

## Assignment No:-52

Name:-Suryawanshi Sangramsingh Sambhaji

Batch: - Delta - DCA (Java) 2024     Date:-24/7/2024

**Q1. Write a program to create a stream from a list of integers and print each element.**

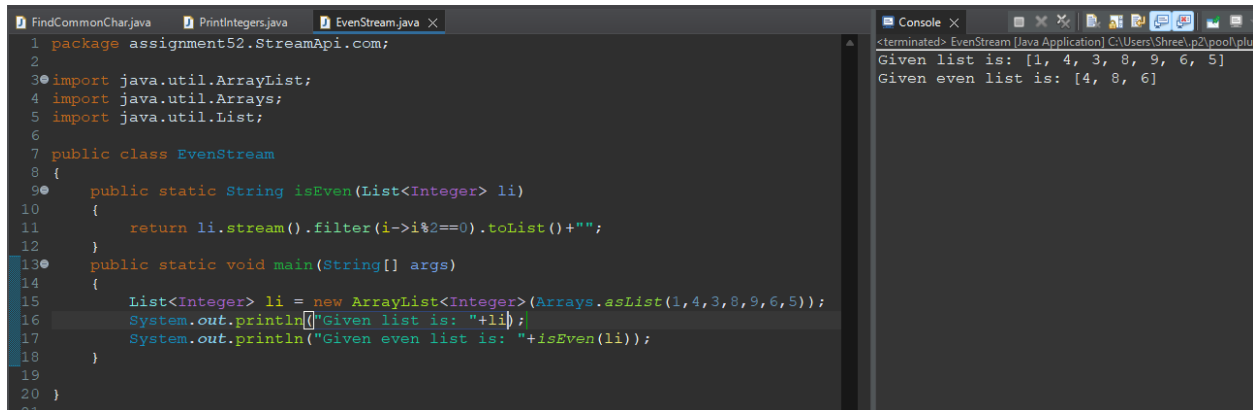


```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class PrintIntegers
8 {
9     public static String isInt(List<Integer> li)
10    {
11        return li.stream().toList()+" ";
12    }
13    public static void main(String[] args)
14    {
15        List<Integer> li = new ArrayList<Integer>(Arrays.asList(1,4,3,8,9,6,5));
16        System.out.println("Given list is: "+isInt(li));
17    }
18 }
19
20
```

Console

<terminated> PrintIntegers [Java Application] C:\Users\Shree\p2\pool\plugi  
Given list is: [1, 4, 3, 8, 9, 6, 5]

**Q2. Implement a program to filter even numbers from a list using the Stream API.**

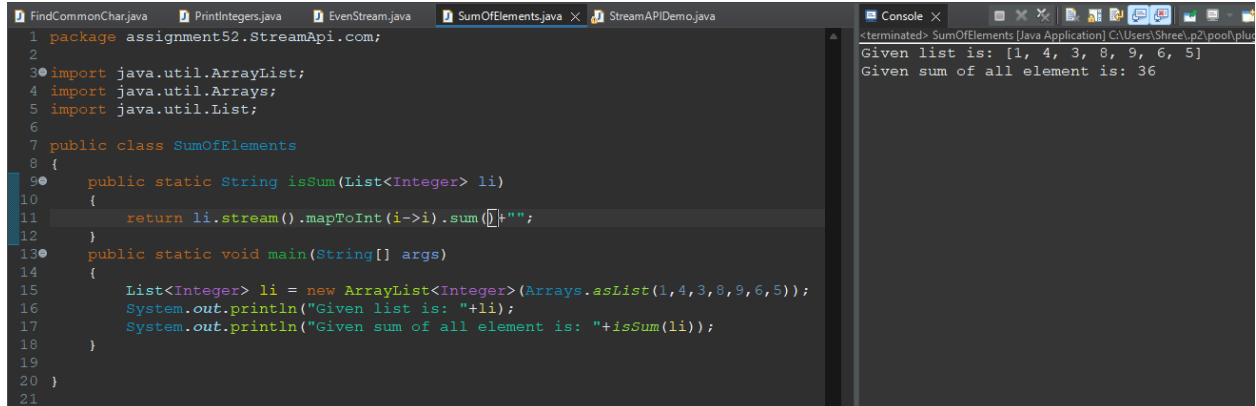


```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class EvenStream
8 {
9     public static String isEven(List<Integer> li)
10    {
11        return li.stream().filter(i->i%2==0).toList()+" ";
12    }
13    public static void main(String[] args)
14    {
15        List<Integer> li = new ArrayList<Integer>(Arrays.asList(1,4,3,8,9,6,5));
16        System.out.println("Given list is: "+li);
17        System.out.println("Given even list is: "+isEven(li));
18    }
19 }
20
21
```

Console

<terminated> EvenStream [Java Application] C:\Users\Shree\p2\pool\plugi  
Given list is: [1, 4, 3, 8, 9, 6, 5]  
Given even list is: [4, 8, 6]

**Q3. Write a program to find the sum of all elements in a list using streams.**



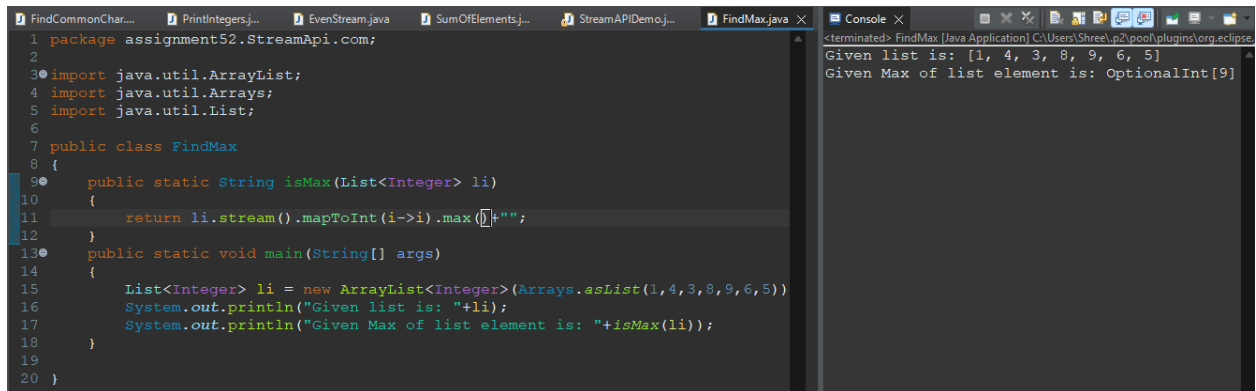
The screenshot shows an IDE with a Java file named `SumOfElements.java`. The code defines a class `SumOfElements` with a static method `isSum` that takes a `List<Integer>` and returns a string representing the sum of its elements using streams. The `main` method creates a list `[1, 4, 3, 8, 9, 6, 5]` and prints the list and the sum.

