**Q1. Swap and Maximize**

[**https://www.geeksforgeeks.org/problems/swap-and-maximize5859/1**](https://www.geeksforgeeks.org/problems/swap-and-maximize5859/1)

**Given an array arr[ ] of positive elements. Consider the array as a circular array, meaning the element after the last element is the first element of the array. The task is to find the maximum sum of the absolute differences between consecutive elements with shuffling of array elements allowed *i.e*. shuffle the array elements and make [a1..an] such order that  |a1 – a2| + |a2 – a3| + …… + |an-1– an| + |an – a1| is maximized.**

**Examples:**

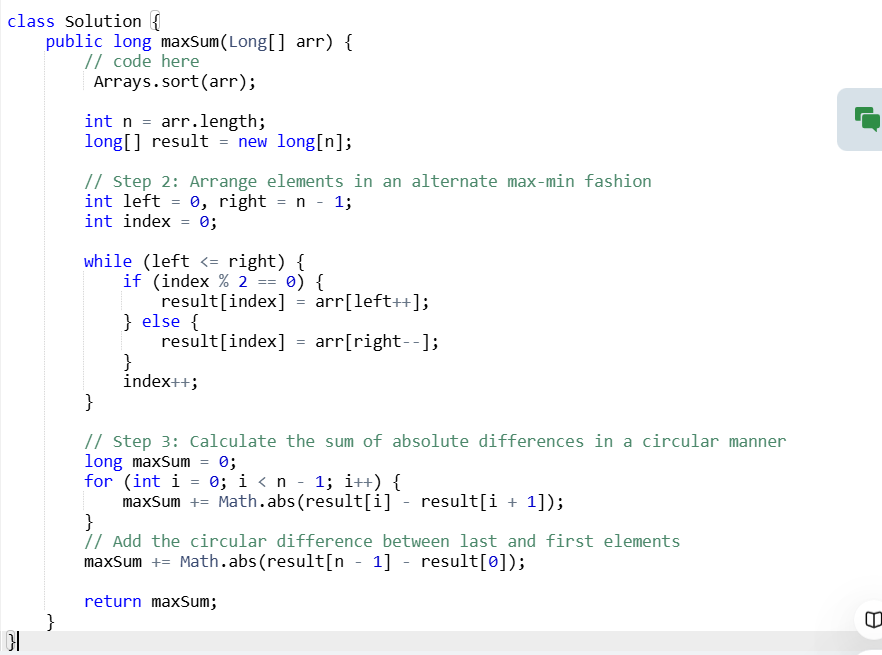
**Input: arr[] = [4, 2, 1, 8]**

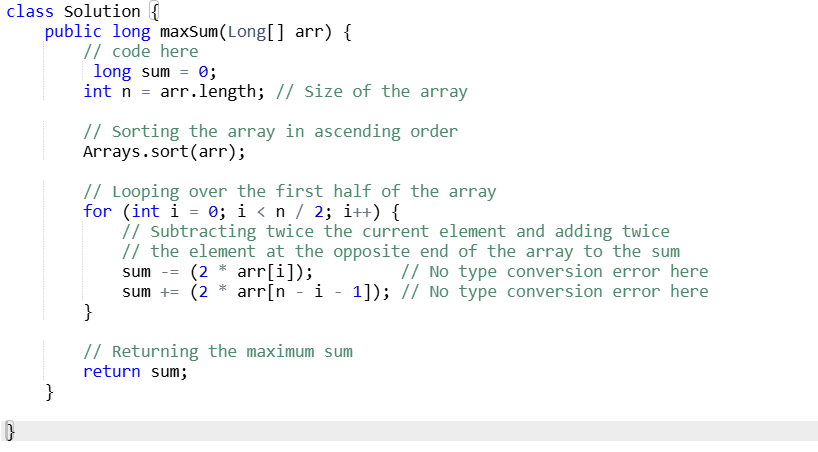
**Output: 18**

**Explanation: After Shuffling, we get [1, 8, 2, 4]. Sum of absolute difference between consecutive elements after rearrangement = |1 - 8| + |8 - 2| + |2 - 4| + |4 - 1| = 7 + 6 + 2 + 3 = 18.**

**Input: arr[] = [10, 12]**

**Output: 4  
Explanation: No need of rearrangement. Sum of absolute difference between consecutive elements = |10 - 12| + |12 - 10| = 2 + 2 = 4.**

****

****

**Time Complexity: O(nlogn).**  
**Auxiliary Space : O(1)**

**Q.2 Kth distance**

[**https://www.geeksforgeeks.org/problems/kth-distance3757/1**](https://www.geeksforgeeks.org/problems/kth-distance3757/1)

