



Data Structures Algorithms Interview Preparation Topic-wise Practice C++ Java Python Competitive Programming Machi

Given an array A[] and a number x, check for pair in A[] with sum as x (aka Two Sum)

Difficulty Level : Easy • Last Updated : 19 Jan, 2022

Write a program that, given an array A[] of n numbers and another number x, determines whether or not there exist two elements in A[] whose sum is exactly x.

Examples:

Input: arr[] = {0, -1, 2, -3, 1}
sum = -2

Output: -3, 1
Valid pair exists.

If we calculate the sum of the output,
 $1 + (-3) = -2$

Input: arr[] = {1, -2, 1, 0, 5}
sum = 0



Start Your Coding Journey Now!

[Login](#)[Register](#)

solution.

Method: Using simple logic by calculating array's elements itself.

C++

```
/*
 * This C++ program tells if there exists a pair in array whose sum results in x.
 */

#include <iostream>

using namespace std;

// Function to find and print pair
bool chkPair(int A[], int size, int x) {
    for (int i = 0; i < (size - 1); i++) {
        for (int j = (i + 1); j < size; j++) {
            if (A[i] + A[j] == x) {
                cout << "Pair with a given sum " << x << " is (" << A[i] << ", " << A[j] << ")"
                    << endl;

                return 1;
            }
        }
    }
}
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
int x = -2;
int size = sizeof(A) / sizeof(A[0]);

if (chkPair(A, size, x)) {
    cout << "Valid pair exists" << endl;
}
else {
    cout << "No valid pair exists for " << x << endl;
}

return 0;
}

// This code is contributed by Manish Kumar (mkumar2789)
```

Java

```
// Java program to check if there exists a pair
// in array whose sum results in x.
class GFG{

    // Function to find and print pair
    static boolean chkPair(int A[], int size, int x) {
        for (int i = 0; i < (size - 1); i++) {
            for (int j = (i + 1); j < size; j++) {
                if (A[i] + A[j] == x) {
                    System.out.println("Pair with a given sum " + x +
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
}  
  
    return false;  
}  
  
public static void main(String [] args) {  
  
    int A[] = {0, -1, 2, -3, 1};  
    int x = -2;  
    int size = A.length;  
  
    if (chkPair(A, size, x)) {  
        System.out.println("Valid pair exists");  
    }  
    else {  
        System.out.println("No valid pair exists for " + x );  
    }  
}  
}  
  
// This code is contributed by AR_Gaurav
```

Python3



```
# Python program to check if there exists a pair  
# in array whose sum results in x.
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
        return True
    return False

# Driver code
A = [0, -1, 2, -3, 1]
x = -2
size = len(A)

if (chkPair(A, size, x)) :
    print("Valid pair exists")

else :
    print("No valid pair exists for " , x )

# This code is contributed by AR_Gaurav
```

C#

```
// C# program to check if there exists a pair
// in array whose sum results in x.
using System;
class GFG{

    // Function to find and print pair
    static bool chkPair(int [] A, int size, int x) {
        for (int i = 0; i < (size - 1); i++) {
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
        return true;
    }
}

return false;
}

public static void Main()
{
    int [] A = {0, -1, 2, -3, 1};
    int x = -2;
    int size = A.Length;

    if (chkPair(A, size, x)) {
        Console.WriteLine("Valid pair exists");
    }
    else {
        Console.WriteLine("No valid pair exists for " + x );
    }
}
}
```

// This code is contributed by AR_Gaurav



Javascript

Start Your Coding Journey Now!

Login

Register

```

    for (j = (i + 1); j < size; j++) {
        if (A[i] + A[j] == x) {
            document.write("Pair with a given sum " + x + " is (" + A[i] + ", " + A[j] + ")");

            return true;
        }
    }

    return false;
}

var A = [ 0, -1, 2, -3, 1 ];
var x = -2;
var size = A.length;

if (chkPair(A, size, x)) {
    document.write("<br/>Valid pair exists");
} else {
    document.write("<br/>No valid pair exists for " + x);
}

This code is contributed by umadevi9616
</pre>

```

Output

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// C++ program for the above approach
#include <bits/stdc++.h>
using namespace std;

