

```
1 26-May-2022      NonPrimeNumbersRange:
2 *****
3 Test1:
4 -----
5 package com.basicjava;
6
7 import java.util.Scanner;
8
9 public class NonPrimeNumbersRange {
10
11     public static void main(String[] args) {
12         Scanner scan = new Scanner(System.in);
13         System.out.println("Enter the first Number : ");
14         int a = scan.nextInt();
15         System.out.println("Enter the Second Number : ");
16         int b = scan.nextInt();
17         System.out.println("The range between the numbers are : ");
18
19         for (int i = a; i <= b; i++) {
20
21             boolean isPrimeNo = true;
22
23             for (int j = 2; j <= Math.sqrt(i); j++) {
24
25                 if (i % j == 0)
26                     isPrimeNo = false;
27
28             }
29
30             if (isPrimeNo)
31                 System.out.println("This is a prime no: " + i);
32             else {
33
34             }
35
```

```
36
37     }
38
39 }
40
41 }
42 -----
43 Output:
44 -----
45 Enter the first Number :
46 10
47 Enter the Second Number :
48 20
49 The range between the numbers are :
50 This is a prime no: 11
51 This is a prime no: 13
52 This is a prime no: 17
53 This is a prime no: 19
54 -----
55 Test1:          AdditionOfTwoNumbers:
56 *****          *****
57 package com.basicjava;
58 import java.util.Scanner;
59 public class AddTwoNumbers {
60
61     public static void main(String[] args)
62     {
63         Scanner scan=new Scanner(System.in);
64         System.out.println("Enter the FirstNumber: ");
65         int val1=scan.nextInt();
66         System.out.println("Enter the secondNumber : ");
67         int val2=scan.nextInt();
68         String strVal1=String.valueOf(val1);
69         String strVal2=String.valueOf(val2);
70         int size1=strVal1.length();
```

```
71     int size2=strVal2.length();
72     if(size1>size2)
73     {
74         String res=strVal1.substring(0, size1-1);
75         System.out.println("After Removing a last Digit : " +res);
76         int c=Integer.parseInt(res);
77         int d=Integer.parseInt(strVal2);
78         int e=c+d;
79         System.out.println("Addition Result : " +e);
80     }
81     else if(size1<size2)
82     {
83         String res=strVal2.substring(0, size2-1);
84         System.out.println("After Removing a last Digit : " +res);
85         int c=Integer.parseInt(res);
86         int d=Integer.parseInt(strVal1);
87         int e=c+d;
88         System.out.println("Addition Result : " +e);
89     }
90     else
91     {
92         int c=Integer.parseInt(strVal1);
93         int d=Integer.parseInt(strVal2);
94         int e=c+d;
95         System.out.println("Addition Result : " +e);
96     }
97 }
98
99 }
```

100 **output:**

101 **-----**

102 **Enter the FirstNumber:**

103 **1234**

104 **Enter the secondNumber :**

105 **135**

106 After Removing a last Digit : 123

107 Addition Result : 258

108 -----

109 test1:            AddOrSubtract

110 \*\*\*\*\*            \*\*\*\*\*

111 Statement:

112 -----

113 #. Write a program to add or subtract. Reverse the first number and check the second  
114 number is prime, if prime add with the reversed number else subtract with the  
115 reversed number

116

117 package com.basicjava;

118 import java.util.Scanner;

119 public class AddOrSubtract {

120

121     public static void main(String[] args) {

122         Scanner scan=new Scanner (System.in);

123         System.out.println("Enter the first no : ");

124         int a=scan.nextInt();

125         String input=String.valueOf(a);

126         StringBuilder sb=new StringBuilder(input);

127         String rev=sb.reverse().toString();

128         int revNumeric=Integer.parseInt(rev);

129         System.out.println("The reversed number is : " +revNumeric);

130

131         System.out.println("Enter the second input : ");

132         int sec=scan.nextInt();

133         boolean isPrime=true;

134         for(int i=2;i<sec;i++)

135         {

136             if(sec%i==0)

137             {

138                 System.out.println(sec +" is not a prime number");

139                 System.out.println(revNumeric-sec);

140                 isPrime=false;

```

141         break;
142     }
143
144 }
145 if(isPrime)
146 {
147     System.out.println(sec +" is a prime number");
148     System.out.println(revNumeric+sec);
149 }
150 }
151 }