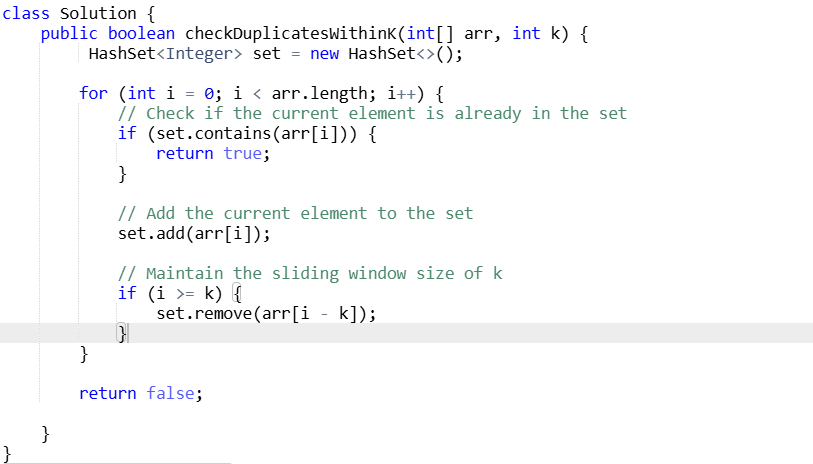
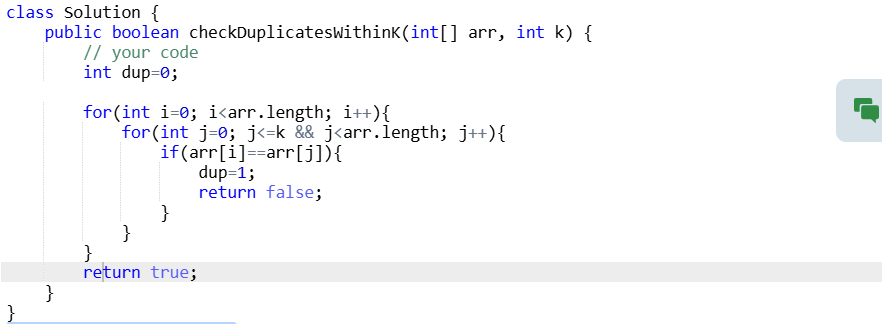
**Given an unsorted array arr and a number k which is smaller than the size of the array. Return true if the array contains any duplicate within k distance throughout the array else false.**

**Examples:**

**Input: arr[] = [1, 2, 3, 4, 1, 2, 3, 4] and k = 3  
Output: false  
Explanation: All duplicates are more than k distance away.**

**Input: arr[] = [1, 2, 3, 1, 4, 5] and k = 3  
Output: true  
Explanation: 1 is repeated at distance 3.**

**Input: arr[] = [6, 8, 4, 1, 8, 5, 7] and k = 3  
Output: true  
Explanation: 8 is repeated at distance 3.**

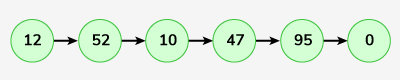


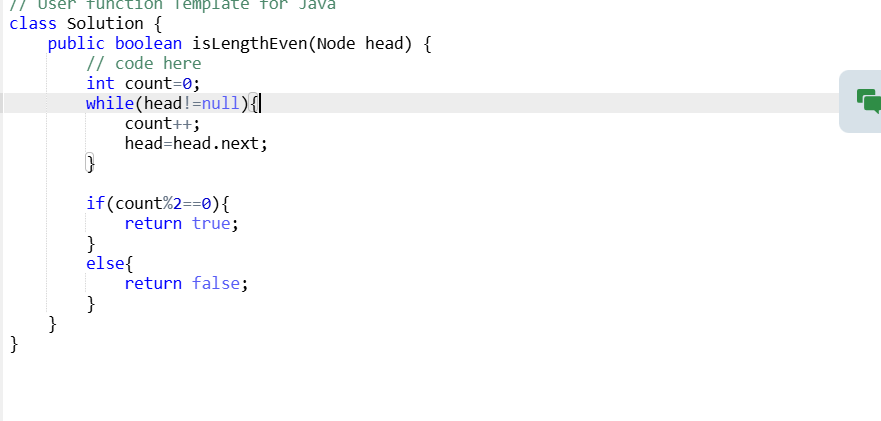
**Q.3 Is Linked List Length Even?**

[**https://www.geeksforgeeks.org/problems/linked-list-length-even-or-odd/1**](https://www.geeksforgeeks.org/problems/linked-list-length-even-or-odd/1)

Given a linked list, your task is to complete the function **isLengthEven()**which contains the **head** of the linked list, and check whether the length of the linked list is even or not. Return true if it is even, otherwise false.