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class SumOfElements
8 {
9     public static String isSum(List<Integer> li)
10    {
11        return li.stream().mapToInt(i->i).sum()+" ";
12    }
13    public static void main(String[] args)
14    {
15        List<Integer> li = new ArrayList<Integer>(Arrays.asList(1,4,3,8,9,6,5));
16        System.out.println("Given list is: "+li);
17        System.out.println("Given sum of all element is: "+isSum(li));
18    }
19 }
20
21
```

The console output shows:

```
<terminated> SumOfElements [Java Application] C:\Users\Shree\p2\pool\plu
Given list is: [1, 4, 3, 8, 9, 6, 5]
Given sum of all element is: 36
```

**Q4. Implement a program to find the maximum element from a list using streams.**



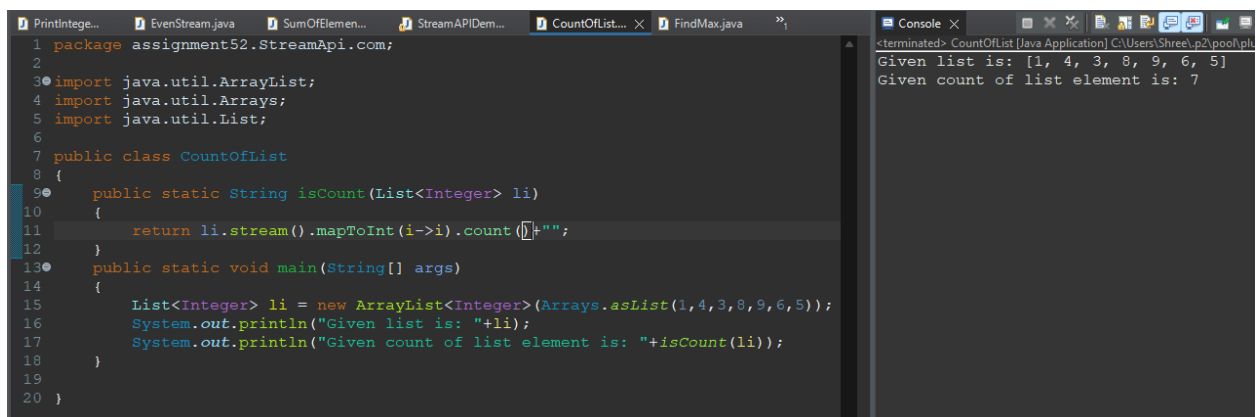
The screenshot shows an IDE with a Java file named `FindMax.java`. The code defines a class `FindMax` with a static method `isMax` that takes a `List<Integer>` and returns a string representing the maximum element using streams. The `main` method creates a list `[1, 4, 3, 8, 9, 6, 5]` and prints the list and the maximum element.

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class FindMax
8 {
9     public static String isMax(List<Integer> li)
10    {
11        return li.stream().mapToInt(i->i).max().get()+" ";
12    }
13    public static void main(String[] args)
14    {
15        List<Integer> li = new ArrayList<Integer>(Arrays.asList(1,4,3,8,9,6,5));
16        System.out.println("Given list is: "+li);
17        System.out.println("Given Max of list element is: "+isMax(li));
18    }
19 }
20
```

The console output shows:

```
<terminated> FindMax [Java Application] C:\Users\Shree\p2\pool\plugins\org.eclipse
Given list is: [1, 4, 3, 8, 9, 6, 5]
Given Max of list element is: OptionalInt[9]
```

**Q5. Write a program to count the number of elements in a list using streams.**



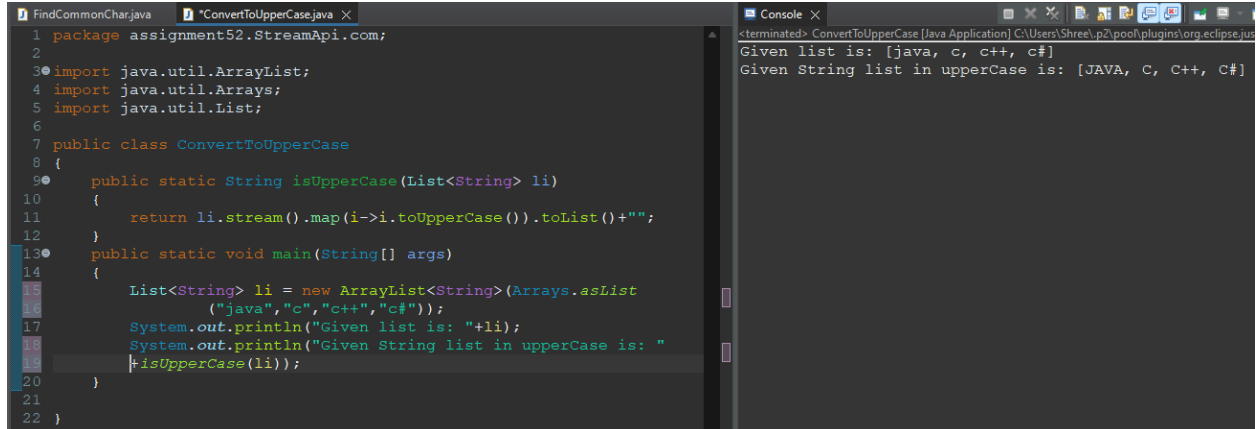
The screenshot shows an IDE with a Java file named `CountOfList.java`. The code defines a class `CountOfList` with a static method `isCount` that takes a `List<Integer>` and returns a string representing the count of elements using streams. The `main` method creates a list `[1, 4, 3, 8, 9, 6, 5]` and prints the list and the count.

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class CountOfList
8 {
9     public static String isCount(List<Integer> li)
10    {
11        return li.stream().mapToInt(i->i).count()+" ";
12    }
13    public static void main(String[] args)
14    {
15        List<Integer> li = new ArrayList<Integer>(Arrays.asList(1,4,3,8,9,6,5));
16        System.out.println("Given list is: "+li);
17        System.out.println("Given count of list element is: "+isCount(li));
18    }
19 }
20
```

The console output shows:

