

1 31-May-2022 Extra element
2 *****

3 Find the extra element and its index

4
5 -----

6 Input Format

7 *****

8 5

9 4

10 10 20 30 12 5

11 10 5 30 20

12

13 Output Format

14 *****

15 12

16 3

17 -----

18 Sample Input 0

19 *****

20 4

21 5

22 -1 0 3 2

23 3 4 0 -1 2

24

25 Sample Output 0

26 *****

27 4

28 1

29 -----

30 package HacerRankProblems;

31

32 import java.util.Arrays;

33 import java.util.List;

34

35 public class ExtraElement {

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

37 List<Integer> al1=Arrays.asList(10 ,20, 30, 12, 5);

38 int size1=al1.size();

39 List<Integer> al2=Arrays.asList(10 ,5 ,30, 20);

40 int size2=al2.size();

41 StringBuilder sb=new StringBuilder();

42 int index = 0;

43 if(size1>size2)

44 {

45 for(int x: al1)

46 {

```

47         if(!(al2.contains(x)))
48         {
49
50             sb.append(x);
51             index=al1.indexOf(x);
52         }
53     }
54     System.out.println("The Extra element are : "+sb);
55     System.out.println("The index of element is : "+index);
56 }
57 else
58 {
59     for(int x: al2)
60     {
61         if(!(al1.contains(x)))
62         {
63             sb.append(x);
64             index=al2.indexOf(x);
65         }
66     }
67 }
68 System.out.println("The Extra element are : "+sb);
69 System.out.println("The index of element is : "+index);
70 }
71 }
72 }

```

OUTPUT:

The Extra element are : 12

The index of element is : 3

31-May-2022 After Element of Minimum Index

Statement:

Find the minimum element's index first from the given array.
Then print the next index's element from the array.

Input Format

6

5 6 3 7 2 8

Output Format

```

93 -----
94 8
95 -----
96 Sample Input 0
97 -----
98 7
99 1 2 3 4 5 6 7
100 Sample Output 0
101 -----
102 2
103 -----
104 package HacerRankProblems;
105 import java.util.ArrayList;
106 import java.util.Collections;
107 import java.util.Scanner;
108 public class AfterElementOfMinimumTest9 {
109
110     public static void main(String[] args) {
111         Scanner scan=new Scanner(System.in);
112         System.out.println("Enter the size of list : ");
113         int size=scan.nextInt();
114         ArrayList<Integer> al1=new ArrayList<Integer>();
115         System.out.println("Enter the element side by side : ");
116         for(int i=0;i<size;i++)
117         {
118             al1.add(scan.nextInt());
119         }
120         int min=Collections.min(al1);
121         System.out.println("The minimum element in the List is : " +min);
122         int minIndex=al1.indexOf(min);
123         int res=al1.get(minIndex+1);
124         System.out.println("After minumum index element is : " +res);
125         scan.close();
126
127
128     }
129
130 }
131
132 -----
133 output:
134 -----
135 Enter the size of list :
136 6
137 Enter the element side by side :
138 5 6 7 1 5 3

```

139 The minimum element in the List is : 1
140 After mininum index element is : 5
141 -----
142 **Count Michael's Stairslocked**
143 *****
144 **Statement:**
145 -----
146 **#. Michael has a beautiful house. There is only one staircase to reach his
bedroom.**
147 **He used to climb either 1 stair or 2 stairs at a time.**
148 **If we consider all possible combinations of climbing,**
149 **in how many ways can Michael reach to the top?**
150
151 **Input Format**
152 -----
153 **An Integer P denotes the number of stairs in the staircase.**
154 **If the number of stairs beyond 20, the output should print "Wrong
Infrastructure".**
155
156 **Sample Input 0**
157 -----
158 **5**
159 **Sample Output 0**
160 -----
161 **8**
162 **Explanation 0**
163 -----
164 **Sample 1:**
165 -----
166 **P=5 is the number of stairs in the staircase,**
167 **the possible way to reach on top would be 8. i.e. 1,1,1,1,1 and 2,2,1.**
168 **Hence, output will print "8"**
169 **Sample 2:**
170 -----
171 **P=22 is the number of stairs in the staircase But the range of stairs would
be 1 to 20.**
172 **22 is out of range entry. Hence, it should print Wrong Infrastructure"**
173
174 **Sample Input 1**
175 -----
176 **22**
177 **Sample Output 1**
178 -----
179 **Wrong Infrastructure**
180 -----
181

```

182 import java.util.Scanner;
183 public class CountMichealstairsTest9 {
184     public static void main(String[] args) {
185         Scanner scan=new Scanner (System.in);
186         System.out.println("Enter the nu mber of stairs : ");
187         int stair=scan.nextInt();
188         if((stair %2==0) && (stair<=20))
189         {
190             int stairQuo=stair/2;
191             System.out.println(stairQuo+stair);
192         }
193         else if((stair %2!=0) && (stair<=20))
194         {
195             int stairQuo=stair/2;
196             System.out.println(stairQuo+stair+1);
197         }
198         else if(stair>20)
199         {
200             System.out.println( "Wrong Infrastructure");
201         }
202     }
203 }
204 -----
205 01-June-2022          Print After Rotate
206 *****                *****
207 Write a program to Rotate kth times of the given array.
208 then print the given index range elements.
209
210 Input Format
211 -----
212 7
213 1 2 3 4 5 6 7
214 2
215 3 5
216
217 Output Format
218 -----
219 2 3 4
220 -----
221 Sample Input 0
222 -----
223 5
224 5 4 3 6 1
225 1
226 1 3
227 Sample Output 0