**Examples:**

**Input:** Linked list: 12->52->10->47->95->0  
  
**Output:** true  
**Explanation:** The length of the linked list is 6 which is even, hence returned true.



**Q.4 Find All Triplets with Zero Sum**

[**https://www.geeksforgeeks.org/problems/find-all-triplets-with-zero-sum/1**](https://www.geeksforgeeks.org/problems/find-all-triplets-with-zero-sum/1)

**Given an array arr[], find all possible indices [i, j, k] of triplets [arr[i], arr[j], arr[k]] in the array whose sum is equal to zero. Return indices of triplets in any order and all the returned triplets indices should also be internally sorted, i.e., for any triplet indices [i, j, k], the condition i < j < k should hold.**

**Note: Try to solve this using the O(n2) approach.**

**Examples:**

**Input: arr[] = [0, -1, 2, -3, 1]**

**Output: [[0, 1, 4], [2, 3, 4]]**

**Explanation: Triplets with sum 0 are:  
arr[0] + arr[1] + arr[4] = 0 + (-1) + 1 = 0**

**arr[2] + arr[3] + arr[4] = 2 + (-3) + 1 = 0**

**Input: arr[] = [1, -2, 1, 0, 5]**

**Output: [[0, 1, 2]]**

**Explanation: Only triplet which satisfies the condition is arr[0] + arr[1] + arr[2] = 1 + (-2) + 1 = 0**

**Input: arr[] = [2, 3, 1, 0, 5]**

**Output: [[]]**

**Explanation: There is no triplet with sum 0.**

****

**Q.5 Split array in three equal sum subarrays**

<https://www.geeksforgeeks.org/problems/split-array-in-three-equal-sum-subarrays/1>

Given anarray, **arr[],** determine if **arr** can be split into three consecutive parts such that the sum of each part is equal. If possible, return any **index**pair(i, j) in an array such that sum(arr[0..i]) = sum(arr[i+1..j]) = sum(arr[j+1..n-1]), otherwise return an array **{-1,-1}**.

Note: Since multiple answers are possible, return any of them. The driver code will print true if it is correct otherwise, it will print false.

**Examples :**

**Input:**  arr[] = [1, 3, 4, 0, 4]

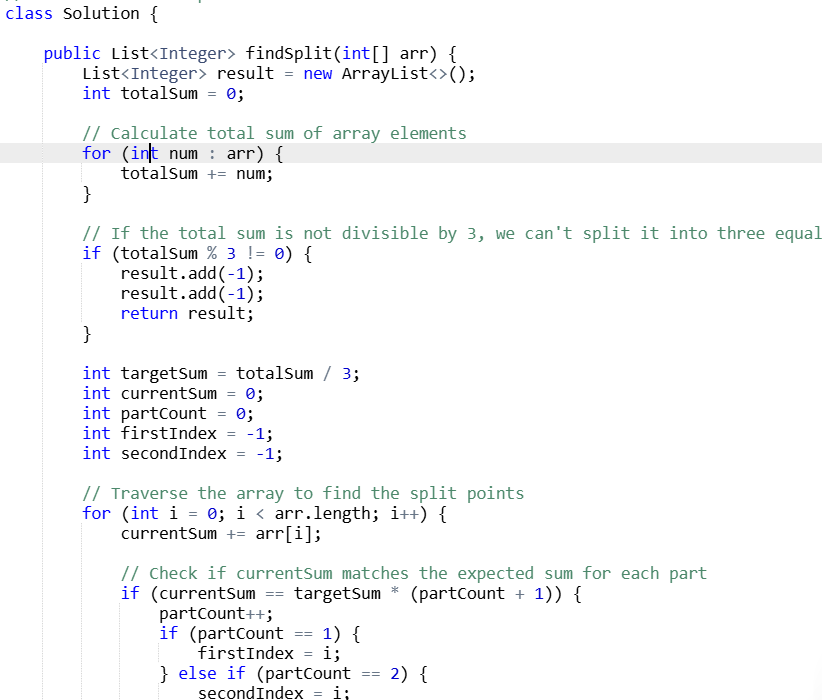
**Output:** true

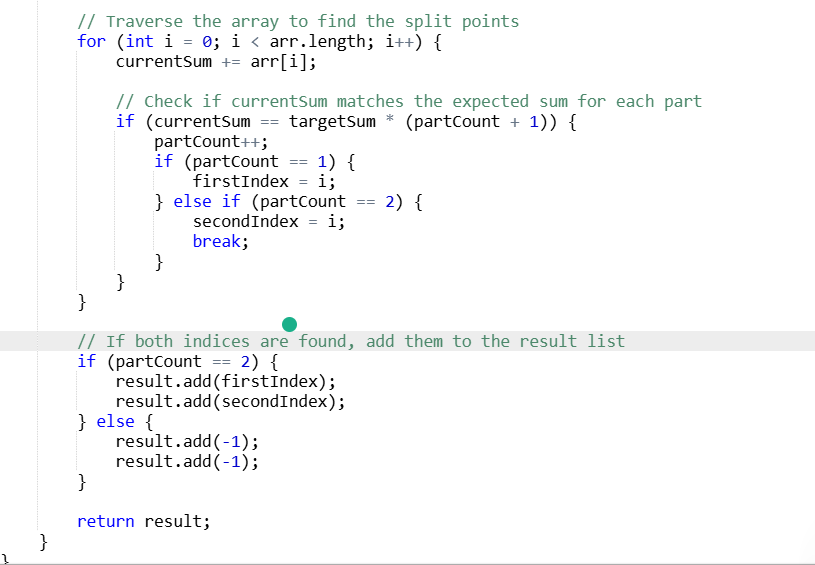
**Explanation:** [1, 2] is valid pair as sum of subarray arr[0..1] is equal to sum of subarray arr[2..3] and also to sum of subarray arr[4..4]. The sum is 4, so driver code prints true.

