

Alternate positive and negative numbers

Easy Accuracy: 33.86% Submissions: 112K+ Points: 2



Don't Miss Out on the Chance to Work with Leading Companies! Visit the GFG Job Fair Now!

Given an unsorted array **Arr** of **N** positive and negative numbers. Your task is to create an array of alternate positive and negative numbers without changing the relative order of positive and negative numbers.

Note: Array should start with a positive number.

Example 1:

Input:

N = 9

Arr[] = {9, 4, -2, -1, 5, 0, -5, -3, 2}

Output:

9 -2 4 -1 5 -5 0 -3 2

Explanation : Positive elements : 9,4,5,0,2

Negative elements : -2,-1,-5,-3

C++ (g++ 5.4)

Average Time: 10m

[Start Timer](#)



```
1 // } Driver Code Ends
2 //User function template for C++
3 class Solution{
4 public:
5
6     int rearrange(int arr[], int n) {
7         // code here
8         vector<int>pos;
9         vector<int>neg;
10
11         for(int i = 0 ; i<n ; i++){
12             if(arr[i] >= 0){
13                 pos.push_back(arr[i]);
14             }
15             else{
16                 neg.push_back(arr[i]);
17             }
18         }
19
20         int i = 0 , j = 0 , k = 0 ;
21         while( i < pos.size() && j < neg.size()){
22             arr[k++] = pos[i++];
23             arr[k++] = neg[j++];
24         }
25         while(i < pos.size()){
26             arr[k++] = pos[i++];
27         }
28         while(j < neg.size()){
29             arr[k++] = neg[j++];
30         }
31     }
32 };
33 // } Driver Code Ends
```



Custom Input

Compile & Run

Submit

Minimum number of jumps

Medium Accuracy: 11.91% Submissions: 547K+ Points: 4



Don't Miss Out on the Chance to Work with Leading Companies! Visit the GFG Job Fair Now!

Given an array of **N** integers **arr[]** where each element represents the **maximum** length of the jump that can be made forward from that element. This means if $arr[i] = x$, then we can jump any distance y such that $y \leq x$.

Find the minimum number of jumps to reach the end of the array (starting from the first element). If an element is **0**, then you cannot move through that element.

Note: Return -1 if you can't reach the end of the array.

Example 1:

Input:

N = 11

arr[] = {1 3 5 8 9 2 6 7 6 8 9}

C++ (g++ 5.4)

Start Timer

```
1 // } Driver Code Ends
2 // Function to return minimum number of jumps to end of array
3
4
5
6
7
8
9 class Solution{
10 public:
11     int minJumps(int arr[], int n){
12         // Your code
13         if(n <= 1) return 0;
14         if(arr[0] == 0) return -1;
15
16         int curr , maxi;
17         curr = maxi = arr[0];
18         int steps = 1;
19         for(int i = 1 ; i < n ; i++){
20             if(i == n-1) return steps;
21
22             maxi = max(maxi , i+arr[i]);
23             curr--;
24
25             if(curr==0){
26                 steps++;
27                 if(i >= maxi) return -1;
28
29                 curr = maxi - i;
30             }
31         }
32         return -1;
33     }
34 };
35
36 // } Driver Code Ends
```



Custom Input

Compile & Run

Submit



Count pairs with given sum

Easy

Accuracy: 31.49%

Submissions: 272K+

Points: 2



Don't Miss Out on the Chance to Work with Leading Companies! Visit the GFG Job Fair Now!

Given an array of **N** integers, and an integer **K**, find the number of pairs of elements in the array whose sum is equal to **K**.

Example 1:

Input:

N = 4, K = 6

arr[] = {1, 5, 7, 1}

Output: 2

Explanation:

arr[0] + arr[1] = 1 + 5 = 6

and arr[1] + arr[3] = 5 + 1 = 6.

```
1 // } Driver Code Ends
2 //User function template for C++
3
4 class Solution{
5 public:
6     int getPairsCount(int arr[], int n, int k) {
7         // code here
8         unordered_map<int , int>mp;
9         int count = 0;
10        for(int i = 0;i<n;i++){
11            int b = k-arr[i];{
12                if(mp[b]){
13                    count+=mp[b];
14                }
15                mp[arr[i]]++;
16            }
17        }
18        return count;
19    }
20 };
21 // } Driver Code Ends
```