```

```

228 -----
229 5 4 3
230 -----
231 package HacerRankProblems;
232 import java.util.ArrayList;
233 import java.util.Collections;
234 import java.util.Scanner;
235 public class RotateArrayAndItsIndex {
236     public static void main(String[] args) {
237         Scanner scan = new Scanner(System.in);
238         System.out.println("Enter the List size : ");
239         int size = scan.nextInt();
240         ArrayList<Integer> al = new ArrayList<Integer>();
241         System.out.println("Enter the element side by side : ");
242         for (int i = 0; i < size; i++) {
243             al.add(scan.nextInt());
244         }
245         System.out.println("Enter the Times to Rotate : ");
246         int times = scan.nextInt();
247         Collections.rotate(al, times);
248         System.out.println("After the Rotation of elements are : ");
249         System.out.println(al);
250         System.out.println("Enter the Start index : ");
251         int startIndex = scan.nextInt();
252         System.out.println("Enter the end index : ");
253         int endIndex = scan.nextInt();
254         for (int i = startIndex; i <= endIndex; i++) {
255             System.out.print(al.get(i) + " ");
256         }
257         scan.close();
258
259     }
260
261 }

```

```

262 -----

```

```

263 output:
264 *****
265 Enter the List size :
266 5
267 Enter the element side by side :
268 5 4 3 6 1
269 Enter the Times to Rotate :
270 1
271 After the Rotation of elements are :
272 [1, 5, 4, 3, 6]
273 Enter the Start index :

```

```

274 1
275 Enter the end index :
276 3
277 5 4 3
278 -----
279             Sorting in Max and Min order :
280             *****
281 Input Format
282 -----
283 7
284 13 2 4 15 12 10 5
285
286 Output Format
287 -----
288 15 2 13 4 12 5 10
289 -----
290 Sample Input 0
291 -----
292 5
293 1 5 6 2 8
294
295 Sample Output 0
296 -----
297 8 1 6 2 5
298 -----
299 package HacerRankProblems;
300
301 import java.util.ArrayList;
302 import java.util.Arrays;
303 import java.util.Scanner;
304
305 public class MinMaxOrder {
306
307     public static void main(String[] args) {
308         Scanner scan=new Scanner(System.in);
309         System.out.println("Enter the ArraySize : ");
310         int size=scan.nextInt();
311         int a[]=new int[size];
312         System.out.println("Enter the element side by side : ");
313         for(int i=0;i<size;i++)
314         {
315             a[i]=scan.nextInt();
316         }
317         Arrays.sort(a);
318         System.out.println("Sorted array are : ");
319         for(int x: a)

```

```

320     {
321         System.out.print(x + " ");
322     }
323     ArrayList<Integer>al=new ArrayList<Integer>();
324     int i,j;
325     for(i=0,j=size-1;i<=j ;i++,j--)
326     {
327         if(i==j)
328         {
329             al.add(a[i]);
330         }
331         else
332         {
333             al.add(a[j]);
334             al.add(a[i]);
335         }
336     }
337     System.out.println("The sorted Max Min order is : ");
338     al.forEach(k->System.out.print(k + " "));
339 }
340 }
341
342 }