**Input:** arr[] = [2, 3, 4]

**Output:** false

**Explanation:** No three subarrays exist which have equal sum





**Q6.Minimum repeat to make substring**

[**https://www.geeksforgeeks.org/problems/minimum-times-a-has-to-be-repeated-such-that-b-is-a-substring-of-it--170645/1**](https://www.geeksforgeeks.org/problems/minimum-times-a-has-to-be-repeated-such-that-b-is-a-substring-of-it--170645/1)

**Given two strings s1 and s2. Return a minimum number of times s1 has to be repeated such that s2 is a substring of it. If s2 can never be a substring then return -1.**

**Note: Both the strings contain only lowercase letters.**

**Examples:**

**Input: s1 = "ww", s2 = "www"**

**Output: 2**

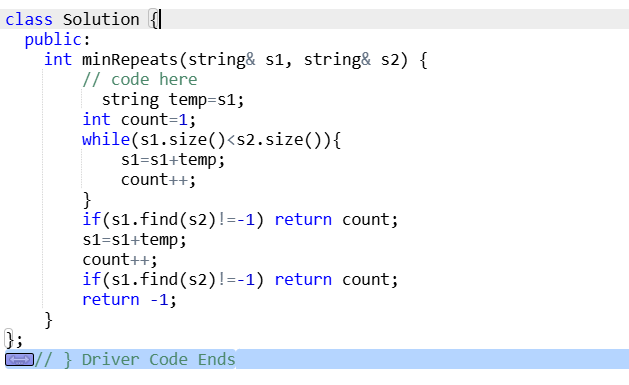
**Explanation: Repeating s1 two times (wwww), s2 is a substring of it.**

**Input: s1 = "abcd", s2 = "cdabcdab"   
Output: 3   
Explanation: Repeating s1 three times (abcdabcdabcd), s2 is a substring of it. s2 is not a substring of s2 when it is repeated less than 3 times.**

**Input: s1 = "ab", s2 = "cab"**

**Output: -1**

**Explanation: No matter how many times we repeat s1, we can't get a string such that s2 is a substring of it.**

****

**Q7. Min and Max in Array**

[**https://www.geeksforgeeks.org/problems/find-minimum-and-maximum-element-in-an-array4428/1?page=1&category=Arrays&sprint=50746f92a895c22a50504ac0c1fb9c84&sortBy=difficulty**](https://www.geeksforgeeks.org/problems/find-minimum-and-maximum-element-in-an-array4428/1?page=1&category=Arrays&sprint=50746f92a895c22a50504ac0c1fb9c84&sortBy=difficulty)

**Given an array arr. Your task is to find the minimum and maximum elements in the array.**

**Note: Return an array that contains two elements the first one will be a minimum element and the second will be a maximum of an array.**

**Examples:**

**Input: arr = [3, 2, 1, 56, 10000, 167]**

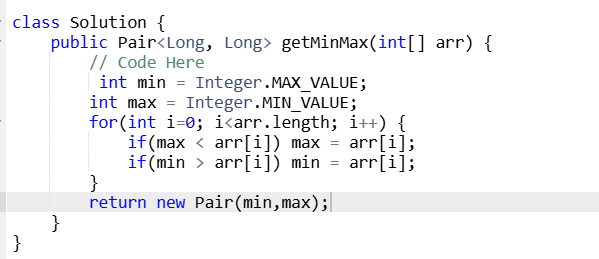
**Output: 1 10000  
Explanation: minimum and maximum elements of array are 1 and 10000.**

**Input: arr = [1, 345, 234, 21, 56789]**

**Output: 1 56789  
Explanation: minimum and maximum element of array are 1 and 56789.**

**Input: arr = [56789]**

**Output: 56789 56789**

****

**Q. Minimum sum**

[**https://www.geeksforgeeks.org/problems/minimum-sum4058/1**](https://www.geeksforgeeks.org/problems/minimum-sum4058/1)

**Given an array arr[] such that each element is in the range [0 - 9], find the minimum possible sum of two numbers formed using the elements of the array. All digits in the given array must be used to form the two numbers. Return a string without leading zeroes.**

