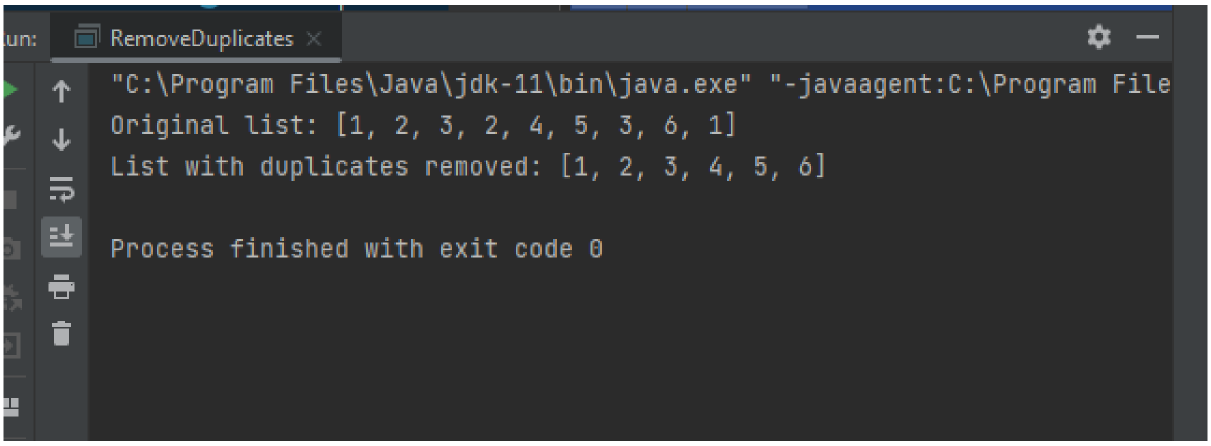
ASSIGNMENT JAVA DAY14

Harshit Kushmakar| 16896

1. Write a program to remove duplicate elements from a List.

```java
package assignment14;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class RemoveDuplicates {
    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<>(Arrays.asList(1, 2, 3, 2,
4, 5, 3, 6, 1));

        // Use a stream to filter out duplicate elements
        List<Integer> uniqueNumbers = numbers.stream()
                .distinct()
                .collect(Collectors.toList());

        System.out.println("Original list: " + numbers);
        System.out.println("List with duplicates removed: " +
uniqueNumbers);
    }
}
```

OUTPUT:

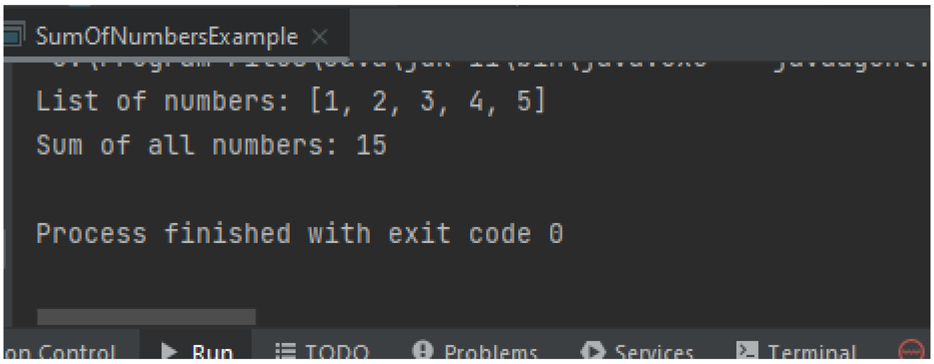


```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program File
Original list: [1, 2, 3, 2, 4, 5, 3, 6, 1]
List with duplicates removed: [1, 2, 3, 4, 5, 6]

Process finished with exit code 0
```

2. Write a program to get sum of all numbers present in a list.

```java
package assignment14;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class SumOfNumbersExample {
    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<>(Arrays.asList(1, 2, 3, 4,
5));

        // Use a stream to get the sum of all the numbers in the list
        int sum = numbers.stream().mapToInt(Integer::intValue).sum();

        System.out.println("List of numbers: " + numbers);
        System.out.println("Sum of all numbers: " + sum);
    }
}
```

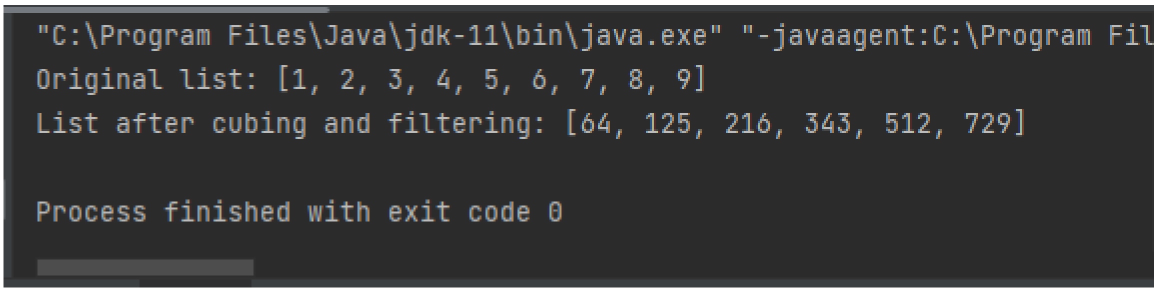
OUTPUT:



```
List of numbers: [1, 2, 3, 4, 5]
Sum of all numbers: 15

Process finished with exit code 0
```