```

152

153

154 **Output:**

155 -----

156 **Enter the first no :**

157 **123**

158 **The reversed number is : 321**

159 **Enter the second input :**

160 **7**

161 **7 is a prime number**

162 **328**

163

164 **123Enter the first no :**

165

166 **The reversed number is : 321**

167 **Enter the second input :**

168 **10**

169 **10 is not a prime number**

170 **311**

171 -----

172 **Test2 :           PositiveOrNegative**

173 **\*\*\*\*\*           \*\*\*\*\***

174 **import java.io.\*;**

175 **import java.util.\*;**

```
176
177 public class Solution {
178
179     public static void main(String[] args) {
180         Scanner scan=new Scanner(System.in);
181         int num=scan.nextInt();
182         if(num>=0)
183         {
184             System.out.println("Positive");
185         }
186         else
187             System.out.println("Negative");
188     }
189 }
190 output:
191 -----
192 5
193 positive
194 -3
195 negative
196 -----
197 Test2:          SumOfNaturalNumbers
198 *****          *****
199 import java.util.Scanner;
200 public class SumOfNaturalNumber {
201     public static void main(String args[])
202     {
203         Scanner scan=new Scanner(System.in);
204         int a=scan.nextInt();
205         int sum=(a*(a+1))/2;
206         System.out.println(sum);
207
208     }
209
210 }
```

```
211 output:
212 -----
213 input: 5
214 output: 15
215 -----
```

216 **Test 3:**

217 **\*\*\*\*\***

218 **#.EvenNumberCount**

219 **#.ReverseAnarray**

220 **#.DeleteAnElement...usingContinue;**

221 **#.SearchAnElement**

222 **#.MinimumNumber**

223 -----

224 **Test4:           HalfReverseAndAdd:-**

225 **\*\*\*\*\*           \*\*\*\*\***

226 **Statement:-    Need to write a program to reverse the half of the elements from an array**  
227 **-----       and add the even index's elements and print it.**

228

229 **package HacerRankProblems;**

230 **import java.util.Scanner;**

231 **public class HalfReverseAndAdd {**

232     **public static void main(String[] args) {**

233         **Scanner scan = new Scanner(System.in);**

234         **System.out.println("Enter the Size : ");**

235         **int size = scan.nextInt();**

236         **int haSize = size % 2 == 0 ? (size / 2) : (size / 2) + 1;**

237         **System.out.println(haSize);**

238         **int a[] = new int[size];**

239         **int b[] = new int[size];**

240         **int sum = 0;**

241         **for (int i = 0; i < size; i++) {**

242             **a[i] = scan.nextInt();**

243         **}**

244         **int j = 0;**

245         **for (int i = 0; i <= haSize - 1; i++) {**

```

246     b[j] = a[i];
247     j++;
248 }
249 for (int i = size - 1; i >= haSize; i--) {
250     b[j] = a[i];
251     j++;
252 }
253
254 for(int i=0;i<size;i++)
255 {
256     if(i%2!=0)
257     {
258         sum=sum+b[i];
259     }
260
261 }
262 for(int i=0;i<size;i++)
263 {
264     System.out.print(b[i] +" ");
265
266 }
267 System.out.println();
268 System.out.println(sum);
269
270 }
271
272 }
273 output:
274 *****
275 Input Format
276 -----
277 7
278 1 2 3 4 5 6 7
279 Output Format
280 -----