**Examples :**

**Input: arr[] = [6, 8, 4, 5, 2, 3]**

**Output: 604**

**Explanation: The minimum sum is formed by numbers 358 and 246.**

**Input: arr[] = [5, 3, 0, 7, 4]**

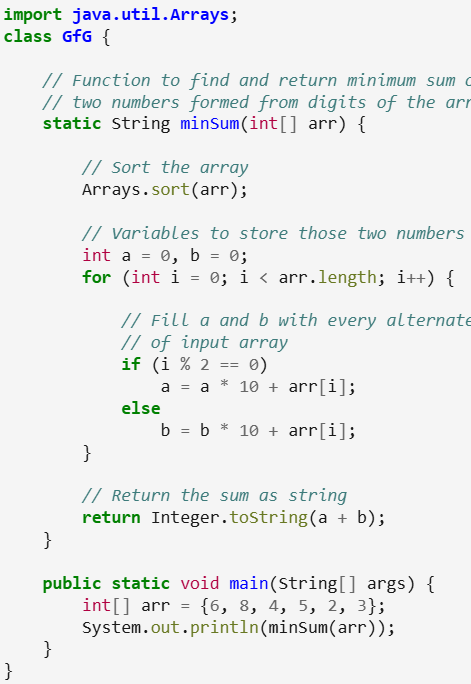
**Output: 82**

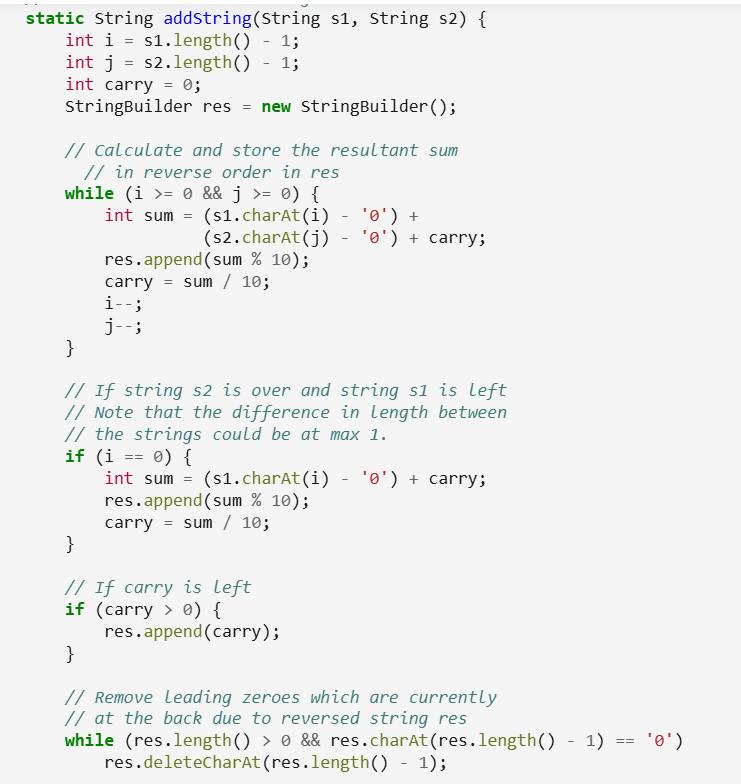
**Explanation: The minimum sum is formed by numbers 35 and 047.**

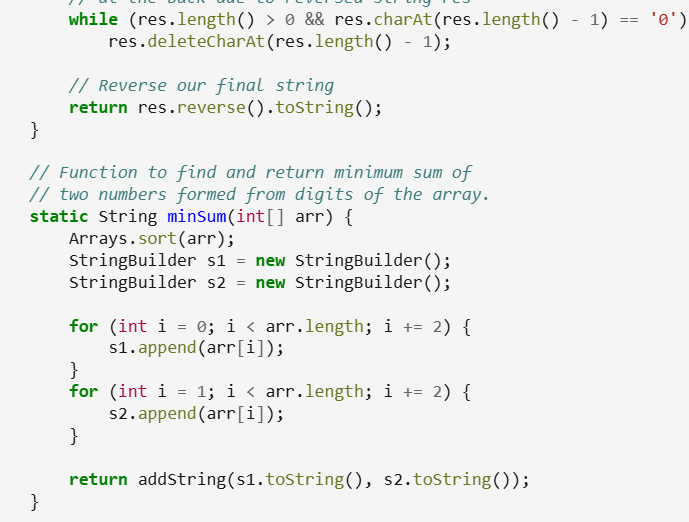
**Input: arr[] = [9, 4]**

**Output: 13**

**Explanation: The minimum sum is formed by numbers 9 and 4.**

****





**Q. Two sum -Pairs with 0 Sum**

[**https://www.geeksforgeeks.org/problems/count-pairs-with-given-sum5022/1?page=1&category=Arrays&sprint=50746f92a895c22a50504ac0c1fb9c84&sortBy=difficulty**](https://www.geeksforgeeks.org/problems/count-pairs-with-given-sum5022/1?page=1&category=Arrays&sprint=50746f92a895c22a50504ac0c1fb9c84&sortBy=difficulty)

Given an integer array **arr**, return all the **unique**pairs [arr[i], arr[j]] such that**i != j**and**arr[i] + arr[j] == 0.**

Note: The pairs must be returned in **sorted**order, the solution arrayshould also be **sorted**, and the answer must not contain any **duplicate**pairs.