```

343 -----

344 **output:**

345 -----

346 **Enter the ArraySize :**

347 **5**

348 **Enter the element side by side :**

349 **2 5 2 6 8**

350 **Sorted array are :**

351 **2 2 5 6 8**

352 **The sorted Max Min order is :**

353 **8 2 6 2 5**

354 -----

355 **13-June-2022 SwapThe Array Element:**

356 *********

357 **package HacerRankProblems;**

358

359 **import java.util.ArrayList;**

360 **import java.util.Scanner;**

361 **public class SwapArrayTest {**

362

363 **public static void main(String[] args)**

364 **{**

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


```

366     System.out.println("Enter the size of Array : ");
367     int size=scan.nextInt();
368     int inputArray[]= new int[size];
369     System.out.println("Enter the Array element : ");
370     for(int i=0;i<size;i++)
371     {
372         inputArray[i]=scan.nextInt();
373     }
374     System.out.println("Enter the Times To Swap : ");
375     int swap=scan.nextInt();
376     ArrayList<Integer>tempList=new ArrayList<Integer>();
377     for(int i=size-swap;i<size;i++)
378     {
379         tempList.add(inputArray[i]);
380     }
381     for(int i=swap;i<size-swap;i++)
382     {
383         tempList.add(inputArray[i]);
384     }
385     for(int i=0;i<swap;i++)
386     {
387         tempList.add(inputArray[i]);
388     }
389     System.out.println("After Swaped : " +tempList);
390
391 }

```

```

392
393 }
394 output:
395 *****

```

```

396 Enter the size of Array :
397 5
398 Enter the Array element :
399 1 2 3 4 5
400 Enter the Times To Swap :
401 2
402 After Swaped : [4, 5, 3, 1, 2]

```

```

403 -----
404 12-JULY-2022           AnagramProgram
405 *****                 *****

```

```

406 problem:
407 -----

```

```

408 #.The given word is said to Anagram if Both String Occurence are same
409 and also tghe word may jumbled
410 example:
411 -----

```

```

412 input:
413 -----
414 mukesh
415 hsemuk
416 output
417 -----
418 Anagram
419 logic
420 *****
421     #. Covert both string into either Capital or small
422     #. Check the length,if both both length are nort same then
423         it is not a anagram
424     #.if both length are same,covert it into character
425     #.and SORT THE BOTH CHARACTER ARRAY
426     #.Compare each and every character with another character array
427     #.if all are same increment the counter,if counter and and length of string
         is same
428         the given string is anagram otherwise not.
429 -----
430 package com.basicjava;
431
432 import java.util.Arrays;
433 import java.util.Scanner;
434
435 class Anagram {
436     String anagramInputs(String input1, String input2) {
437         String in1=input1.toLowerCase();
438         String in2=input2.toLowerCase();
439
440         int size1 = in1.length();
441         int size2 = in2.length();
442         if (size1 != size2) {
443             return false;
444         } else {
445             char input1Char[] = input1.toCharArray();
446             char input2Char[] = input2.toCharArray();
447             Arrays.sort(input1Char);
448             Arrays.sort(input2Char);
449             int c = 0;
450             for (int i = 0; i < size1; i++) {
451                 if (input1Char[i] == input2Char[i]) {
452                     c++;
453                 }
454             }
455             if (c == size1) {
456                 return true;

```

```

457         } else {
458             return false;
459         }
460     }
461 }
462 }
463
464 public class AnagramTest {
465
466     public static void main(String[] args) {
467         Scanner scan = new Scanner(System.in);
468         System.out.println("Enter the input1 : ");
469         String input1 = scan.next().toLowerCase();
470         System.out.println("Enter the input2 : ");
471         String input2 = scan.next().toLowerCase();
472         Anagram a = new Anagram();
473         a.anagramInputs(input1, input2);
474
475     }
476
477 }
478 output:
479 *****
480 Enter the input1 :
481 earth
482 Enter the input2 :
483 thear
484 Anagram
485 -----
486 Enter the input1 :
487 mukesh
488 Enter the input2 :
489 logesh
490 Not a anagram
491
492 =====
493 19-July-2022      LoopingTest
494 *****          *****
495
496 package com.basicjava;
497
498 import java.util.ArrayList;
499
500 public class LoopingTest {
501
502     public static void main(String[] args) {