```
<terminated> CountOfList [Java Application] C:\Users\Shree\p2\pool\plu
Given list is: [1, 4, 3, 8, 9, 6, 5]
Given count of list element is: 7
```

**Q6. Implement a program to convert a list of strings to uppercase using streams.**



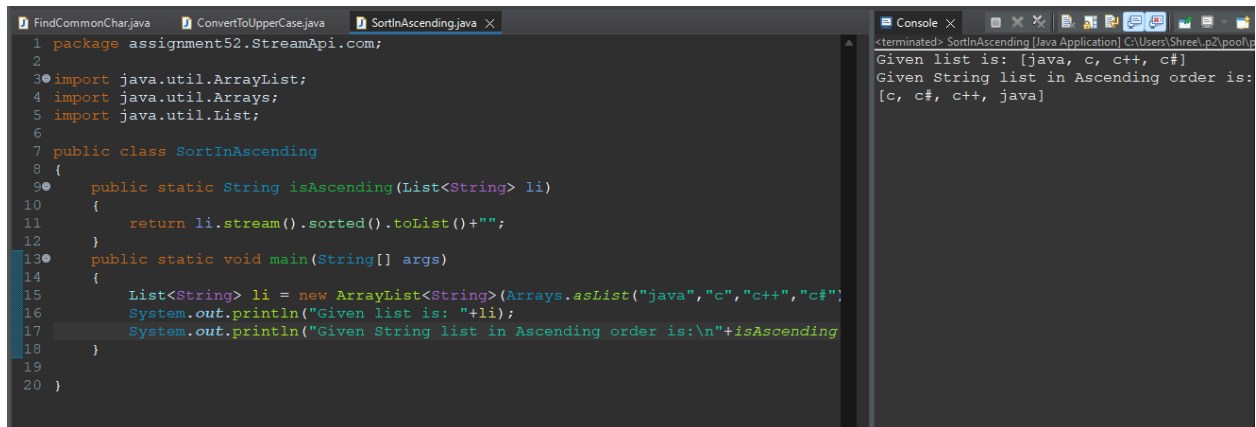
The screenshot shows an IDE with two tabs: 'FindCommonChar.java' and 'ConvertToUpperCase.java'. The 'ConvertToUpperCase.java' tab is active, displaying the following code:

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class ConvertToUpperCase
8 {
9     public static String isUpperCase(List<String> li)
10    {
11        return li.stream().map(i->i.toUpperCase()).toList()+"";
12    }
13    public static void main(String[] args)
14    {
15        List<String> li = new ArrayList<String>(Arrays.asList
16            ("java","c","c++","c#"));
17        System.out.println("Given list is: "+li);
18        System.out.println("Given String list in upperCase is: "
19            +isUpperCase(li));
20    }
21 }
22 }
```

The console output on the right shows:

```
<terminated> ConvertToUpperCase [Java Application] C:\Users\Shree\p2\pool\plugins\org.eclipse.jdt
Given list is: [java, c, c++, c#]
Given String list in upperCase is: [JAVA, C, C++, C#]
```

**Q7. Write a program to sort a list of strings in ascending order using streams.**



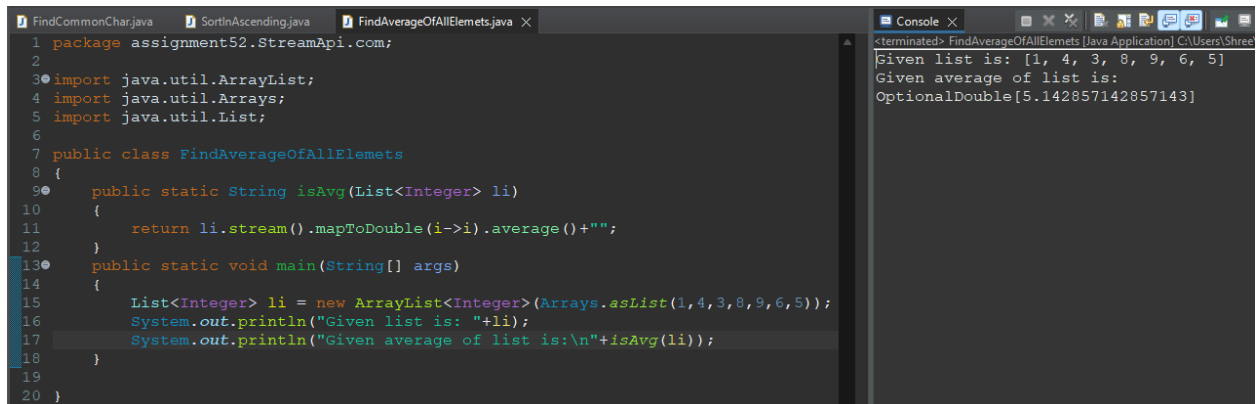
The screenshot shows an IDE with three tabs: 'FindCommonChar.java', 'ConvertToUpperCase.java', and 'SortInAscending.java'. The 'SortInAscending.java' tab is active, displaying the following code:

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class SortInAscending
8 {
9     public static String isAscending(List<String> li)
10    {
11        return li.stream().sorted().toList()+"";
12    }
13    public static void main(String[] args)
14    {
15        List<String> li = new ArrayList<String>(Arrays.asList("java","c","c++","c#"));
16        System.out.println("Given list is: "+li);
17        System.out.println("Given String list in Ascending order is:\n"+isAscending
18            (li));
19    }
20 }
```

The console output on the right shows:

```
<terminated> SortInAscending [Java Application] C:\Users\Shree\p2\pool\p
Given list is: [java, c, c++, c#]
Given String list in Ascending order is:
[c, c#, c++, java]
```

**Q8. Implement a program to find the average of all numbers in a list using streams.**



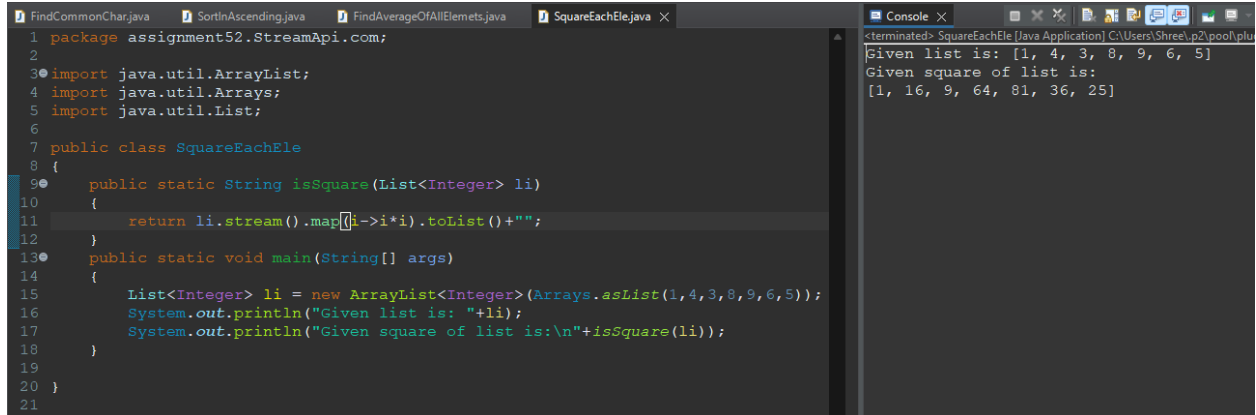
The screenshot shows an IDE with three tabs: 'FindCommonChar.java', 'SortInAscending.java', and 'FindAverageOfAllElemets.java'. The 'FindAverageOfAllElemets.java' tab is active, displaying the following code:

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class FindAverageOfAllElemets
8 {
9     public static String isAvg(List<Integer> li)
10    {
11        return li.stream().mapToDouble(i->i).average()+"";
12    }
13    public static void main(String[] args)
14    {
15        List<Integer> li = new ArrayList<Integer>(Arrays.asList(1,4,3,8,9,6,5));
16        System.out.println("Given list is: "+li);
17        System.out.println("Given average of list is:\n"+isAvg(li));
18    }
19 }
20 }
```