**Examples:**

**Input:** arr = [-1, 0, 1, 2, -1, -4]

**Output:** [[-1, 1]]

**Explanation:** arr[0] + arr[2] = (-1)+ 1 = 0.

arr[2] + arr[4] = 1 + (-1) = 0.

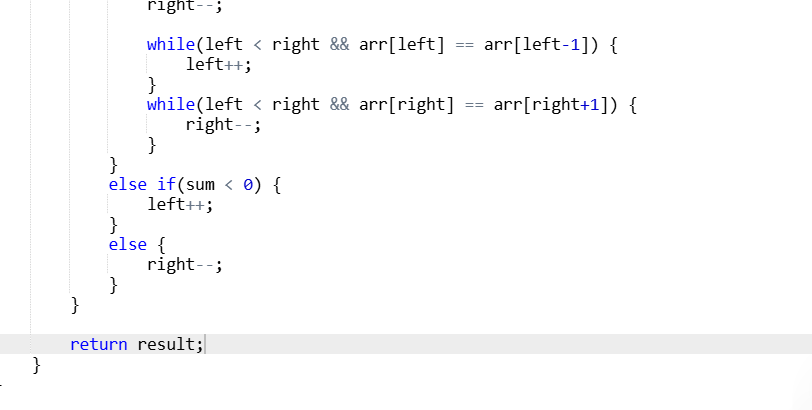
The distinct triplets are [-1,1].

**Input:** arr = [6, 1, 8, 0, 4, -9, -1, -10, -6, -5]

**Output:** [[-6, 6],[-1, 1]]

**Explanation:** The distinct triplets are [-1, 1] and [-6, 6].





**Q. Missing And Repeating**

Given an unsorted array arr of positive integers. One number a from the set [1, 2,....,n] is missing and one number b occurs twice in the array. Find numbers a and b.

Note: The test cases are generated such that there always exists one missing and one repeating number within the range [1,n].

Examples:

Input: arr[] = [2, 2]

Output: [2, 1]

Explanation: Repeating number is 2 and smallest positive missing number is 1.

Input: arr[] = [1, 3, 3]

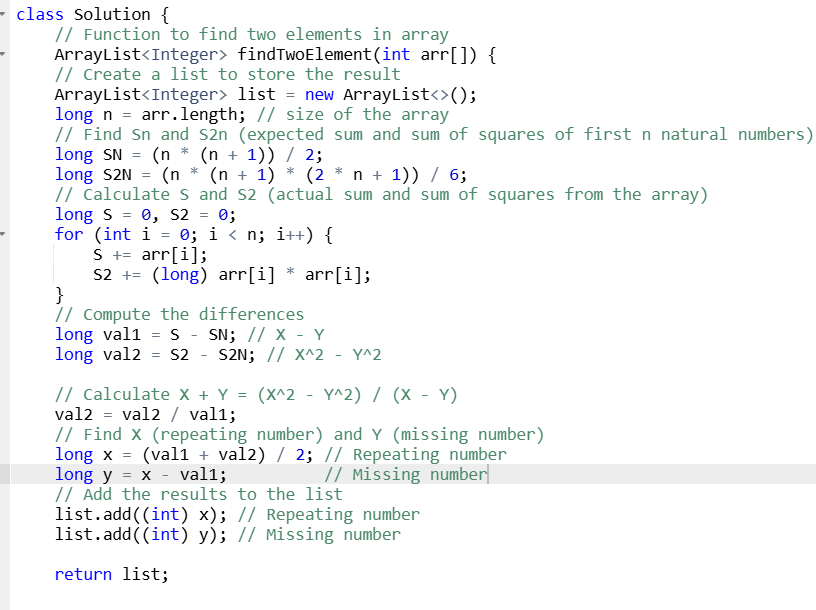
Output: [3, 2]

Explanation: Repeating number is 3 and smallest positive missing number is 2.

Input: arr[] = [4, 3, 6, 2, 1, 1]

Output: [1, 5]

Explanation: Repeating number is 1 and the missing number is 5.



**Coin Change (Count Ways)**

<https://www.geeksforgeeks.org/problems/coin-change2448/1?page=1&category=Arrays&sprint=50746f92a895c22a50504ac0c1fb9c84&sortBy=difficulty>

Difficulty: **Medium**Accuracy: **43.1%**Submissions: **272K+**Points: **4**

Given an integer array **coins[ ]**representing different denominations of currency and an integer **sum**, find the number of ways you can make **sum** by using different combinations from coins[ ].   
Note: Assume that you have an infinite supply of each type of coin. And you can use any coin as many times as you want.  
Answers are guaranteed to fit into a 32-bit integer.

