System.out.println("Sum = " + sum);

List<Integer> newList = list.stream()
    .filter(x -> x % 2 == 0)
    .map(x -> x * 10)
    .collect(Collectors.toList());
System.out.println(Arrays.toString(newList.toArray()));
```

Exercício resolvido - filter, sorted, map, reduce

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Fazer um programa para ler um conjunto de produtos a partir de um arquivo em formato .csv (suponha que exista pelo menos um produto).

Em seguida mostrar o preço médio dos produtos. Depois, mostrar os nomes, em ordem decrescente, dos produtos que possuem preço inferior ao preço médio.

Veja exemplo na próxima página.

<https://github.com/acenelio/lambda6-java>

Input file:

```
Tv,900.00  
Mouse,50.00  
Tablet,350.50  
HD Case,80.90  
Computer,850.00  
Monitor,290.00
```

Execution:

```
Enter full file path: c:\temp\in.txt  
Average price: 420.23  
Tablet  
Mouse  
Monitor  
HD Case
```

<https://github.com/acenelio/lambda6-java>