// Function to find and print pair
bool chkPair(int A[], int size, int x) {
    for (int i = 0; i < (size - 1); i++) {
        for (int j = (i + 1); j < size; j++) {
            if (A[i] + A[j] == x) {
                cout << "Pair with a given sum " << x << " is (" << A[i] << ", " << A[j] << ")"
                    << endl;

                return 1;
            }
        }
    }

    return 0;
}

int main() {
    int A[] = {0, -1, 2, -3, 1};
    int x = -2;
    int size = sizeof(A) / sizeof(A[0]);

    if (chkPair(A, size, x)) {
        cout << "Valid pair exists" << endl;
    }
}
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
}
```

```
// This code is contributed by Samim Hossain Mondal.
```

C

```
/*  
 * This C program tells if there exists a pair in array whose sum results in x.  
 */
```

```
#include <stdio.h>
```

```
// Function to find and print pair
```

```
int chkPair(int A[], int size, int x) {  
    for (int i = 0; i < (size - 1); i++) {  
        for (int j = (i + 1); j < size; j++) {  
            if (A[i] + A[j] == x) {  
                printf("Pair with a given sum %d is (%d, %d)\n", x, A[i], A[j]);  
  
                return 1;  
            }  
        }  
    }  
  
    return 0;  
}
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
printf("Valid pair exists\n");
}
else {
    printf("No valid pair exists for %d\n", x);
}

return 0;
}
// This code is contributed by Manish Kumar (mkumar2789)
```

Java

```
// Java program to check if there exists a pair
// in array whose sum results in x.
class GFG{

    // Function to find and print pair
    static boolean chkPair(int A[], int size, int x) {
        for (int i = 0; i < (size - 1); i++) {
            for (int j = (i + 1); j < size; j++) {
                if (A[i] + A[j] == x) {
                    System.out.println("Pair with a given sum " + x +
                                       " is (" + A[i] + ", " + A[j] + ")");

                    return true;
                }
            }
        }
    }
}
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
int A[] = {0, -1, 2, -3, 1};
int x = -2;
int size = A.length;

if (chkPair(A, size, x)) {
    System.out.println("Valid pair exists");
}
else {
    System.out.println("No valid pair exists for " + x );
}
}
```

// This code is contributed by umadevi9616

Python3

This python program tells if there exists a pair in array whose sum results in x.

Function to find and print pair

```
def chkPair(A, size, x):
    for i in range(0, size - 1):
        for j in range(i + 1, size):
            if (A[i] + A[j] == x):
                print(f"Pair with a given sum {x} is ({A[i]},{A[j]})")
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
size = len(A)

if (chkPair(A, size, x)):
    print("Valid pair exists")

else:
    print(f"No valid pair exists for {x}")

# This code is contributed by rakeshsahni
```

C#

```
// C# program to check if there exists a pair
// in array whose sum results in x.
using System;
class GFG{

// Function to find and print pair
static bool chkPair(int [] A, int size, int x) {
    for (int i = 0; i < (size - 1); i++) {
        for (int j = (i + 1); j < size; j++) {
            if (A[i] + A[j] == x) {
                Console.WriteLine("Pair with a given sum " + x +
                                " is (" + A[i] + ", " + A[j] + ")");
            }
        }
    }

    return true;
}
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
public static void Main()  
{  
    int [] A = {0, -1, 2, -3, 1};  
    int x = -2;  
    int size = A.Length;  
  
    if (chkPair(A, size, x)) {  
        Console.WriteLine("Valid pair exists");  
    }  
    else {  
        Console.WriteLine("No valid pair exists for " + x );  
    }  
}
```

// This code is contributed by Samim Hossain Mondal.

Javascript

<script>

// Javascript program to check if there exists a pair
// in array whose sum results in x.

// Function to find and print pair
function chkPair(A , size , x) {

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
        return true;
    }
}

return false;
}

let A = [ 0, -1, 2, -3, 1 ];
let x = -2;
let size = A.length;

if (chkPair(A, size, x)) {
    document.write("<br/>Valid pair exists");
}
else {
    document.write("<br/>No valid pair exists for " + x);
}
```

```
// This code is contributed by Samim Hossain Mondal.
</script>
```



Output

Start Your Coding Journey Now!