**Examples:**

**Input:** coins[] = [1, 2, 3], sum = 4

**Output:** 4

**Explanation**: Four Possible ways are: [1, 1, 1, 1], [1, 1, 2], [2, 2], [1, 3].

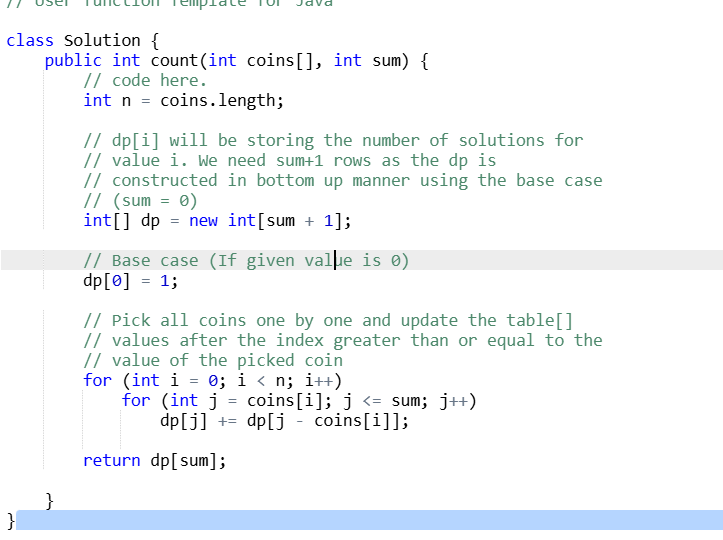
**Input**: coins[] = [2, 5, 3, 6], sum = 10

**Output:** 5

**Explanation**: Five Possible ways are: [2, 2, 2, 2, 2], [2, 2, 3, 3], [2, 2, 6], [2, 3, 5] and [5, 5].

**Input**: coins[] = [5, 10], sum = 3

**Output:** 0  
**Explanation:** Since all coin denominations are greater than sum, no combination can make the target sum.



**Second Largest**

Difficulty: **Easy**Accuracy: **26.72%**Submissions: **809K+**Points: **2**

Given an array of **positive** integers **arr[]**, return the **second largest** element from the array. If the second largest element doesn't exist then return **-1.**

Note: The second largest element should not be equal to the largest element.

**Examples:**

**Input:** arr[] = [12, 35, 1, 10, 34, 1]

**Output:** 34

**Explanation:** The largest element of the array is 35 and the second largest element is 34.

**Input:** arr[] = [10, 5, 10]

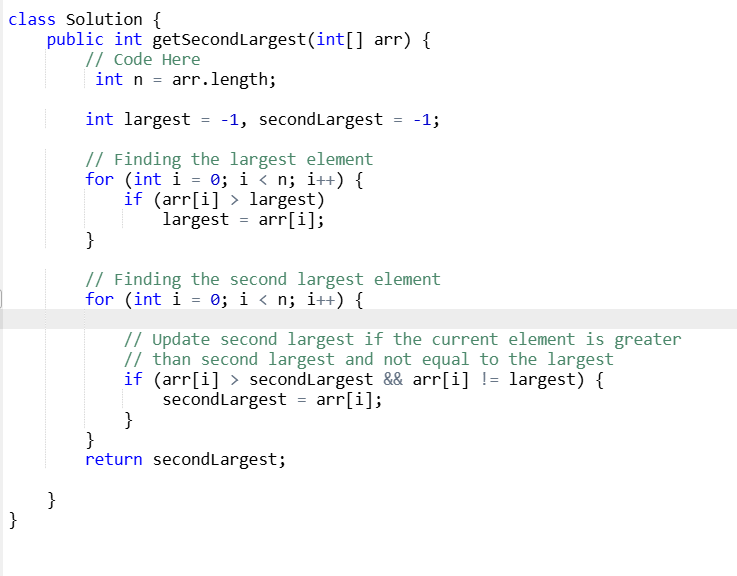
**Output:** 5

**Explanation:** The largest element of the array is 10 and the second largest element is 5.

**Input:** arr[] = [10, 10, 10]

**Output:** -1

**Explanation:** The largest element of the array is 10 and the second largest element does not exist.



**Move All Zeroes to End**

Difficulty: **Easy**Accuracy: **45.51%**Submissions: **221K+**Points: **2**

https://www.geeksforgeeks.org/batch/gfg-160-problems/track/arrays-gfg-160/problem/move-all-zeroes-to-end-of-array0751

Given an array **arr[]**. Push all the zeros of the given array to the right end of the array while maintaining the order of non-zero elements. Do the mentioned change in the **array in place**.