The console output on the right shows:

```
<terminated> FindAverageOfAllElemets [Java Application] C:\Users\Shree
Given list is: [1, 4, 3, 8, 9, 6, 5]
Given average of list is:
OptionalDouble[5.142857142857143]
```

**Q9. Write a program to find the square of each element in a list using streams.**



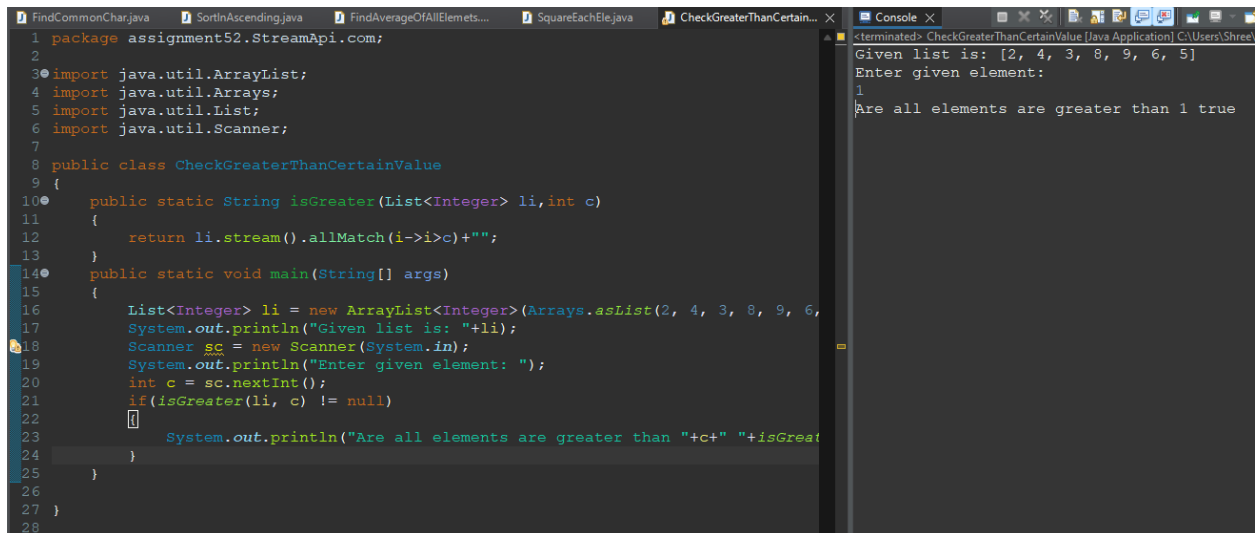
The screenshot shows an IDE with a Java file named `SquareEachEle.java` and a console window. The code defines a class `SquareEachEle` with a static method `isSquare` that takes a `List<Integer>` and returns a string of squared elements. The `main` method creates a list `[1, 4, 3, 8, 9, 6, 5]` and prints the original list and the squared list.

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class SquareEachEle
8 {
9     public static String isSquare(List<Integer> li)
10    {
11        return li.stream().map(i->i*i).toList()+" ";
12    }
13    public static void main(String[] args)
14    {
15        List<Integer> li = new ArrayList<Integer>(Arrays.asList(1,4,3,8,9,6,5));
16        System.out.println("Given list is: "+li);
17        System.out.println("Given square of list is:\n"+isSquare(li));
18    }
19 }
20 }
21
```

Console output:

```
<terminated> SquareEachEle [Java Application] C:\Users\Shree\p2\pool\plu
Given list is: [1, 4, 3, 8, 9, 6, 5]
Given square of list is:
[1, 16, 9, 64, 81, 36, 25]
```

**Q10. Implement a program to check if all elements in a list are greater than a certain value using streams.**



The screenshot shows an IDE with a Java file named `CheckGreaterThenCertainValue.java` and a console window. The code defines a class `CheckGreaterThenCertainValue` with a static method `isGreater` that takes a `List<Integer>` and an integer `c`, and returns a string indicating if all elements are greater than `c`. The `main` method creates a list `[2, 4, 3, 8, 9, 6, 5]`, prompts the user for a value `c` (1), and prints the result.

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6 import java.util.Scanner;
7
8 public class CheckGreaterThenCertainValue
9 {
10    public static String isGreater(List<Integer> li,int c)
11    {
12        return li.stream().allMatch(i->i>c)+" ";
13    }
14    public static void main(String[] args)
15    {
16        List<Integer> li = new ArrayList<Integer>(Arrays.asList(2, 4, 3, 8, 9, 6, 5));
17        System.out.println("Given list is: "+li);
18        Scanner sc = new Scanner(System.in);
19        System.out.println("Enter given element: ");
20        int c = sc.nextInt();
21        if(isGreater(li, c) != null)
22        {
23            System.out.println("Are all elements are greater than "+c+" "+isGreater(li, c));
24        }
25    }
26 }
27 }
28
```

Console output:

```
<terminated> CheckGreaterThenCertainValue [Java Application] C:\Users\Shree
Given list is: [2, 4, 3, 8, 9, 6, 5]
Enter given element:
1
Are all elements are greater than 1 true
```

**Q11. Write a program to find the first occurrence of a given element in a list using streams.**

```
FindCommonChar.java  SortInAscending.java  SortByLength.java  SortThierElementsTheirLength.java  FirstOccurrenceOfList.java  Console
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
11
12 public class FirstOccurrenceOfList
13 {
14     public static Optional<String> findFirstRepeating(List<String> list)
15     {
16         Set<String> seen = new LinkedHashSet<>();
17
18         return list.stream()
19             .filter(s -> !seen.add(s))
20             .findFirst();
21     }
22     public static void main(String[] args)
23     {
24         List<String> list = new ArrayList<>(Arrays.asList("java", "c", "c", "c#", "ja
25         System.out.println("Given list is: " + list);
26
27         Optional<String> firstRepeating = findFirstRepeating(list);
28
29         if (firstRepeating.isPresent())
30         {
31             System.out.println("First repeating string is: " + firstRepeating.get());
32         }
33         else
34         {
35             System.out.println("No repeating string found.");
36         }
37     }
38 }
```

```
<terminated> FirstOccurrenceOfList [Java Application] C:\Users\Shree\p2\pool
Given list is: [java, c, c, c#, java]
First repeating string is: c
```