3. Write a program to perform cube on list elements and filter numbers greater than 50.

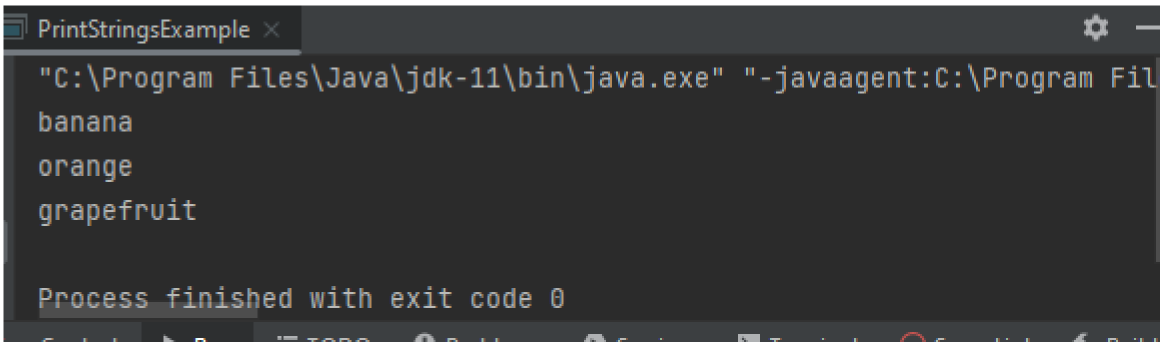
```java
package assignment14;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class CubeAndFilterExample {
    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<>(Arrays.asList(1, 2, 3, 4,
5, 6, 7, 8, 9));

        // Use a stream to cube each number and filter out numbers greater
than 50
        List<Integer> result = numbers.stream()
                .map(n -> n * n * n)
                .filter(n -> n > 50)
                .collect(Collectors.toList());

        System.out.println("Original list: " + numbers);
        System.out.println("List after cubing and filtering: " + result);
    }
}
```



```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program Fil
Original list: [1, 2, 3, 4, 5, 6, 7, 8, 9]
List after cubing and filtering: [64, 125, 216, 343, 512, 729]

Process finished with exit code 0
```

4. Write a program to print strings whose length is greater than 5 in list.

```
package assignment14;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class PrintStringsExample {
    public static void main(String[] args) {
        List<String> strings = new ArrayList<>(Arrays.asList("apple",
"banana", "orange", "pear", "kiwi", "grapefruit"));

        // Use a stream to filter out strings with length greater than 5
and print them
        strings.stream()
                .filter(s -> s.length() > 5)
                .forEach(System.out::println);
    }
}
```

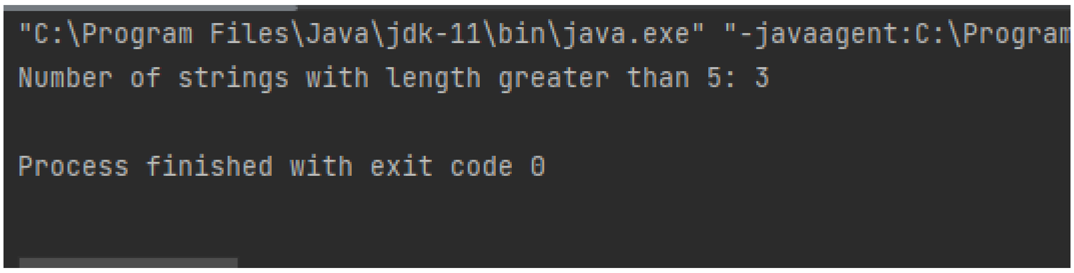


```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program Fil
banana
orange
grapefruit

Process finished with exit code 0
```

5. Write a program to count strings whose length is greater than 5 in list.

```
package assignment14;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class CountStringsExample {
```

