

# Prefix Sum Array - Implementation - GeeksforGeeks

**Source:** <https://www.geeksforgeeks.org/prefix-sum-array-implementation-applications-competitive-programming/>

Courses Tutorials Practice Jobs DSA Tutorial Interview Questions Quizzes Must Do Advanced DSA System Design Aptitude Puzzles Interview Corner DSA Python Technical Scripter 2026 Explore DSA Fundamentals Logic Building Problems Analysis of Algorithms Data Structures Array Data Structure String in Data Structure Hashing in Data Structure Linked List Data Structure Stack Data Structure Queue Data Structure Tree Data Structure Graph Data Structure Trie Data Structure Algorithms Searching Algorithms Sorting Algorithms Introduction to Recursion Greedy Algorithms Tutorial Graph Algorithms Dynamic Programming or DP Bitwise Algorithms Advanced Segment Tree Binary Indexed Tree or Fenwick Tree Square Root (Sqrt) Decomposition Algorithm Binary Lifting Geometry Interview Preparation Interview Corner GfG160 Practice Problem GeeksforGeeks Practice - Leading Online Coding Platform Problem of The Day - Develop the Habit of Coding DSA Course 90% Refund Prefix Sum Array - Implementation Last Updated : 13 Jul, 2025 Given an array `arr[]`, Find the prefix sum of the array. A prefix sum array is another array `prefixSum[]` of the same size, such that `prefixSum[i]` is `arr[0] + arr[1] + arr[2] . . . arr[i]`. Examples: Input: `arr[] = [10, 20, 10, 5, 15]` Output: `[10, 30, 40, 45, 60]` Explanation: For each index `i`, add all the elements from 0 to `i`: `prefixSum[0] = 10`, `prefixSum[1] = 10 + 20 = 30`, `prefixSum[2] = 10 + 20 + 10 = 40` and so on. Input: `arr[] = [30, 10, 10, 5, 50]` Output: `[30, 40, 50, 55, 105]` Explanation: For each index `i`, add all the elements from 0 to `i`: `prefixSum[0] = 30`, `prefixSum[1] = 30 + 10 = 40`, `prefixSum[2] = 30 + 10 + 10 = 50` and so on. Try it on GfG Practice Prefix Sum Implementation The idea is to create an array `prefixSum[]` of size `n`, and for each index `i` in range 1 to `n - 1`, set `prefixSum[i] = prefixSum[i - 1] + arr[i]`. To solve the problem follow the given steps: Declare a new array `prefixSum[]` of the same size as the input array Run a for loop to traverse the input array For each index add the value of the current element and the previous value of the prefix sum array Below is the implementation of the above approach: C++ `#include <iostream> #include <vector> using namespace std; // function to find the prefix sum array vector<int> prefSum ( vector<int> & arr ) { int n = arr . size (); // to store the prefix sum vector<int> prefixSum ( n ); // initialize the first element prefixSum [ 0 ] = arr [ 0 ]; // Adding present element with previous element for ( int i = 1 ; i < n ; i ++ ) prefixSum [ i ] = prefixSum [ i - 1 ] + arr [ i ]; return prefixSum ; } int main () { vector<int> arr = { 10 , 20 , 10 , 5 , 15 }; vector<int> prefixSum = prefSum ( arr ); for ( auto i : prefixSum ) { cout << i << " "; } return 0 ; }` Java `import java.util.ArrayList ; public class GfG { // function to find the prefix sum array public static ArrayList< Integer > prefSum ( int [] arr ) { int n = arr . length ; // to store the prefix sum ArrayList< Integer > prefixSum = new ArrayList<> (); // initialize the first element prefixSum . add ( arr [ 0 ] ); // Adding present element with previous element for ( int i = 1 ; i < n ; i ++ ) prefixSum . add ( prefixSum . get ( i - 1 ) + arr [ i ] ); return prefixSum ; } public static void main ( String [] args ) { int [] arr = { 10 , 20 , 10 , 5 , 15 }; ArrayList< Integer > prefixSum = prefSum ( arr ); for ( int i : prefixSum ) { System . out . print ( i + " " ); } }` Python `# function to find the prefix sum array def prefSum ( arr ): n = len ( arr ) # to store the prefix sum prefixSum = [ 0 ] * n # initialize the first element prefixSum [ 0 ] = arr [ 0 ] # Adding present element with previous element for i in range ( 1 , n ): prefixSum [ i ] = prefixSum [ i - 1 ] + arr [ i ] return prefixSum if __name__ == "__main__" : arr = [ 10 , 20 , 10 , 5 , 15 ] prefixSum = prefSum ( arr ) for i in prefixSum : print ( i , end = " " )` C# `using System ; using System.Collections.Generic ; class GfG { // function to find the prefix sum array static List< int > prefSum ( int [] arr ) { int n = arr . Length ; // to store the prefix sum List< int > prefixSum = new List< int > ( new int [ n ] ); // initialize the first element prefixSum [ 0 ] = arr [ 0 ]; // Adding present element with previous element for ( int i = 1 ; i < n ; i ++ ) prefixSum [ i ] = prefixSum [ i - 1 ] + arr [ i ]; return prefixSum ; } static void Main () { int [] arr = { 10 , 20 , 10 , 5 , 15 }; List< int > prefixSum = prefSum ( arr ); foreach ( int i in prefixSum ) { Console . Write ( i + " " ); } }` JavaScript `// function to find the prefix sum array function prefSum ( arr ) { let n = arr . length ; // to store the prefix sum let prefixSum = new Array ( n ); // initialize the first element prefixSum [ 0 ] = arr [ 0 ]; // Adding present element with previous element for ( let i = 1 ; i < n ; i ++ ) prefixSum [ i ] = prefixSum [ i - 1 ] + arr [ i ]; return prefixSum ; } // Driver Code let arr = [ 10 , 20 , 10 , 5 , 15 ]; let prefixSum = prefSum ( arr ); for ( let i of prefixSum ) { process . stdout . write ( i + " " ); } Output 10 30 40 45 60 Time Complexity: O(n) Auxiliary Space: O(n) Please refer Top Problems on Prefix Sum Technique for Interviews for more prefix sum problems.`

Comment Article Tags: Article Tags: DSA prefix-sum