[Login](#)[Register](#)

Auxiliary Space: $O(1)$

Method 1: Sorting and Two-Pointers technique.

Approach: A tricky approach to solve this problem can be to use the two-pointer technique. But for using two pointer technique, the array must be sorted. Once the array is sorted the two pointers can be taken which mark the beginning and end of the array respectively. If the sum is **greater** than the sum of those two elements, shift the right pointer to decrease the value of required sum and if the sum is **lesser** than the required value, shift the left pointer to increase the value of the required sum. Let's understand this using an example.

Let an array be {1, 4, 45, 6, 10, -8} and sum to find be 16

After sorting the array

$A = \{-8, 1, 4, 6, 10, 45\}$

Now, increment 'l' when the sum of the pair is less than the required sum and decrement 'r' when the sum of the pair is more than the required sum.

This is because when the sum is less than the required sum then to get the number which could increase the sum of pair, start moving from left to right (also sort the array) thus "l++" and vice versa.

Initialize $l = 0, r = 5$



Start Your Coding Journey Now!

[Login](#)[Register](#)

$A[l] + A[r] (6 + 10) == 16 \Rightarrow \text{Found candidates (return 1)}$

Note: If there is more than one pair having the given sum then this algorithm reports only one. Can be easily extended for this though.

Algorithm:

1. hasArrayTwoCandidates (A[], ar_size, sum)
2. Sort the array in non-decreasing order.
3. Initialize two index variables to find the candidate elements in the sorted array.
 1. Initialize first to the leftmost index: $l = 0$
 2. Initialize second the rightmost index: $r = \text{ar_size} - 1$
4. Loop while $l < r$.
 1. If $(A[l] + A[r] == \text{sum})$ then return 1
 2. Else if $(A[l] + A[r] < \text{sum})$ then $l++$
 3. Else $r--$
5. No candidates in the whole array – return 0



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// to the given value
```

```
#include <bits/stdc++.h>
using namespace std;

// Function to check if array has 2 elements
// whose sum is equal to the given value
bool hasArrayTwoCandidates(int A[], int arr_size,
                           int sum)
{
    int l, r;

    /* Sort the elements */
    sort(A, A + arr_size);

    /* Now look for the two candidates in
       the sorted array*/
    l = 0;
    r = arr_size - 1;
    while (l < r) {
        if (A[l] + A[r] == sum)
            return 1;
        else if (A[l] + A[r] < sum)
            l++;
        else // A[i] + A[j] > sum
            r--;
    }
    return 0;
}
```



Login

Register

C

```
bool hasArrayTwoCandidates(
    int A[], int arr_size, int sum)
{
    int l, r;
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
int isTwoSum(int arr[], int sum)
{
    l = 0;
    r = arr_size - 1;
    while (l < r) {
        if (A[l] + A[r] == sum)
            return 1;
        else if (A[l] + A[r] < sum)
            l++;
        else // A[i] + A[j] > sum
            r--;
    }
    return 0;
}

/* FOLLOWING FUNCTIONS ARE ONLY FOR SORTING
PURPOSE */
void exchange(int* a, int* b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

int partition(int A[], int si, int ei)
{
    int x = A[ei];
    int i = (si - 1);
    int j;
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
,
exchange(&A[i + 1], &A[ei]);
return (i + 1);
}

/* Implementation of Quick Sort
A[] --> Array to be sorted
si --> Starting index
ei --> Ending index
*/
void quickSort(int A[], int si, int ei)
{
    int pi; /* Partitioning index */
    if (si < ei) {
        pi = partition(A, si, ei);
        quickSort(A, si, pi - 1);
        quickSort(A, pi + 1, ei);
    }
}

/* Driver program to test above function */
int main()
{
    int A[] = { 1, 4, 45, 6, 10, -8 };
    int n = 16;
    int arr_size = 6;

    if (hasArrayTwoCandidates(A, arr_size, n))
        printf("Array has two elements with given sum");
    else
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