```



```
281 1 2 3 4 7 6 5
282 12
283 Here the second half is reversed
```

```
284 -----
```

```
285 Sample Input 0
```

```
286 -----
```

```
287 6
```

```
288 7 6 5 4 3 2
```

```
289 Sample Output 0
```

```
290 -----
```

```
291 7 6 5 2 3 4
```

```
292 12
```

```
293 -----
```

```
294 Test4:          Reverse first and second half
```

```
295 *****          *****
```

```
296 package HacerRankProblems;
```

```
297
```

```
298 import java.util.Scanner;
```

```
299
```

```
300 public class HalfReverseAndAdd {
```

```
301
```

```
302     public static void main(String[] args) {
```

```
303         Scanner scan = new Scanner(System.in);
```

```
304         System.out.println("Enter the Size : ");
```

```
305         int size = scan.nextInt();
```

```
306         int haSize = size / 2;
```

```
307         System.out.println(haSize);
```

```
308         int a[] = new int[size];
```

```
309         int b[] = new int[size];
```

```
310         int sum = 0;
```

```
311         for (int i = 0; i < size; i++) {
```

```
312             a[i] = scan.nextInt();
```

```
313
```

```
314     }
```

```
315     int j = 0;
```

```
316
317     if (size % 2 == 0) {
318         for (int i = haSize - 1; i >= 0; i--) {
319             b[j] = a[i];
320             j++;
321         }
322         for (int i = size - 1; i >= haSize; i--) {
323             b[j] = a[i];
324             j++;
325         }
326     } else {
327
328         for (int i = haSize - 1; i >= 0; i--) {
329             b[j] = a[i];
330             j++;
331         }
332
333         b[j] = a[haSize];
334         j++;
335         for (int i = size - 1; i > haSize; i--) {
336             b[j] = a[i];
337             j++;
338         }
339
340     }
341     for (int i = 0; i < size; i++) {
342         System.out.print(b[i] + " ");
343
344     }
345
346 }
347
348 }
349 OutPut:-
350 -----
```

351 **Sample Input 0**

352 -----

353 **6**

354 **1 2 3 4 5 6**

355 **Sample Output 0**

356 -----

357 **3 2 1 6 5 4**

358 -----

359 **Test6:           Print without Duplication**

360 **\*\*\*\*\*           \*\*\*\*\***

361 **Solving this problem by using LinkedHashSet:**

362 **\*\*\*\*\***

363 **import java.io.\*;**

364 **import java.util.\*;**

365

366 **public class Solution {**

367

368     **public static void main(String[] args) {**

369     **LinkedHashSet <Integer> lhs=new LinkedHashSet<Integer>();**

370     **Scanner scan=new Scanner(System.in);**

371     **int size=scan.nextInt();**

372     **for(int i=0;i<size;i++)**

373     **{**

374         **lhs.add(scan.nextInt());**

375     **}**

376     **for(int x: lhs)**

377     **{**

378         **System.out.print(x +" ");**

379     **}**

380     **}**

381 **}**

382 **Sample Input 0**

383 -----

384 **7**

385 **2 2 6 5 2 6 7**

386 **Sample Output 0**

387 -----

388 **2 6 5 7**

389 -----

390 **Test7: CountThe DistinctElement:**

391 **\*\*\*\*\***

392

393 **package HacerRankProblems;**

394 **import java.util.Scanner;**

395 **import java.util.ArrayList;**

396 **import java.util.Arrays;**

397 **import java.util.HashSet;**

398 **import java.util.List;**

399

400 **public class RemovingAllDuploicatesNumber {**

401

402 **public static void main(String[] args) {**

403 **Scanner scan=new Scanner (System.in);**

404 **System.out.println("Enter the size : ");**

405 **int size=scan.nextInt();**

406 **List<Integer> inputList = new ArrayList<Integer>();**

407 **for(int i=0;i<size;i++)**

408 **{**

409 **inputList.add(scan.nextInt());**

410 **}**

411 **HashSet<Integer> hs = new HashSet<Integer>();**

412 **HashSet<Integer> duplicateValues = new HashSet<Integer>();**

413 **for (int dupNum : inputList) {**

414 **if (!(hs.add(dupNum))) {**

415

416 **duplicateValues.add(dupNum);**

417 **}**

418 **}**

419 **System.out.println("Collected Duplicated numbers are : " +duplicateValues);**

420 **inputList.removeAll(duplicateValues);**

```
421     int size1=inputList.size();
422     System.out.println(size1);
423
424 }
425
426 }
427 Sample Input 0
428 -----
429 6
430 1 2 3 4 2 3
431 Sample Output 0
432 -----
433 2
434 -----
435 Test7:          Replace 0's with 1 and 1's with 0's
436 *****          *****
437 package com.basicjava;
438
439 import java.util.ArrayList;
440 import java.util.Scanner;
441
442 public class Replacing0sAnd1s {
443
444     public static void main(String[] args) {
445         Scanner scan=new Scanner(System.in);
446         System.out.println("Enter the number : ");
447         String a=scan.next();
448         char b[]=a.toCharArray();
449         ArrayList<Character> al=new ArrayList<Character>();
450         for(char c : b)
451         {
452             if(c=='0')
453             {
454                 al.add('1');
455             }
```

```

456         else
457         {
458             al.add('0');
459         }
460     }
461     al.forEach(System.out::print);
462 }
463
464 }
465 output:
466 -----
467 Sample Input 0
468 -----
469 101010
470 Sample Output 0
471 -----
472 010101
473 -----
474 test7:-           Prime or not
475 *****          *****
476
477 package HacerRankProblems;
478 import java.util.Scanner;
479 class PrimeCheck
480 {
481     void displayPrime(int num)
482     {
483         int flag=0;
484         for(int i=2;i<Math.sqrt(num);i++)
485         {
486             if(num%i==0)
487             {
488                 System.out.println("Not a Prime");
489                 flag=1;
490                 break;