**Q12. Implement a program to find the distinct elements in a list using streams.**

```
FindCommonChar.java  SortInAscending.java  FirstOccurrenceOfList.java  DistinctElemets.java  Console
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class DistinctElemets
8 {
9     public static String isDistinct(List<Integer> li)
10     {
11         return li.stream().distinct().toList()+"";
12     }
13     public static void main(String[] args)
14     {
15         List<Integer> li = new ArrayList<Integer>(Arrays.asList(2,4,3,4,9,6,1));
16         System.out.println("Given list is: "+li);
17         System.out.println("Given remove distinct element is:\n"+isDistinct(li));
18     }
19
20 }
21
```

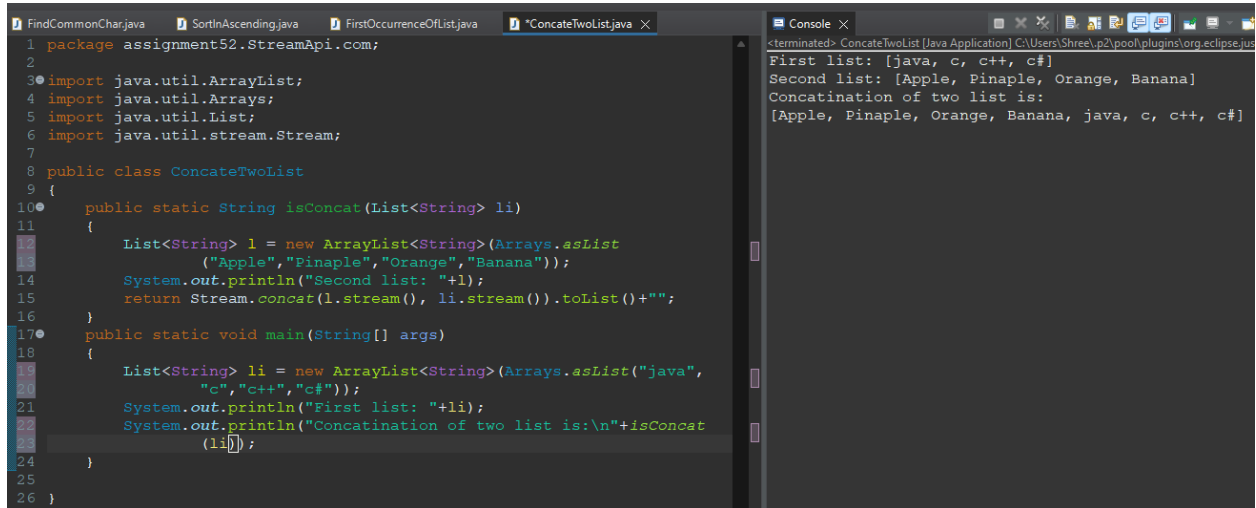
```
<terminated> DistinctElemets [Java Application] C:\Users\Shree\p2\pool
Given list is: [2, 4, 3, 4, 9, 6, 1]
Given remove distinct element is:
[2, 4, 3, 9, 6, 1]
```

**Q13. Write a program to remove null values from a list using streams.**

```
FindCommonChar.java  SortInAscending.java  FirstOccurrenceOfList.java  DistinctElemets.java  RemoveNull.java  Console
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class RemoveNull
8 {
9     public static String isNull(List<Integer> li)
10     {
11         return li.stream().filter(i->i!=0).toList()+"";
12     }
13     public static void main(String[] args)
14     {
15         List<Integer> li = new ArrayList<Integer>(Arrays.asList(2,4,3,0,9,6,1));
16         System.out.println("Given list is: "+li);
17         System.out.println("Given removed null element is:\n"+isNull(li));
18     }
19
20 }
```

```
<terminated> RemoveNull [Java Application] C:\Users\Shree\p2\pool
Given list is: [2, 4, 3, 0, 9, 6, 1]
Given removed null element is:
[2, 4, 3, 9, 6, 1]
```

**Q14. Implement a program to concatenate two lists using streams.**



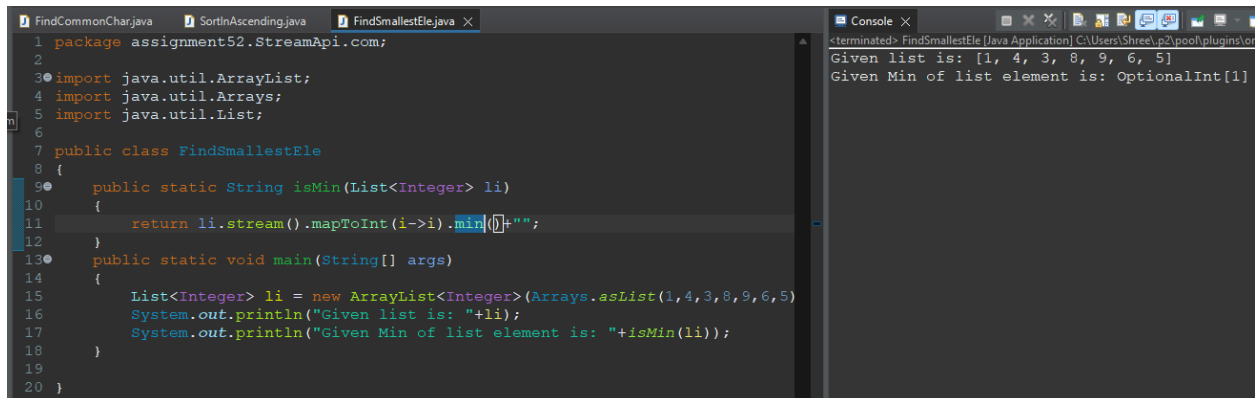
The screenshot shows an IDE with a Java file named `ConcatTwoList.java` and a console window. The code defines a class `ConcatTwoList` with a static method `isConcat` that takes a `List<String>` and returns a concatenated string. The `main` method creates two lists: one with fruits and one with programming languages, and prints their concatenation using streams.