Java

```
// Java program to check if given array
// has 2 elements whose sum is equal
// to the given value
import java.util.*;

class GFG {
    // Function to check if array has 2 elements
    // whose sum is equal to the given value
    static boolean hasArrayTwoCandidates(
        int A[],
        int arr_size, int sum)
    {
        int l, r;

        /* Sort the elements */
        Arrays.sort(A);

        /* Now look for the two candidates
        in the sorted array*/
        l = 0;
        r = arr_size - 1;
        while (l < r) {
            if (A[l] + A[r] == sum)
                return true;
            else if (A[l] + A[r] < sum)
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// Driver code
public static void main(String args[])
{
    int A[] = { 1, 4, 45, 6, 10, -8 };
    int n = 16;
    int arr_size = A.length;

    // Function calling
    if (hasArrayTwoCandidates(A, arr_size, n))
        System.out.println("Array has two "
                           + "elements with given sum");
    else
        System.out.println("Array doesn't have "
                           + "two elements with given sum");
}
```

Python

```
# Python program to check for the sum
# condition to be satisfied
```

```
def hasArrayTwoCandidates(A, arr_size, sum):

    # sort the array
    quickSort(A, 0, arr_size-1)
```

Start Your Coding Journey Now!

[Login](#)
[Register](#)

```

    return 1
elif (A[l] + A[r] < sum):
    l += 1
else:
    r -= 1
return 0

```

Implementation of Quick Sort

A[] --> Array to be sorted

si --> Starting index

ei --> Ending index

```

def quickSort(A, si, ei):
    if si < ei:
        pi = partition(A, si, ei)
        quickSort(A, si, pi-1)
        quickSort(A, pi + 1, ei)

```

Utility function for partitioning

the array(used in quick sort)

```

def partition(A, si, ei):
    x = A[ei]
    i = (si-1)
    for j in range(si, ei):
        if A[j] <= x:
            i += 1

```

```

    # This operation is used to swap
    # two variables in python
    A[i], A[j] = A[j], A[i]

```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
# Driver program to test the functions
A = [1, 4, 45, 6, 10, -8]
n = 16
if (hasArrayTwoCandidates(A, len(A), n)):
    print("Array has two elements with the given sum")
else:
    print("Array doesn't have two elements
          with the given sum")

## This code is contributed by __Devesh Agrawal__
```

C#

```
// C# program to check for pair
// in A[] with sum as x

using System;

class GFG {
    static bool hasArrayTwoCandidates(int[] A,
                                      int arr_size, int sum)
    {
        int l, r;

        /* Sort the elements */
        sort(A, 0, arr_size - 1);
```


Start Your Coding Journey Now!

[Login](#)[Register](#)

```
        return true;
    else if (A[l] + A[r] < sum)
        l++;
    else // A[i] + A[j] > sum
        r--;
    }
    return false;
}

/* Below functions are only to sort the
array using QuickSort */

/* This function takes last element as pivot,
places the pivot element at its correct
position in sorted array, and places all
smaller (smaller than pivot) to left of
pivot and all greater elements to right
of pivot */
static int partition(int[] arr, int low, int high)
{
    int pivot = arr[high];

    // index of smaller element
    int i = (low - 1);
    for (int j = low; j <= high - 1; j++) {
        // If current element is smaller
        // than or equal to pivot
        if (arr[j] <= pivot) {
            i++;
        }
    }
    // swap arr[i] and arr[high]
    int temp = arr[i];
    arr[i] = arr[high];
    arr[high] = temp;
    return i;
}
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
}  
  
// swap arr[i+1] and arr[high] (or pivot)  
int temp1 = arr[i + 1];  
arr[i + 1] = arr[high];  
arr[high] = temp1;  
  
return i + 1;  
}  
  
/* The main function that  
implements QuickSort()  
arr[] --> Array to be sorted,  
low --> Starting index,  
high --> Ending index */  
static void sort(int[] arr, int low, int high)  
{  
    if (low < high) {  
        /* pi is partitioning index, arr[pi]  
        is now at right place */  
        int pi = partition(arr, low, high);  
  
        // Recursively sort elements before  
        // partition and after partition  
        sort(arr, low, pi - 1);  
        sort(arr, pi + 1, high);  
    }  
}
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
if (hasArrayTwoCandidates(A, arr_size, n))
    Console.WriteLine("Array has two elements"
        + " with given sum");
else
    Console.WriteLine("Array doesn't have "
        + "two elements with given sum");
}
```

// This code is contributed by Sam007

PHP

