

```

1 1.first repeated and non repated
2 -----
3     String str="dadapeer";
4
5
6     System.out.println(" the first repeated element "+str.chars()
7         .mapToObj(x-> (char)x)
8         .filter(c-> str.chars().filter(x->x==c).count(>1)
9         .findFirst()
10        .get());
11
12    System.out.println(" the first non repeated element "+str.chars()
13        .mapToObj(x-> (char)x)
14        .filter(c-> str.chars().filter(x->x==c).count()==1)
15        .findFirst()
16        .get());
17
18    //second repeated
19    System.out.println(" the second repeated element "+str.chars()
20        .mapToObj(x-> (char)x)
21        .filter(c-> str.chars().filter(x->x==c).count(>1)
22        .skip(1)
23        .findFirst()
24        .get());
25
26    System.out.println(" the second repeated element "+str.chars()
27        .mapToObj(x-> (char)x)
28        .filter(c-> str.chars().filter(x->x==c).count()==1)
29        .skip(1)
30        .findFirst()
31        .get());
32
33 2. case change the string
34 -----
35    String str="SHaIkdadapeerR";
36
37    System.out.println(str.chars().mapToObj(x-> (char)x)
38        .map(x-> Character.isUpperCase(x)? Character.toLowerCase(x)
39        :Character.toUpperCase(x))
40        .map(String::valueOf)
41        .collect(Collectors.joining()));
42
43 3. find the common character strings in a list
44 -----
45    List<String> list=List.of("axh","xxx","anc","sss");
46    System.out.println(list.stream().filter(x-> x.chars().allMatch(c-> c ==
47        x.charAt(0)))
48        .collect(Collectors.toList()));
49
50 4. find the strings whose length is greather than 5
51 -----
52
53    List<String> list=List.of("axh","xxx","anc","sss");
54
55    System.out.println(list.stream().filter(x->x.length(>5).collect(Collectors.toList(
56        )))
57
58 5. find occuring character in string
59 -----
60
61    String s="kitty";
62    System.out.println(s.chars().mapToObj(c-> (char)c).filter(x-> s.chars()
63        .filter(y->
64        y==x).count(>1).distinct().collect(Collectors.toList()));
65
66 6. replace char 'a' with '$' in the given string
67 -----

```

```

65
66     String str="dadapeer"
67
68         System.out.println(s.chars()
69             .mapToObj(x-> (char)x)
70             .map(c-> c == 'a' ? '$' : c)
71             .map(Object::toString)
72             .collect(Collectors.joining())
73             );
74
75 7. replace char at particular index with '@'
76 -----
77
78     String sr = "dadapeer";
79
80         System.out.println(
81             IntStream.range(0, sr.length())
82                 .mapToObj(i -> i == 5 ? '@' : sr.charAt(i)) //index=5
83                 .map(Object::toString)
84                 .collect(Collectors.joining())
85                 );
86
87
88 8. remove spaces in a string using
89 -----
90
91     String sp="ma d am";
92     System.out.println(sp.chars()
93         .mapToObj(x -> String.valueOf((char)x))
94         .filter(x -> !x.equals(" "))
95         .collect(Collectors.joining()));
96
97 8. palindrome or not using java 8
98 -----
99
100 9. palindrome strings in a list
101 -----
102
103     List<String> stringList=List.of("ada","dadapeer","madam","farak");
104
105     System.out.println(stringList.stream()
106         .map(x-> new StringBuilder(x))
107         .filter(x-> stringList.contains(x.reverse().toString()))
108         .map(Object::toString)
109         .collect(Collectors.toList())
110         );
111
112 10. find the number of Strings in a list whose length is greater 5
113 -----
114
115     List<String> ls=List.of("wdes","qawsed","wedrfnum","abc","poiujk");
116
117     System.out.println("the number of strings with greather than length 5 is : "+
118         ls.stream().filter(x-> x.length()>5)
119         .count()
120         );
121
122     // finding Strings in a list whose length is greater 5
123
124     System.out.println("the number of strings with greather than length 5 is : "+
125         ls.stream().filter(x-> x.length()>5)
126         .collect(Collectors.toList())
127         );
128
129 11. converting Strings in a list in uppercase and lowercase
130 -----
131     List<String> ls=List.of("wdes","qawsed","wedrfnum","abc","poiujk");
132
133

```

```

134         System.out.println(ls.stream().map(String::toUpperCase).collect(Collectors.toList()));
135
136         System.out.println(ls.stream().map(String::toLowerCase).collect(Collectors.toList()));
137
138 12. filter the element whose 2nd number is 2 and find the average also
139 -----
140
141         List<Integer> lit = List.of(123, 142, 723, 124);
142
143         List<Integer> ll=lit.stream().map(x-> x.toString())
144             .filter(x-> x.charAt(1)=='2')
145             .map(x->Integer.parseInt(x)).collect(Collectors.toList());
146
147         System.out.println(ll );
148         System.out.printf ("the average of elements "+ "%.2f",
149             ll.stream().collect(Collectors.averagingInt(x->x)));
150
151 13. replace alternate chars with '$'
152 -----
153
154         String sd="Capgemini";
155
156         System.out.println(IntStream.range(0, sd.length())
157             .mapToObj(x-> (x%2!=0) ? "$":sd.charAt(x))
158             .map(Object::toString)
159             .collect(Collectors.joining()));
160
161 14. count frequency of each elements
162 -----
163
164         List<Integer> les = List.of(2,3,34,5,5,4,3,3,3);
165
166         System.out.println(les.stream()
167             .collect(Collectors.groupingBy(x-> x, ()->new
168                 LinkedHashMap<>(),Collectors.counting())));
169
170 15. reverse a string using java8
171 -----
172
173         String sltr = "Java is Good";
174
175         String ans = Stream.of(sltr)
176             .map(x -> new StringBuilder(x).reverse())
177             .collect(Collectors.joining(" "));
178         System.out.println(ans);
179
180 16. reverse each character in a String
181 -----
182
183         String sltr = "Java is Good";
184         System.out.println(
185             Arrays.stream(sltr.split(" "))
186             .map(x -> new StringBuilder().append(x).reverse().toString())
187             .collect(Collectors.joining(" ")));
188
189 17. find first char in list
190 -----
191
192         List<Character> charList = List.of('c', 'f', 'g', 'f', 'c', 'b');
193         Set<Character> cs=new HashSet<Character>();
194         System.out.println(charList.stream().filter(x->!cs.add(x)).findFirst().get());
195     }
196
197 18. min max avg on list
198 -----