```

```

491     }
492 }
493 if(flag==0)
494 {
495     System.out.println("Prime Number");
496 }
497
498 }
499 }
500 public class Test1PrimeOrNot {
501
502     public static void main(String[] args) {
503         Scanner scan=new Scanner(System.in);
504         System.out.println("Enter the number : ");
505         int num=scan.nextInt();
506         PrimeCheck pc=new PrimeCheck();
507         pc.displayPrime(num);
508
509
510     }
511
512 }
513 output:--
514 -----
515 11
516 not prime
517 10
518 prime
519 -----
520 Test8:      RemoveAllThe DuplicatesFromString
521 *****      *****
522 package com.basicjava;
523
524 import java.util.HashSet;
525

```

```

526 public class RemoveAllDuplicate {
527
528     public static void main(String[] args) {
529         String input="monoonabc3";
530         char b[]=input.toCharArray();
531         HashSet <Character> hs= new HashSet <Character>();
532         StringBuilder sb=new StringBuilder();
533         for(char c : b)
534         {
535             if(!(hs.add(c)))
536             {
537
538                 sb.append(c + "|");
539                 continue;
540             }
541         }
542         System.out.println("duplicates : " +sb);
543         String res=input.replaceAll(sb.toString(), "");
544         System.out.println("After removal of Duplicate : " +res);
545         System.out.println("The First Non Repeating Chacter in the string is : "+res.charAt(0));
546
547     }
548
549 }

```

550 -----

551 **output:-**

552 **\*\*\*\*\***

553 **duplicates : o|o|n|**

554 **After removal of Duplicate : mabc3**

555 **The First Non Repeating Chacter in the string is : m**

556

557 -----

558 **Test8:-            FirstNonRepeatingCharacter:-**

559 **\*\*\*\*\*            \*\*\*\*\***

560 **Above program use for this problem:**



```

561 -----
562 System.out.println("The First Non Repeating Chacter in the string is : "+res.charAt(0));
563 The First Non Repeating Chacter in the string is : m
564 -----
565 Test8:-                                MaximumOccuringCharacter:-
566 *****                                *****
567 #.Write an efficient code to find maximum occurring character in the input string
568 package com.mukesh.wipro.assignments;
569 import java.util.Scanner;
570 public class Arr7 {
571     public static void main(String[] args) {
572         Scanner scan=new Scanner(System.in);
573         String a=scan.next();
574         char b[]=a.toCharArray();
575         int size=a.length();
576         for(int i=0;i<size;i++)
577         {
578             for(int j=i+1;j<size;j++)
579             {
580                 if(b[i]==b[j])
581                 {
582                     System.out.println(b[j]);
583                     break;
584                 }
585             }
586         }
587     }
588
589 }
590 output:-
591 -----
592 Input Format
593 -----
594 test
595 Output Format

```

```

596 -----
597 t
598 Sample Input 0
599 -----
600 java
601 Sample Output 0
602 -----
603 a
604 -----
605 Test9:-          FactorialOfNumber:-
606 *****          *****
607 package com.basicjava;
608 import java.util.Scanner;
609 public class FactorialOfNumber {
610     public static void main(String args[])
611     {
612         Scanner scan =new Scanner (System.in);
613         int a;
614         int fact=1;
615         System.out.println("enter the number:");
616         a=scan.nextInt();
617         System.out.println("the factorial of a given number is:");
618         for(int i=1;i<=a;i++)
619         {
620             fact=fact*i;
621         }
622         System.out.println(fact);
623         scan.close();
624     }
625 }
626 }
627 -----
628 Test9:-          MergeTwoArrays:-
629 *****          *****
630 #.I can solve this problem efficently by using

```

631 **TreeSet Collection Interface:**

632 -----

633 **Statement:**

634 -----

635 the task is to get the two arrays and merge the two arrays

636 futher print the merged array in ascending format.