**Examples:**

**Input:** arr[] = [1, 2, 0, 4, 3, 0, 5, 0]

**Output:** [1, 2, 4, 3, 5, 0, 0, 0]

**Explanation:** There are three 0s that are moved to the end.

**Input:** arr[] = [10, 20, 30]

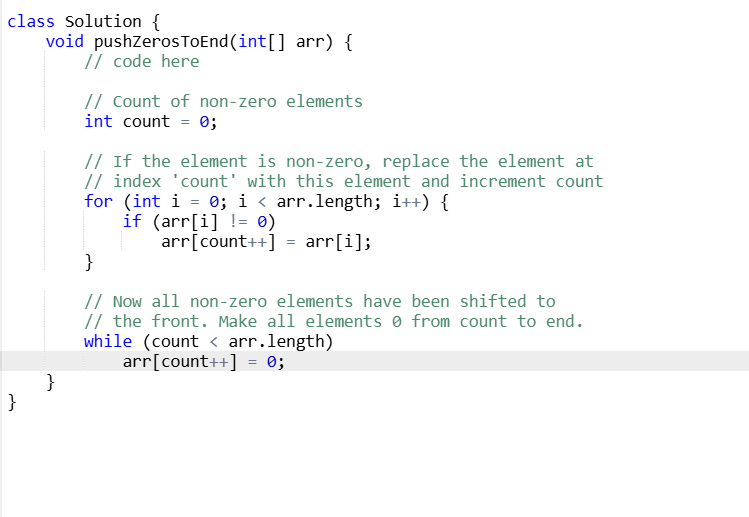
**Output:** [10, 20, 30]

**Explanation:** No change in array as there are no 0s.

**Input:** arr[] = [0, 0]

**Output:** [0, 0]

**Explanation:** No change in array as there are all 0s



**Reverse an Array**

Difficulty: **Easy**Accuracy: **55.32%**Submissions: **64K+**Points: **2**

You are given an array of integers **arr[]**. Your task is to **reverse** the given array.

**Examples:**

**Input:** arr = [1, 4, 3, 2, 6, 5]

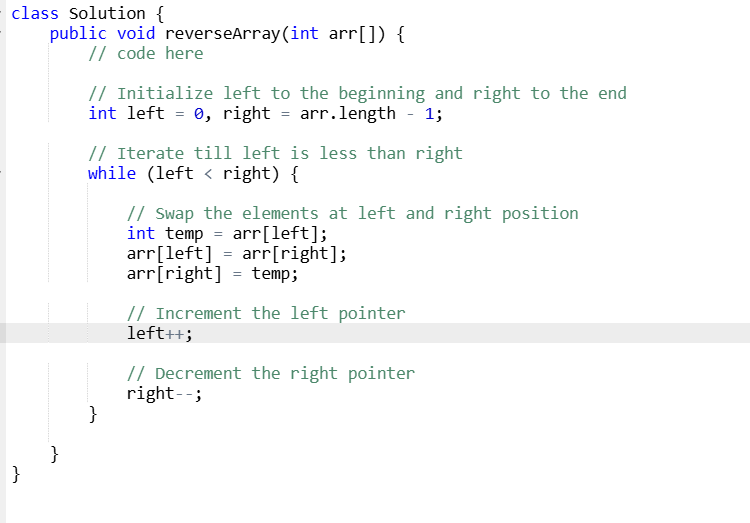
**Output:** [5, 6, 2, 3, 4, 1]  
**Explanation:** The elements of the array are 1 4 3 2 6 5. After reversing the array, the first element goes to the last position, the second element goes to the second last position and so on. Hence, the answer is 5 6 2 3 4 1.

**Input**: arr = [4, 5, 2]

**Output:** [2, 5, 4]  
**Explanation:** The elements of the array are 4 5 2. The reversed array will be 2 5 4.

**Input**: arr = [1]

**Output:** [1]  
**Explanation:** The array has only single element, hence the reversed array is same as the original.



**Rotate Array**

**https://www.geeksforgeeks.org/batch/gfg-160-problems/track/arrays-gfg-160/problem/rotate-array-by-n-elements-1587115621**

Difficulty: **Medium**Accuracy: **37.06%**Submissions: **380K+**Points: **4**

Given an unsorted array *arr[]****.*** Rotate the array to the left (counter-clockwise direction) by*d* steps, where *d* is a positive integer. Do the mentioned change in the **array in place**.

Note:Consider the array as circular.

**Examples :**

**Input:** arr[] = [1, 2, 3, 4, 5], d = 2

**Output:** [3, 4, 5, 1, 2]

**Explanation:** when rotated by 2 elements, it becomes 3 4 5 1 2.

**Input:** arr[] = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20], d = 3

**Output:** [8, 10, 12, 14, 16, 18, 20, 2, 4, 6]

**Explanation:** when rotated by 3 elements, it becomes 8 10 12 14 16 18 20 2 4 6.

**Input:** arr[] = [7, 3, 9, 1], d = 9

**Output:** [3, 9, 1, 7]

**Explanation:** when we rotate 9 times, we'll get 3 9 1 7 as resultant array.