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6 import java.util.stream.Stream;
7
8 public class ConcatTwoList
9 {
10     public static String isConcat(List<String> li)
11     {
12         List<String> l = new ArrayList<String>(Arrays.asList
13             ("Apple","Pinapple","Orange","Banana"));
14         System.out.println("Second list: "+l);
15         return Stream.concat(l.stream(), li.stream()).toList()+"";
16     }
17     public static void main(String[] args)
18     {
19         List<String> li = new ArrayList<String>(Arrays.asList("java",
20             "c","c++","c#"));
21         System.out.println("First list: "+li);
22         System.out.println("Concatination of two list is:\n"+isConcat
23             (li));
24     }
25 }
26 }
```

Console output:

```
<terminated> ConcatTwoList [Java Application] C:\Users\Shree\p2\pool\plugins\org.eclipse.j...
First list: [java, c, c++, c#]
Second list: [Apple, Pinapple, Orange, Banana]
Concatination of two list is:
[Apple, Pinapple, Orange, Banana, java, c, c++, c#]
```

**Q15. Write a program to find the second smallest element in a list using streams.**



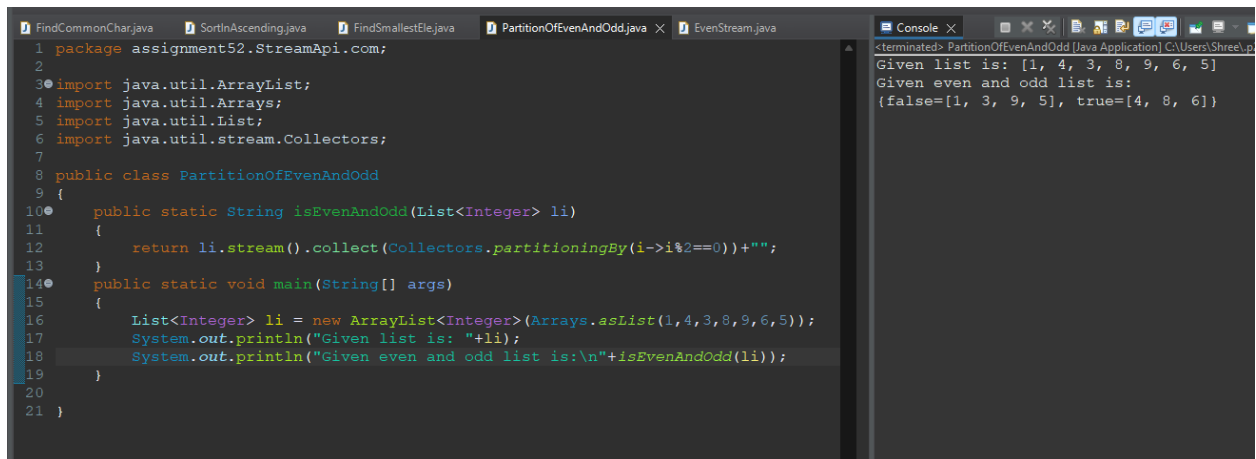
The screenshot shows an IDE with a Java file named `FindSmallestEle.java` and a console window. The code defines a class `FindSmallestEle` with a static method `isMin` that takes a `List<Integer>` and returns the second smallest element. The `main` method creates a list of integers and prints the result using streams.

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class FindSmallestEle
8 {
9     public static String isMin(List<Integer> li)
10     {
11         return li.stream().mapToInt(i->i).min(0)+"";
12     }
13     public static void main(String[] args)
14     {
15         List<Integer> li = new ArrayList<Integer>(Arrays.asList(1,4,3,8,9,6,5));
16         System.out.println("Given list is: "+li);
17         System.out.println("Given Min of list element is: "+isMin(li));
18     }
19 }
20 }
```

Console output:

```
<terminated> FindSmallestEle [Java Application] C:\Users\Shree\p2\pool\plugins\or...
Given list is: [1, 4, 3, 8, 9, 6, 5]
Given Min of list element is: OptionalInt[1]
```

**Q16. Implement a program to partition a list into even and odd numbers using streams.**



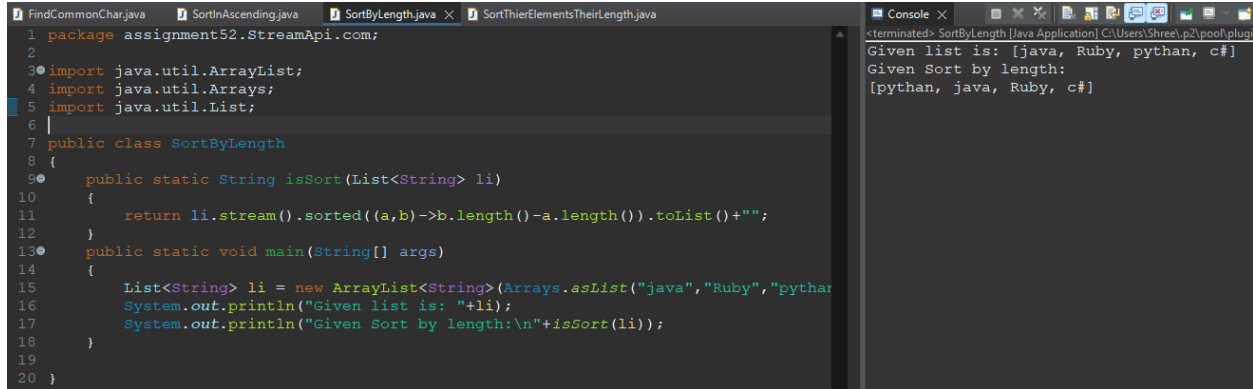
The screenshot shows an IDE with a Java file named `PartitionOfEvenAndOdd.java` and a console window. The code defines a class `PartitionOfEvenAndOdd` with a static method `isEvenAndOdd` that takes a `List<Integer>` and returns a partitioned list. The `main` method creates a list of integers and prints the result using streams.

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6 import java.util.stream.Collectors;
7
8 public class PartitionOfEvenAndOdd
9 {
10     public static String isEvenAndOdd(List<Integer> li)
11     {
12         return li.stream().collect(Collectors.partitioningBy(i->i%2==0))+"";
13     }
14     public static void main(String[] args)
15     {
16         List<Integer> li = new ArrayList<Integer>(Arrays.asList(1,4,3,8,9,6,5));
17         System.out.println("Given list is: "+li);
18         System.out.println("Given even and odd list is:\n"+isEvenAndOdd(li));
19     }
20 }
21 }
```

Console output:

```
<terminated> PartitionOfEvenAndOdd [Java Application] C:\Users\Shree\p...
Given list is: [1, 4, 3, 8, 9, 6, 5]
Given even and odd list is:
{false=[1, 3, 9, 5], true=[4, 8, 6]}
```

**Q17. Write a program to group elements of a list by their lengths using streams.**

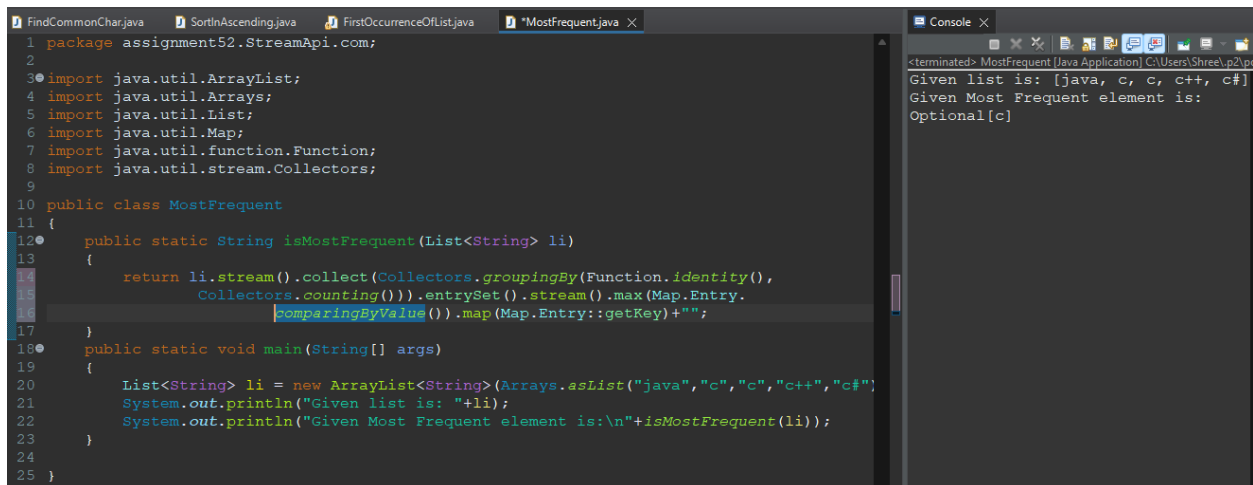