```

```

503
504     int a=5,b=3,t=5;
505     int temp=0,total;
506     ArrayList <Integer> al=new ArrayList<Integer>();
507     total=a+b;
508     al.add(total);
509     int j = 0,i;
510     for( i=2; j<5-1;j++)
511     {
512         total=total+i*b;
513         al.add(total);
514         i=i*2;
515     }
516     System.out.println(al);
517 }
518
519 }
520 output
521 *****
522 [8, 14, 26, 50, 98]
523
524 import java.util.ArrayList;
525 import java.util.Scanner;
526
527 public class LoopingIncrement {
528
529     public static void main(String[] args) {
530
531         int c = 0, i = 0;
532         Scanner in = new Scanner(System.in);
533         int t = in.nextInt();
534         int a[] = new int[t];
535         int b[] = new int[t];
536         int n[] = new int[t];
537
538         for ( i = 0; i < t; i++) {
539             a[i] = in.nextInt();
540             b[i] = in.nextInt();
541             n[i] = in.nextInt();
542
543         }
544
545         for ( i = 0; i < t; i++)
546         {
547             ArrayList<Integer> al = new ArrayList<Integer>();
548             c = a[i] + b[i];

```

```

549         al.add(c);
550         int k=0;
551         for(int j=2;k<n[i]-1;k++)
552         {
553             c = c+j*b[i];
554             al.add(c);
555             j=j*2;
556
557         }
558         al.forEach(y -> System.out.print(y + " "));
559         System.out.println();
560     }
561
562 }
563
564 }

```

565 **output: HackkerRank**
566 *********

567 **Sample Input**
568 **-----**

569 **2**
570 **0 2 10**
571 **5 3 5**

572 **Sample Output**
573 **-----**

574 **2 6 14 30 62 126 254 510 1022 2046**
575 **8 14 26 50 98**

576 **=====**

577 **SocksProblem:**
578 *********

579 **John works at a clothing store. He has a large pile of socks that he must pair by color for sale.**

580 **Given an array of integers representing the color of each sock,**
581 **determine how many pairs of socks with matching colors there are.**

582
583 **Input Format**
584 **-----**

585 **The first line contains an integer the number of socks represented in The**
586 **second line contains**
587 **space-separated integers describing the colors of the socks in the pile.**

588 **Output Format**
589 **-----**

590 **Return the total number of matching pairs of socks that John can sell.**

591
592 **Sample Input 0**

```

593 -----
594 9
595 10 20 20 10 10 30 50 10 20
596 Sample Output 0
597 -----
598 3
599 Program
600 *****
601 package HacerRankProblems;
602
603 import java.util.ArrayList;
604
605 import java.util.LinkedHashMap;
606 import java.util.Map.Entry;
607 import java.util.Scanner;
608
609 public class SocksProblem {
610
611     public static void main(String[] args) {
612
613         ArrayList<Integer>al=new ArrayList<Integer>();
614         Scanner scan=new Scanner(System.in);
615         System.out.print("Enter the Size Of List :");
616         int size=scan.nextInt();
617         System.out.print("Enter the Element side by side");
618         for(int i=0;i<size;i++ )
619         {
620             al.add(scan.nextInt());
621         }
622         ArrayList<Integer>resList=new ArrayList<Integer>();
623         LinkedHashMap <Integer,Integer>lhm=new
        LinkedHashMap<Integer,Integer>();
624         for(int x: al)
625         {
626             if(lhm.containsKey(x))
627             {
628                 lhm.put(x, lhm.get(x)+1);
629             }
630             else
631             {
632                 lhm.put(x, 1);
633             }
634         }
635         System.out.println("Result in Key Value Formate : ");
636         for(Entry entry:lhm.entrySet())
637         {

```

```

638         System.out.println(entry.getKey() + " - " + entry.getValue());
639         resList.add((Integer) entry.getValue());
640
641     }
642     System.out.println(" Values in the List are : "+resList);
643     ArrayList<Integer>sumList=new ArrayList<Integer>();
644     for(int i=0;i<resList.size();i++)
645     {
646         if(resList.get(i)!=1)
647         {
648             sumList.add(resList.get(i)/2);
649         }
650     }
651     int sum=0;
652     for(int x:sumList)
653     {
654         sum = sum+x;
655     }
656     System.out.println("Sum of Socks pairs is : " +sum);
657
658
659
660 }
661
662 }
663 output:
664 *****
665 Enter the Size Of List :5
666 Enter the Element side by side10 20 10 20 30
667 Result in Key Value Formate :
668 10 - 2
669 20 - 2
670 30 - 1
671 Values in the List are : [2, 2, 1]
672 Sum of Socks pairs is : 2
673 =====

```