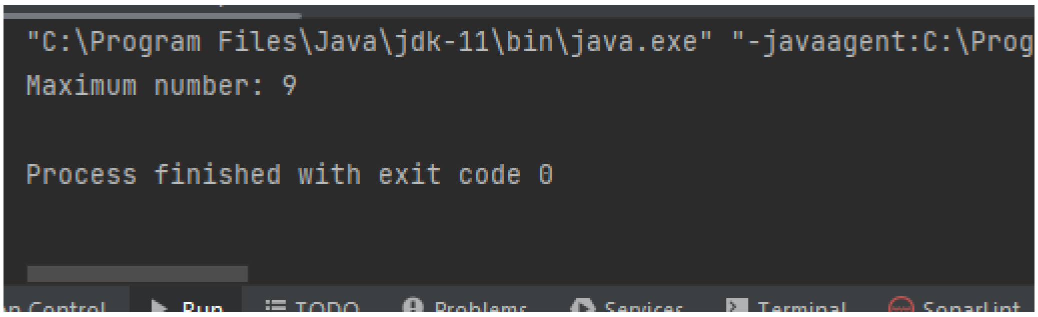
```java
    public static void main(String[] args) {
        List<String> strings = new ArrayList<>(Arrays.asList("apple",
"banana", "orange", "pear", "kiwi", "grapefruit"));

        // Use a stream to filter out strings with length greater than 5
and count them
        long count = strings.stream()
                .filter(s -> s.length() > 5)
                .count();

        System.out.println("Number of strings with length greater than 5: "
+ count);
    }
}
```



```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program
Number of strings with length greater than 5: 3

Process finished with exit code 0
```

6. Write a program to find the maximum number of a list.

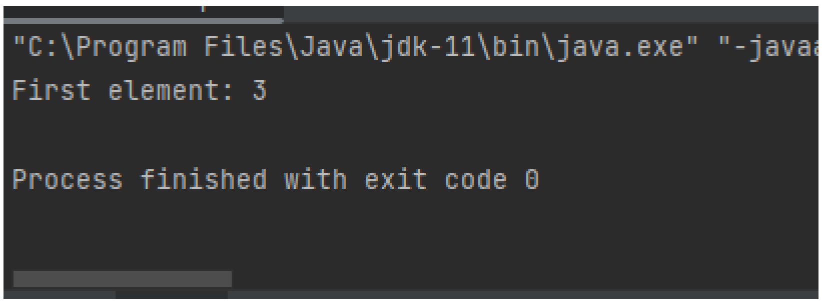
```java
package assignment14;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class MaxNumberExample {
    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<>(Arrays.asList(3, 5, 1, 9,
2, 7, 4));

        // Use a stream to find the maximum number
        int max = numbers.stream()
                .mapToInt(Integer::intValue)
                .max()
                .orElseThrow();

        System.out.println("Maximum number: " + max);
    }
}
```



```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Prog
Maximum number: 9

Process finished with exit code 0
```

7. Given the list of integers, find the first element of the list using Stream functions.

```
package assignment14;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class FirstElementExample {
    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<>(Arrays.asList(3, 5, 1, 9,
2, 7, 4));

        // Use a stream to find the first element
        int first = numbers.stream()
                .findFirst()
                .orElseThrow();

        System.out.println("First element: " + first);
    }
```

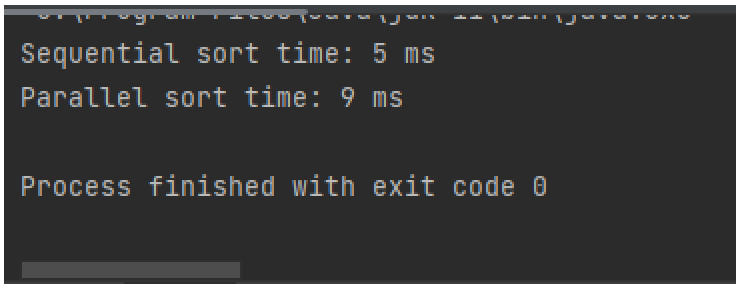


```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-java
First element: 3

Process finished with exit code 0
```

8. Generate 10000 random Customer objects (class created in Java Day-3 assignment) and store them in an array. Use Sequential Stream and Parallel Stream to store the objects in an array. Find out the performance of both type of streams and print the result.

```
package assignment14;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Random;

public class CustomerSortExample {
    public static void main(String[] args) {
        // Generate 10000 random customers
        List<Customer> customers = new ArrayList<>();
        Random random = new Random();
        for (int i = 0; i < 10000; i++) {
            String name = "Customer " + i;
            int age = random.nextInt(100);
            customers.add(new Customer(name, age));
        }

        // Convert the list to an array
        Customer[] customerArray = customers.toArray(new Customer[0]);

        // Use a sequential stream to sort the array
        long startTime = System.currentTimeMillis();
        Arrays.sort(customerArray, (c1, c2) ->
c1.getName().compareTo(c2.getName()));
        long endTime = System.currentTimeMillis();
        long sequentialTime = endTime - startTime;
        System.out.println("Sequential sort time: " + sequentialTime + "
ms");

        // Use a parallel stream to sort the array
        startTime = System.currentTimeMillis();
        Arrays.parallelSort(customerArray, (c1, c2) ->
c1.getName().compareTo(c2.getName()));
        endTime = System.currentTimeMillis();
        long parallelTime = endTime - startTime;
        System.out.println("Parallel sort time: " + parallelTime + " ms");
    }
}

class Customer {
    private String name;
    private int age;

    public Customer(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
```

OUTPUT:



```
[illegible]
Sequential sort time: 5 ms
Parallel sort time: 9 ms

Process finished with exit code 0
```