```
1.first repeated and non repated
----------------------------------
        String str="dadapeer";


        System.out.println(" the first repeated element "+str.chars()
                .mapToObj(x-> (char)x)
                .filter(c-> str.chars().filter(x->x==c).count()>1)
                .findFirst()
                .get());

        System.out.println(" the first non repeated element "+str.chars()
                .mapToObj(x-> (char)x)
                .filter(c-> str.chars().filter(x->x==c).count()==1)
                .findFirst()
                .get());

        //second repeated
        System.out.println(" the second repeated element "+str.chars()
            .mapToObj(x-> (char)x)
            .filter(c-> str.chars().filter(x->x==c).count()>1)
            .skip(1)
            .findFirst()
            .get());

        System.out.println(" the second repeated element "+str.chars()
            .mapToObj(x-> (char)x)
            .filter(c-> str.chars().filter(x->x==c).count()==1)
            .skip(1)
            .findFirst()
            .get());

2. case change the string
------------------------------
String str="SHaIkdadapeeR";

System.out.println(str.chars().mapToObj(x-> (char)x)
                .map(x->  Character.isUpperCase(x)? Character.toLowerCase(x) :Character.toUpperCase(x))
                .map(String::valueOf)
                .collect(Collectors.joining()));


3. find the common character strings in a list
----------------------------------------------
    List<String> list=List.of("axh","xxx","anc","sss");
        System.out.println(list.stream().filter(x-> x.chars().allMatch(c-> c == x.charAt(0)))
                                                .collect(Collectors.toList()));


4. find the strings whose length is greather than 5
-------------------------------------------------------

    List<String> list=List.of("axh","xxx","anc","sss");

      System.out.println(list.stream().filter(x->x.length()>5).collect(Collectors.toList()));

5. find occuring character in string
---------------------------------------

    String s="kittyy";
        System.out.println(s.chars().mapToObj(c-> (char)c).filter(x-> s.chars()
                    .filter(y-> y==x).count()>1).distinct().collect(Collectors.toList()));

6. replace char 'a' with '$' in the given string
--------------------------------------------------
```

```
    String str="dadapeer"

        System.out.println(s.chars()
                .mapToObj(x-> (char)x)
                .map(c-> c == 'a' ? '$' : c)
                .map(Object::toString)
                .collect(Collectors.joining())
                );
```

7. replace char at particular index with '@'

```
    String sr = "dadapeer";

        System.out.println(
               IntStream.range(0, sr.length())
                .mapToObj(i -> i == 5 ? '@' : sr.charAt(i)) //index=5
                .map(Object::toString)
                .collect(Collectors.joining())
                );
```

8. remove spaces in a string using

```
    String sp="ma d am";
        System.out.println(sp.chars()
                .mapToObj(x -> String.valueOf((char)x))
                .filter(x -> !x.equals(" "))
                .collect(Collectors.joining()));
```

8. palindrome or not using java 8
----

9. palindrome strings in a list

```
    List<String> stringList=List.of("ada","dadapeer","madam","farak");

        System.out.println(stringList.stream()
                .map(x-> new StringBuilder(x))
                .filter(x-> stringList.contains(x.reverse().toString()))
                .map(Object::toString)
                .collect(Collectors.toList())
                );
```

10. find the number of Strings in a list whose length is greater 5

```
    List<String> ls=List.of("wdes","qawsed","wedrfnum","abc","poiujk");

        System.out.println("the number of strings with greather than length 5 is : "+
                ls.stream().filter(x-> x.length()>5)
                .count()
                );

        // finding Strings in a list whose length is greater 5

        System.out.println("the number of strings with greather than length 5 is : "+
                ls.stream().filter(x-> x.length()>5)
                .collect(Collectors.toList())
                );
```