```
<?php
// PHP program to check if given
// array has 2 elements whose sum
// is equal to the given value

// Function to check if array has
// 2 elements whose sum is equal
// to the given value
function hasArrayTwoCandidates($A, $arr_size,
                                $sum)
{
    $l; $r;
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
... int arr_size, sum) {
    $l = 0;
    $r = $arr_size - 1;
    while ($l < $r)
    {
        if($A[$l] + $A[$r] == $sum)
            return 1;
        else if($A[$l] + $A[$r] < $sum)
            $l++;
        else // A[i] + A[j] > sum
            $r--;
    }
    return 0;
}

// Driver Code
$A = array (1, 4, 45, 6, 10, -8);
$n = 16;
$arr_size = sizeof($A);

// Function calling
if(hasArrayTwoCandidates($A, $arr_size, $n))
    echo "Array has two elements " .
        "with given sum";
else
    echo "Array doesn't have two " .
        "elements with given sum";

// This code is contributed by m_kit
?>
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// Javascript program to check if given array
// has 2 elements whose sum is equal
// to the given value

// Function to check if array has 2 elements
// whose sum is equal to the given value
function hasArrayTwoCandidates(A, arr_size, sum)
{
    var l, r;

    /* Sort the elements */
    A.sort();

    /* Now look for the two candidates in
    the sorted array*/
    l = 0;
    r = arr_size - 1;
    while (l < r) {
        if (A[l] + A[r] == sum)
            return 1;
        else if (A[l] + A[r] < sum)
            l++;
        else // A[i] + A[j] > sum
            r--;
    }
    return 0;
}
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// JavaScript code to check for pair in array with sum as x
document.write("Array has two elements" +
    " with the given sum");
else
    document.write("Array doesn't have two" +
        " elements with the given sum");
</script>
```

Output

Array has two elements with given sum

Complexity Analysis:

- **Time Complexity:** Depends on what sorting algorithm we use.
 - If Merge Sort or Heap Sort is used then $O(n \log n)$ in the worst case.
 - If Quick Sort is used then $O(n^2)$ in the worst case.
- **Auxiliary Space:** This too depends on sorting algorithm. The auxiliary space is $O(n)$ for merge sort and $O(1)$ for Heap Sort.

Method 2: [Hashing](#).



Approach: This problem can be solved efficiently by using the technique of hashing. Use a **hash_map** to check for the current array value **x(let)**, if there exists a value **target_sum-x** which on adding to the

Start Your Coding Journey Now!

[Login](#)[Register](#)

Now start traversing:

Step 1: For '0' there is no valid number '-2' so store '0' in hash_map.

Step 2: For '-1' there is no valid number '-1' so store '-1' in hash_map.

Step 3: For '2' there is no valid number '-4' so store '2' in hash_map.

Step 4: For '-3' there is no valid number '1' so store '-3' in hash_map.

Step 5: For '1' there is a valid number '-3' so answer is 1, -3

Algorithm:

1. Initialize an empty hash table s.
2. Do following for each element A[i] in A[]
 1. If s[x - A[i]] is set then print the pair (A[i], x - A[i])
 2. Insert A[i] into s.

Pseudo Code :

```
unordered_set s
for(i=0 to end)
    if(s.find(target_sum - arr[i]) == s.end)
        insert(arr[i] into s)
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// C++ program to check if given array
// has 2 elements whose sum is equal
// to the given value
#include <bits/stdc++.h>

using namespace std;

void printPairs(int arr[], int arr_size, int sum)
{
    unordered_set<int> s;
    for (int i = 0; i < arr_size; i++)
    {
        int temp = sum - arr[i];

        if (s.find(temp) != s.end())
            cout << "Pair with given sum "
                 << sum << " is (" << arr[i] << ", "
                 << temp << ")" << endl;

        s.insert(arr[i]);
    }
}

/* Driver Code */
int main()
{
    int A[] = { 1, 4, 45, 6, 10, 8 };
    int n = 16;
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
}  
}
```

C

```
// C program to check if given array  
// has 2 elements whose sum is equal  
// to the given value  
  