```

```

198
199     List<Integer> list=Arrays.asList(1,12,12,3,4,5);
200
201     System.out.println(list.stream().count()); //count
202
203     System.out.println(list.stream().max(Integer::compare).get()); //max
204
205     System.out.println(list.stream().min(Integer::compare).get()); //min
206
207     System.out.printf("%.2f%n",list.stream().collect(Collectors.averagingInt(x->x)));
208
209
210 19 . even and odd
211 -----
212
213     System.out.println(list.stream().filter(x->
214     x%2==0).collect(Collectors.toList()));
215
216     System.out.println(list.stream().filter(x->
217     x%2!=0).collect(Collectors.toList()));
218     /*
219     *
220     list.stream().filter(x->
221     x%2==0).collect(Collectors.toList())           /// even odd
222     .forEach(x-> System.out.print(" "+x));
223     */
224
225 20. sorting and reversing
226 -----
227
228     System.out.println(list.stream().sorted().collect(Collectors.toList()));
229     ///sorting or ascending order
230
231
232     System.out.println(list.stream().sorted(Comparator.reverseOrder()).collect(Collectors.toList())); //descending order
233
234 21. starts with '1'
235 -----
236
237     System.out.println(list.stream().filter(x->x.toString().startsWith("1")).collect(
238     Collectors.toList())); //start with 1
239
240 22. run a thread using lamda function
241 -----
242
243     Runnable task = () -> {
244         System.out.println("Task is running");
245     };
246     // Start a new thread and run the task
247     new Thread(task).start();
248
249 23. duplicates in the list
250 -----
251
252     System.out.println(list.stream().filter(x->Collections.frequency(list, x)<=1)
253     .collect(Collectors.toList())); // removing duplicate in list
254
255     System.out.println("duplicates
256     "+list.stream().filter(x->Collections.frequency(list, x)>1)
257     .collect(Collectors.toList())); // finding duplicate
258
259     // list.stream().filter(c->Collections.frequency(list, c)>1)
260     // .collect(Collectors.toSet()).forEach(System.out::println);
261
262 24. swapping of two numbers
263 -----
264     int a=5,b=6;

```

```

258         System.out.println(" before swaping a= "+a+", "+"b= "+b);   /// swaping no's
        without 3rd
259         a=a+b;
260         b=a-b;
261         a=a-b;
262         System.out.println(" after swaping a= "+a+", "+"b= "+b);
263         *
264         */
265
266
267
268
269 25 . removing spaces
270 -----
271
272         String str=" d a d a p e e r";
273         String result = str.replace(" ", "");
274         System.out.println(result);
275
276
277         /* System.out.println(str.chars()
278                 .filter(c -> c != ' ')           //remove space using stream api
279                 .mapToObj(c -> (char) c)
280                 .map(Object::toString)
281                 .collect(Collectors.joining())); */
282
283 26. removing white spaces in strings
284 -----
285         List<String> strings=Arrays.asList("heel cclo"," sdf fd","dada p e e r");    //
        removing white spaces in strings
286         System.out.println(strings.stream().map(x->x.replace("
        ", "")).collect(Collectors.toList()));
287
288 27. find first
289 -----
290         System.out.println(" find first "+list.stream().findFirst().get()); //find the
        first element in list
291         System.out.println();
292
293         String input= "Cappgemiini";    // character how manytimes it repeated
294         Map<Character, Long> charCountMap=input.chars().mapToObj(c->(char) c)
295
        .collect(Collectors.groupingBy(Function.identity(), Collectors.counting()));
296
297         System.out.println(charCountMap);
298         System.out.println();
299
300         List<String> inputList = Arrays.asList("2", "3", "34", "", "5", "5", "4", "3",
        "3", "3");
301         Map<String, Long> counts = inputList.stream()
302                 .filter(s -> !s.isEmpty()) // Filter out empty strings
303                 .collect(Collectors.groupingBy(s -> s, Collectors.counting()));
304         counts.forEach((key, value) -> System.out.println(key + "-" + value));
305         System.out.println();
306
307         Runnable task = () -> {
308             System.out.println("Task is running");
309         };
310         // Start a new thread and run the task
311         new Thread(task).start();
312
313 28 . reverse a string using stream
314 -----
315         System.out.println(reverse("Anirudh")); //
316
317         public static String reverse(String string) {
318             return Stream.of(string)
319                     .map(word->new StringBuilder(word).reverse())

```

```
320         .collect(Collectors.joining(" "));
321
322     }
323
```