11. converting Strings in a list in uppercase and lowercase

```
    List<String> ls=List.of("wdes","qawsed","wedrfnum","abc","poiujk");
```

```
        System.out.println(ls.stream().map(String::toUpperCase).collect(Collectors.toList
        ()));

        System.out.println(ls.stream().map(String::toLowerCase).collect(Collectors.toList
        ()));

12. filter the element whose 2nd number is 2  and find the average also
-------------------------------------------------------------------------

        List<Integer> lit = List.of(123, 142, 723, 124);

         List<Integer>ll=lit.stream().map(x-> x.toString())
                          .filter(x-> x.charAt(1)=='2')
                          .map(x->Integer.parseInt(x)).collect(Collectors.toList());

         System.out.println(ll  );
         System.out.printf ("the average of elements "+"%.2f",
                        ll.stream().collect(Collectors.averagingInt(x->x)));


 13. replace alternate chars with '$'
 ----------------------------------------------

         String sd="Capgemini";

         System.out.println(IntStream.range(0, sd.length())
                 .mapToObj(x->  (x%2!=0) ? "$":sd.charAt(x))
                 .map(Object::toString)
                 .collect(Collectors.joining()));

14. count frequeny of each elements
-----------------------------------------

      List<Integer> les = List.of(2,3,34,5,5,4,3,3,3);

        System.out.println(les.stream()
          .collect(Collectors.groupingBy(x-> x,()->new
          LinkedHashMap<>(),Collectors.counting())));


15. reverse a string using java8
-------------------------------------

    String s1tr = "Java is Good";

    String ans = Stream.of(s1tr)
                    .map(x -> new StringBuilder(x).reverse())
                    .collect(Collectors.joining(" "));
            System.out.println(ans);

16. reverse each character in a String
------------------------------------------

    String s1tr = "Java is Good";
        System.out.println(
                    Arrays.stream(s1tr.split(" "))
                    .map(x -> new StringBuilder().append(x).reverse().toString())
                    .collect(Collectors.joining(" ")));


17.  find first char in list
-----------------------------------
List<Character> charList = List.of('c', 'f', 'g', 'f', 'c', 'b');
        Set<Character>cs=new HashSet<Character>();
        System.out.println(charList.stream().filter(x->!cs.add(x)).findFirst().get());
    }

18. min max avg on list
------------------------------
```

```
    List<Integer> list=Arrays.asList(1,12,12,3,4,5);

        System.out.println(list.stream().count());//count

        System.out.println(list.stream().max(Integer::compare).get());//max

        System.out.println(list.stream().min(Integer::compare).get());//min

        System.out.printf("%.2f%n",list.stream().collect(Collectors.averagingInt(x->x)));
```

19 . even and odd

```
        System.out.println(list.stream().filter(x->x%2==0).collect(Collectors.toList()));

        System.out.println(list.stream().filter(x->x%2!=0).collect(Collectors.toList()));
        /*
         *
        list.stream().filter(x->x%2==0).collect(Collectors.toList())                           /// even odd
        .forEach(x-> System.out.print(" "+x));

         */
```

20. sorting and reversing

```
        System.out.println(list.stream().sorted().collect(Collectors.toList())); ///sorting  or ascending order


        System.out.println(list.stream().sorted(Comparator.reverseOrder()).collect(Collectors.toList()));  //descending order
```

21. starts with '1'

```
        System.out.println(list.stream().filter(x->x.toString().startsWith("1")).collect(Collectors.toList())); //start with 1
```

22. run a thread using lamda function

```
        Runnable task = () -> {
            System.out.println("Task is running");
        };
        // Start a new thread and run the task
        new Thread(task).start();
```

23. duplicates in the list

```
        System.out.println(list.stream().filter(x->Collections.frequency(list, x)<=1)
                    .collect(Collectors.toList()));  // removing duplicate in list

        System.out.println("duplicates "+list.stream().filter(x->Collections.frequency(list, x)>1)
                .collect(Collectors.toList())); // finding duplicate

//      list.stream().filter(c->Collections.frequency(list, c)>1)
//      .collect(Collectors.toSet()).forEach(System.out::println);
```

24. swapping of two numbers

```
        int a=5,b=6;
        System.out.println(" before swaping a= "+a+", "+"b= "+b);  /// swaping no's without 3rd
        a=a+b;
        b=a-b;
        a=a-b;
        System.out.println(" after swaping a= "+a+", "+"b= "+b);
        *
         */




25  . removing spaces
-----------------------

        String str="  d a d a p e e r";
        String result = str.replace(" ","");
        System.out.println(result);


        /* System.out.println(str.chars()
                    .filter(c -> c != ' ')       //remove space using stream api
                    .mapToObj(c -> (char) c)
                    .map(Object::toString)
                    .collect(Collectors.joining())); */

26. removing white spaces in strings
----------------------------------------
        List<String> strings=Arrays.asList("heel cclo"," sdf fd","dada p e e r");   // removing white spaces in strings
        System.out.println(strings.stream().map(x->x.replace(" ","")).collect(Collectors.toList()));

27. find first
---------------
        System.out.println(" find first "+list.stream().findFirst().get());//find the first element in list
        System.out.println();

        String input= "Cappgemiini";    // character how manytimes it repeated
        Map<Character, Long> charCountMap=input.chars().mapToObj(c->(char)c)
                                                .collect(Collectors.groupingBy(Function.identity(),Collectors.counting()));
        System.out.println(charCountMap);
        System.out.println();

        List<String> inputList = Arrays.asList("2", "3", "34", "", "5", "5", "4", "3", "3", "3");
        Map<String, Long> counts = inputList.stream()
                .filter(s -> !s.isEmpty())  // Filter out empty strings
                .collect(Collectors.groupingBy(s -> s, Collectors.counting()));
        counts.forEach((key, value) -> System.out.println(key + "-" + value));
        System.out.println();


        Runnable task = () -> {
            System.out.println("Task is running");
        };
        // Start a new thread and run the task
        new Thread(task).start();

28 .  reverse a string using stream
------------------------------------------------
        System.out.println(reverse("Anirudh"));//

    public static String reverse(String string) {
        return Stream.of(string)
            .map(word->new StringBuilder(word).reverse())
            .collect(Collectors.joining(" "));

    }

```