**Arrays- SET3**

1. [Write a C program for the following: Given an array A[] and a number x, check and display for all pairs in A[] with sum as x](https://www.geeksforgeeks.org/write-a-c-program-that-given-a-set-a-of-n-numbers-and-another-number-x-determines-whether-or-not-there-exist-two-elements-in-s-whose-sum-is-exactly-x/).
2. Write a C program to find minimum difference between any two elements.

Given an unsorted array **arr[]** of size **n**, the task is to find the minimum difference between any pair in the given array.

**Input:** arr[] = {1, 2, 3, 4}  
**Output:** 1  
The possible absolute differences are:  
{1, 2, 3, 1, 2, 1}

**Input:** arr[] = {10, 2, 5, 4}  
**Output:** 1

1. Given an array of N integers, count the total pairs of integers that have a difference of K.

Sample:

arraysize: 5 difference: 2

1 5 3 4 2

Output: count=3 3.

Explanation:

There are 3 pairs with difference 2 :( 1,3) , (5,3), (4,2)

1. Write a C program to read n numbers into an array and to rotate the elements to the right and left as shown below:

Rotate to Right:

Sample: 12 34 67 89

Output: 34 67 89 12.

Rotate to left

Sample: 12 34 67 89

Output: 89 12 34 67

1. Given an array **arr[]** of non-negative integers and an integer **sum**, find a subarray that adds to a given **sum**.

**Note:** There may be more than one subarray with sum as the given sum, print first such subarray.

**Examples:**

**Input**: arr[] = {1, 4, 20, 3, 10, 5}, sum = 33  
**Output**: Sum found between indexes 2 and 4  
**Explanation:**Sum of elements between indices 2 and 4 is 20 + 3 + 10 = 33

1. **Rearrange an array such that arr[i]= i**

Given an array of elements of length N, ranging from 0 to N – 1. All elements may not be present in the array. If the element is not present then there will be -1 present in the array. Rearrange the array such that A[i] = i and if i is not present, replace with -1 at that place.

**Examples:**

Input :

arr = {-1, -1, 6, 1, 9, 3, 2, -1, 4, -1}

Output : [-1, 1, 2, 3, 4, -1, 6, -1, -1, 9]

Input :

arr = {19, 7, 0, 3, 18, 15, 12, 6, 1, 8, 11, 10, 9, 5, 13, 16, 2, 14, 17, 4}

Output : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

1. Given a **sorted**array of n distinct integers where each integer is in the range from 0 to m-1 and m > n. Find the smallest number that is missing from the array.

**Examples:**

**Input:** {0, 1, 2, 6, 9}, n = 5, m = 10

**Output:** 3

**Input:** {4, 5, 10, 11}, n = 4, m = 12

**Output:** 0

**Input:** {0, 1, 2, 3}, n = 4, m = 5

**Output:** 4

**Input:** {0, 1, 2, 3, 4, 5, 6, 7, 10}, n = 9, m = 11

**Output:** 8

1. Given a sequence **arr[]** of size **n**, Write a function **int equilibrium(int[] arr, int n)** that returns an equilibrium index (if any) or -1 if no equilibrium index exists.

The **equilibrium index of an array** is an index such that the sum of elements at lower indexes is equal to the sum of elements at higher indexes.

**Examples:**

**Input**: A[] = {-7, 1, 5, 2, -4, 3, 0}   
**Output**: 3   
3 is an equilibrium index, because:   
A[0] + A[1] + A[2] = A[4] + A[5] + A[6]

**Input: A[] = {1, 2, 3}   
Output: -1**

1. Rearrange array such that even positioned are greater than odd

Given an array A of n elements, sort the array according to the following relations :

* A[i]>=A[i-1], if i is even,  ∀ 1 <= i < n
* A[i]<=A[i-1],, if i is odd ,  ∀ 1 <= i < n

Print the resultant array.

**Examples :**

Input : A[] = {1, 2, 2, 1}

Output : 1 2 1 2

Explanation :

For 1st element, 1 1, i = 2 is even.

3rd element, 1 1, i = 4 is even.

Input : A[] = {1, 3, 2}

Output : 1 3 2

Explanation :

Here, the array is also sorted as per the conditions.

1 1 and 2 < 3.