```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class SortByLength
8 {
9     public static String isSort(List<String> li)
10    {
11        return li.stream().sorted((a,b)->b.length()-a.length()).toList()+"";
12    }
13    public static void main(String[] args)
14    {
15        List<String> li = new ArrayList<String>(Arrays.asList("java","Ruby","pythan","c#"));
16        System.out.println("Given list is: "+li);
17        System.out.println("Given Sort by length:\n"+isSort(li));
18    }
19 }
20 }
```

Console

```
<terminated> SortByLength [Java Application] C:\Users\Shree\p2\pool\plugi
Given list is: [java, Ruby, pythan, c#]
Given Sort by length:
[pythan, java, Ruby, c#]
```

**Q18. Implement a program to find the most frequently occurring element in a list using streams.**

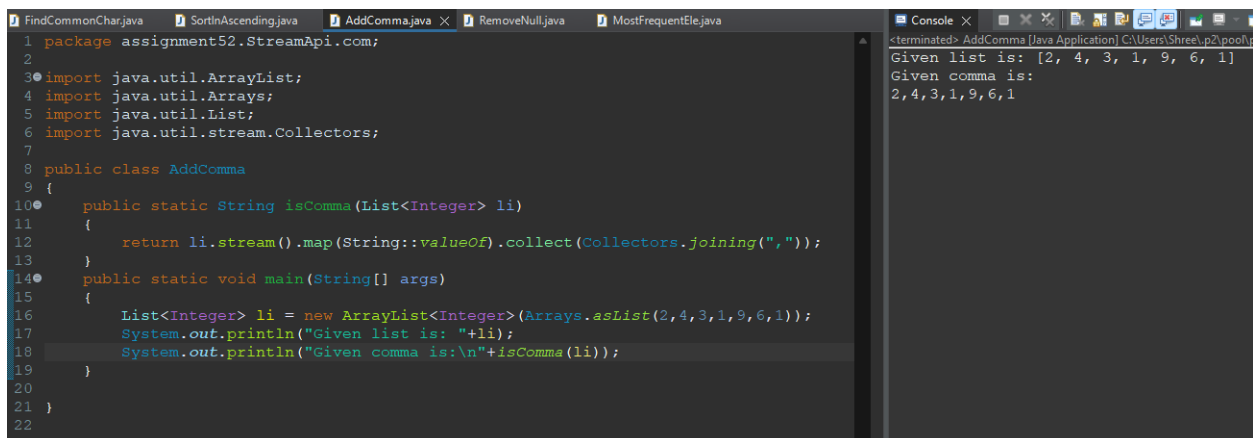


```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6 import java.util.Map;
7 import java.util.function.Function;
8 import java.util.stream.Collectors;
9
10 public class MostFrequent
11 {
12     public static String isMostFrequent(List<String> li)
13     {
14         return li.stream().collect(Collectors.groupingBy(Function.identity(),
15             Collectors.counting())).entrySet().stream().max(Map.Entry.
16             comparingByValue()).map(Map.Entry::getKey)+"";
17     }
18     public static void main(String[] args)
19     {
20         List<String> li = new ArrayList<String>(Arrays.asList("java","c","c","c++","c#"));
21         System.out.println("Given list is: "+li);
22         System.out.println("Given Most Frequent element is:\n"+isMostFrequent(li));
23     }
24 }
25 }
```

Console

```
<terminated> MostFrequent [Java Application] C:\Users\Shree\p2\pool\p
Given list is: [java, c, c, c++, c#]
Given Most Frequent element is:
Optional[c]
```

**Q19. Write a program to convert a list of integers to a comma-separated string using streams.**



```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6 import java.util.stream.Collectors;
7
8 public class AddComma
9 {
10     public static String isComma(List<Integer> li)
11     {
12         return li.stream().map(String::valueOf).collect(Collectors.joining(","));
13     }
14     public static void main(String[] args)
15     {
16         List<Integer> li = new ArrayList<Integer>(Arrays.asList(2,4,3,1,9,6,1));
17         System.out.println("Given list is: "+li);
18         System.out.println("Given comma is:\n"+isComma(li));
19     }
20 }
21 }
22 }
```

Console

```
<terminated> AddComma [Java Application] C:\Users\Shree\p2\pool\p
Given list is: [2, 4, 3, 1, 9, 6, 1]
Given comma is:
2,4,3,1,9,6,1
```

**Q20. Implement a program to find the longest string in a list using streams.**

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class LongestString
8 {
9     public static String isLong(List<String> li)
10    {
11        return li.stream().max((a,b)->Integer.compare
12            (a.length(), b.length()))+"";
13    }
14    public static void main(String[] args)
15    {
16        List<String> li = new ArrayList<String>(Arrays.asList("java","c",
17            "programming","c++","c#"));
18        System.out.println("Given list is: "+li);
19        System.out.println("Given longest String is:\n"+isLong(li));
20    }
21 }
22
23
```

```
<terminated> LongestString [Java Application] C:\Users\Shree\p2\pool\plugins\org.eclipse
Given list is: [java, c, programming, c++, c#]
Given longest String is:
Optional[programming]
```

**Q21. Write a program to remove duplicate elements from a list using streams.**

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7 public class DistinctElemets
8 {
9     public static String isDistinct(List<Integer> li)
10    {
11        return li.stream().distinct().toList()+"";
12    }
13    public static void main(String[] args)
14    {
15        List<Integer> li = new ArrayList<Integer>(Arrays.asList(2,4,3,4,9,6,1));
16        System.out.println("Given list is: "+li);
17        System.out.println("Given remove distinct element is:\n"+isDistinct(li));
18    }
19 }
20
21
```