637

638 package HacerRankProblems;

639 import java.util.TreeSet;

640 import java.util.Arrays;

641 import java.util.Scanner;

642 public class MergeTwoArray {

643 public static void main(String[] args) {

644 Scanner scan=new Scanner(System.in);

645 TreeSet <Integer> ts=new TreeSet<Integer>();

646 System.out.println("Enter the Array Size : ");

647 int size=scan.nextInt();

648 for(int i=0;i<size;i++)

649 {

650 ts.add(scan.nextInt());

651 }

652 ts.addAll(Arrays.asList(new Integer[] (scan.nextInt())));

653 TreeSet <Integer> ts1=new TreeSet<Integer>();

654 System.out.println("Enter the Array Size1 : ");

655 int size1=scan.nextInt();

656 for(int i=0;i<size1;i++)

657 {

658 ts1.add(scan.nextInt());

659 }

660 ts.addAll(ts1); //this is the only businees logic in this problem

661 System.out.println("Final result in Set formet : "+ts);

662 for(int x: ts)

663 {

664 System.out.print(x+" ");

665 }

```

666
667     }
668 }
669 output:-
670 -----
671 Sample Input 0
672 -----
673 6
674 3 5 6 9 13 16
675 5
676 4 5 7 10 11
677 Sample Output 0
678 -----
679 3 4 5 6 7 9 10 11 13 16
680 -----
681 Test9:-          largest sum contiguous sub array
682 *****          *****
683 Find the largest sum contiguous sub array which should not have negative numbers. We
684 have to print the sum and the corresponding array elements which brought up the sum.
685 -----
686 package com.basicjava;
687 public class Test9LargestSubArray {
688
689     public static void main(String[] args)
690     {
691         int a[]={1,2,-8,2,15,3,-9};
692         int size=a.length;
693         int firNeg = 0,secNeg=0,sum=0;
694         StringBuilder sb=new StringBuilder();
695         for(int i=0;i<size;i++)
696         {
697             if(a[i]<0)
698             {
699                 System.out.println("First negative element in the location : "+i);
700                 firNeg=i;

```

```

701         break;
702     }
703 }
704 for(int i=firNeg+1;i<size;i++)
705 {
706     if(a[i]<0)
707     {
708         System.out.println("Second negative element in the location : "+i);
709         secNeg=i;
710         break;
711     }
712 }
713 for(int i=firNeg+1;i<secNeg;i++)
714 {
715     sum=sum+a[i];
716     sb.append(a[i]+" ");
717 }
718 System.out.println("The sum of numbers between the negative number is : "+sum);
719 System.out.println("The elements in between the numbers are : ");
720 System.out.println(sb);
721 }
722
723 }
724 output:
725 -----
726 First negative element in the location : 2
727 Second negative element in the location : 6
728 The sum of numbers between the negative number is : 20
729 The elements in between the numbers are :
730 2 15 3
731 -----
732 test:10                ReverseTheWord:
733 *****                *****
734 public class Solution {
735     public static void main(String args[] ) throws Exception {

```

```

736     Scanner scan=new Scanner(System.in);
737     String a[]=scan.nextLine().split(" ");
738     int size=a.length;
739     String sum="";
740     for(int i=size-1;i>=0;i--)
741     {
742         sum=sum+(a[i] +" ");
743     }
744     System.out.println(sum);
745 }
746 }
747 }
748 output
749 -----
750 Sample Input 0
751 -----
752 when are you coming
753 Sample Output 0
754 -----
755 coming you are when
756 -----
757 Test10:-          GreaterThanPrevoiusElementsAre:
758 *****          *****
759 package com.basicjava;
760
761 public class GreaterThanPreviousElement {
762
763     public static void main(String[] args) {
764         int a[]={1,-3,-5,7,6,2,-9,8};
765         StringBuilder sb=new StringBuilder();
766         int max=a[0];
767         sb.append(max +" ");
768         for(int i=1;i<a.length;i++)
769         {
770             if(max<a[i])