```
<terminated> DistinctElemets [Java Application] C:\Users\Shree\p2\pool\pl
Given list is: [2, 4, 3, 4, 9, 6, 1]
Given remove distinct element is:
[2, 4, 3, 9, 6, 1]
```

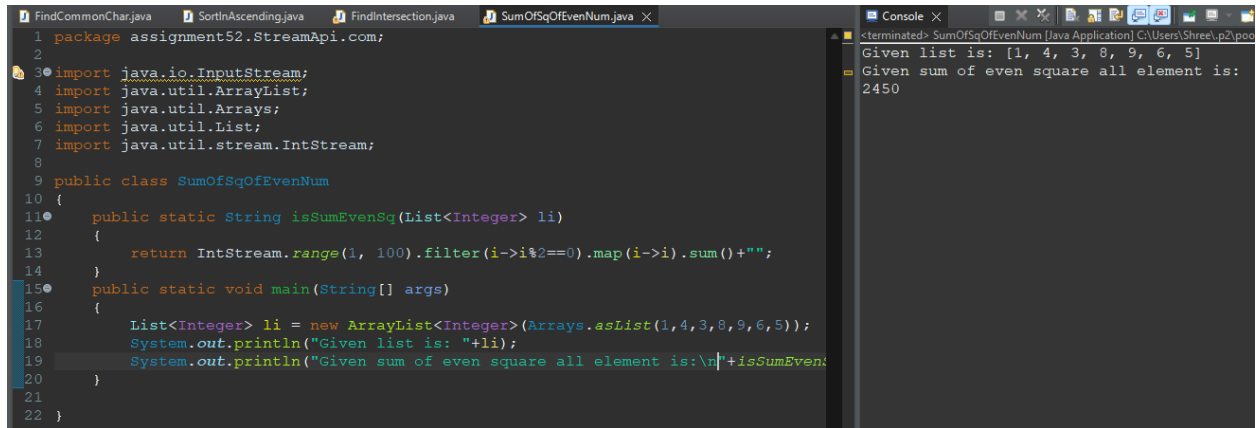
**Q22. Implement a program to find the intersection of two lists using streams.**

```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6 import java.util.stream.Collectors;
7
8 public class FindIntersection
9 {
10    public static String isInter(List<Integer> li, List<Integer> li1)
11    {
12        List let = new ArrayList(li);
13        List let1 = new ArrayList(li1);
14        return let.stream().filter(let1::contains).collect(Collectors.toList())+"
15    }
16    public static void main(String[] args)
17    {
18        List<Integer> li = new ArrayList<Integer>(Arrays.asList(1,4,3,8,9,6,5));
19        List<Integer> list2 = new ArrayList<>(Arrays.asList(3, 6, 9, 11));
20        System.out.println("Given list 1 is: "+li);
21        System.out.println("Given list 2 is: "+list2);
22        System.out.println("Given intersection element is: "+isInter(li, list2));
23    }
24 }
25
```

```
<terminated> FindIntersection [Java Application] C:\Users\Shree\p2\pool\pl
Given list 1 is: [1, 4, 3, 8, 9, 6, 5]
Given list 2 is: [3, 6, 9, 11]
Given intersection element is: [3, 9, 6]
```



**Q23. Write a program to create a stream of numbers from 1 to 100 and find the sum of squares of all even numbers.**



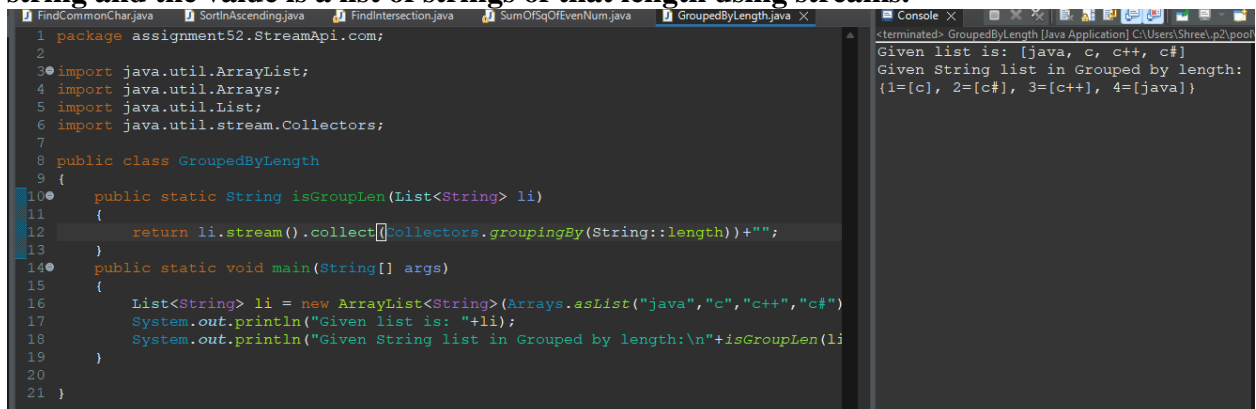
```
1 package assignment52.StreamApi.com;
2
3 import java.io.InputStream;
4 import java.util.ArrayList;
5 import java.util.Arrays;
6 import java.util.List;
7 import java.util.stream.IntStream;
8
9 public class SumOfSqOfEvenNum
10 {
11     public static String isSumEvenSq(List<Integer> li)
12     {
13         return IntStream.range(1, 100).filter(i->i%2==0).map(i->i).sum()+"";
14     }
15     public static void main(String[] args)
16     {
17         List<Integer> li = new ArrayList<Integer>(Arrays.asList(1,4,3,8,9,6,5));
18         System.out.println("Given list is: "+li);
19         System.out.println("Given sum of even square all element is:\n"+isSumEvenSq(li));
20     }
21 }
22 }
```

Console

```
<terminated> SumOfSqOfEvenNum [Java Application] C:\Users\Shree\p2\pool
Given list is: [1, 4, 3, 8, 9, 6, 5]
Given sum of even square all element is:
2450
```

**Q24. Implement a program to find the cumulative sum of elements in a list using streams.**

**Q25. Write a program to convert a list of strings to a map where the key is the length of the string and the value is a list of strings of that length using streams.**



```
1 package assignment52.StreamApi.com;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6 import java.util.stream.Collectors;
7
8 public class GroupedByLength
9 {
10     public static String isGroupLen(List<String> li)
11     {
12         return li.stream().collect(Collectors.groupingBy(String::length))+"";
13     }
14     public static void main(String[] args)
15     {
16         List<String> li = new ArrayList<String>(Arrays.asList("java","c","c++","c#"));
17         System.out.println("Given list is: "+li);
18         System.out.println("Given String list in Grouped by length:\n"+isGroupLen(li));
19     }
20 }
21 }
```

Console

```
<terminated> GroupedByLength [Java Application] C:\Users\Shree\p2\pool
Given list is: [java, c, c++, c#]
Given String list in Grouped by length:
{1=[c], 2=[c#], 3=[c++], 4=[java]}
```