```

```

771     {
772         sb.append(a[i]+" ");
773     }
774     max=a[i];
775 }
776 System.out.println("The elements that greater than previous are : " +sb);
777
778 }

```

```

779
780 }
781 output:

```

```

782 -----
783 The elements that greater than previous are : 1 7 8

```

```

784 -----

```

```

785 Test12:                ReverseTheStringItSelf:
786 *****                *****

```

```

787 package HacerRankProblems;
788 import java.util.Scanner;
789 public class ReverseTheWordItSelf {
790     public static void main(String args[])
791     {
792         Scanner scan=new Scanner(System.in);
793         StringBuilder sb=new StringBuilder();
794         //System.out.println("Enter the String : ");
795         String a=scan.nextLine();
796         sb.append(a);
797         String b=sb.reverse().toString();
798         String res[]=b.split(" ");
799         String sum="";
800         for(int i=res.length-1;i>=0;i--)
801         {
802             sum=sum+(res[i] +" ");
803         }
804         System.out.println(sum.trim());
805

```

```
806     }
807
808 }
809 Sample Input
810 -----
811 when will you come
812 Sample Output
813 -----
814 nehw lliw uoy emoc
815 -----
816 Test13:-          RotateArray:-
817 *****          *****
818 package com.basicjava;
819
820 public class LogTest2 {
821
822     public static void main(String[] args) {
823         int inputArray[] = { 1,2,3,4,5 };
824         int size = inputArray.length;
825         int num = 3;
826         int revArray[] = new int[size];
827         int r = 0;
828         int tempArray[] = new int[size];
829         int t = 0;
830         for (int i = size - 1; i >= 0; i--) {
831             revArray[r] = inputArray[i];
832             r++;
833         }
834         for (int i = num - 1; i >= 0; i--) {
835             tempArray[t] = revArray[i];
836             t++;
837         }
838         for (int i = size - 1; i >= num; i--) {
839             tempArray[t] = revArray[i];
840             t++;
```



```
841     }
842     for (int x : tempArray) {
843         System.out.println(x);
844     }
845
846 }
847
848 }
849 output:
850 -----
851 Sample Input 0
852 -----
853 5
854 5 4 3 2 1
855 3
856 Sample Output 0
857 -----
858 3 2 1 5 4
859 -----
860 test13:      CollectionsRotationMinMaxSort
861 *****      *****
862 package com.basicjava;
863 import java.util.Collections;
864 import java.util.LinkedList;
865 import java.util.List;
866 import java.util.Scanner;
867 class Rotate
868 {
869     Scanner scan=new Scanner(System.in);
870
871     void program()
872     {
873         List <Integer> a=new LinkedList <Integer> ();
874         int size;
875         System.out.println("enter the array size: ");
```

```
876     size=scan.nextInt();
877     System.out.print("enter the element side by side: ");
878     for(int i=0;i<size;i++)
879     {
880         a.add(scan.nextInt());
881     }
882
883     System.out.print("enter the times to rotate:");
884     int times=scan.nextInt();
885     Collections.rotate(a, times);
886     System.out.println("after rotation : " +a.toString());
887     System.out.println("Rotated list is: " + a);
888
889
890     Collections.reverse(a);
891     System.out.println("reversed list is: " +a);
892
893     Collections.sort(a);
894     System.out.println("the Sorted list is: " + a);
895
896     System.out.println("the maximum element in the listbis:" +Collections.max(a));
897
898     System.out.println("the minimum element in the listbis:" +Collections.min(a));
899
900     Collections.shuffle(a);
901     System.out.println("the shuffeled list will be: " +a);
902 }
903 }
904 public class RotateArrayUsingCollection {
905     public static void main(String args[])
906     {
907         Rotate ts=new Rotate();
908         ts.program();
909     }
910 }
```

```
911
912 }
913
914 -----
915 output
916 *****
917 enter the array size:
918 5
919 enter the element side by side: 5 10 15 20 25
920 enter the times to rotate:3
921 after rotation : [15, 20, 25, 5, 10]
922 Rotated list is: [15, 20, 25, 5, 10]
923 reversed list is: [10, 5, 25, 20, 15]
924 the Sorted list is: [5, 10, 15, 20, 25]
925 the maximum element in the listbis:25
926 the minimum element in the listbis:5
927 the shuffeled list will be: [25, 10, 20, 15, 5]
928 -----
929
```