// Works only if range elements is limited  
#include <stdio.h>  
#define MAX 100000  
  
void printPairs(int arr[], int arr_size, int sum)  
{  
    int i, temp;  
  
    /*initialize hash set as 0*/  
    bool s[MAX] = { 0 };  
  
    for (i = 0; i < arr_size; i++)  
    {  
        temp = sum - arr[i];  
        if (s[temp] == 1)  
            printf(  
                "Pair with given sum %d is (%d, %d) n",  
                sum, arr[i], temp);  
        s[arr[i]] = 1;  
    }  
}
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
int A[] = { 1, 4, 45, 6, 10, 8 };
int n = 16;
int arr_size = sizeof(A) / sizeof(A[0]);

printPairs(A, arr_size, n);

getchar();
return 0;
}
```

Java

```
// Java implementation using Hashing
import java.io.*;
import java.util.HashSet;

class PairSum {
    static void printpairs(int arr[], int sum)
    {
        HashSet<Integer> s = new HashSet<Integer>();
        for (int i = 0; i < arr.length; ++i)
        {
            int temp = sum - arr[i];

            // checking for condition
            if (s.contains(temp)) {
                System.out.println(
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
}  
  
// Driver Code  
public static void main(String[] args)  
{  
    int A[] = { 1, 4, 45, 6, 10, 8 };  
    int n = 16;  
    printpairs(A, n);  
}  
}  
  
// This article is contributed by Aakash Hasiya
```

Python3

```
# Python program to find if there are  
# two elements with given sum  
  
# function to check for the given sum  
# in the array  
def printPairs(arr, arr_size, sum):  
  
    # Create an empty hash map  
    # using a hashmap allows us to store the indices  
    hashmap = {}  
  
    for i in range(0, arr_size):
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
A = [1, 4, 45, 6, 10, 8]
```

```
n = 16
```

```
printPairs(A, len(A), n)
```

```
# This code will also work in case the array has the same number twice
```

```
# and target is the sum of those numbers
```

```
# Eg: Array = [4,6,4] Target = 8
```

```
# This code is contributed by __Achyut Upadhyay__
```

C#

```
// C# implementation using Hashing
```

```
using System;
```

```
using System.Collections.Generic;
```

```
class GFG {
```

```
    static void printpairs(int[] arr,  
                           int sum)
```

```
{
```

```
    HashSet<int> s = new HashSet<int>();
```

```
    for (int i = 0; i < arr.Length; ++i)
```

```
{
```

```
        int temp = sum - arr[i];
```

```
        // checking for condition
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
}  
  
// Driver Code  
static void Main()  
{  
    int[] A = new int[] { 1, 4, 45,  
                          6, 10, 8 };  
  
    int n = 16;  
    printpairs(A, n);  
}  
}  
  
// This code is contributed by  
// Manish Shaw(manishshaw1)
```

Javascript

<script>

```
// JavaScript program to check if given array  
// has 2 elements whose sum is equal  
// to the given value
```

```
// Javascript implementation using Hashing
```

```
function printpairs(arr, sum)
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// checking for condition
if (s.has(temp)) {
    document.write(
        "Pair with given sum "
        + sum + " is (" + arr[i]
        + ", " + temp + ")");
}
s.add(arr[i]);
}
```

// Driver Code

```
let A = [ 1, 4, 45, 6, 10, 8 ];
let n = 16;
printpairs(A, n);
```

</script>

Output:

Pair with given sum 16 is (6,10) at indices (3,4)



Complexity Analysis:

Start Your Coding Journey Now!

[Login](#)[Register](#)

Note: The solution will work even if the range of numbers includes negative numbers + if the pair is formed by numbers recurring twice in array eg: array = [3,4,3]; pair = (3,3); target sum = 6.

Method 3: Using remainders of the elements less than x.

Approach:

The idea is to count the elements with remainders when divided by x, i.e **0 to x-1**, each remainder separately. Suppose we have **x as 6**, then the numbers which are less than 6 and have remainders which add up to 6 gives sum as 6 when added. For example, we have elements, 2,4 in the array and $2\%6 = 2$ and $4\%6 = 4$, and these remainders add up to give 6. Like that we have to check for pairs with remainders (1,5), (2,4), (3,3). if we have one or more elements with remainder 1 and one or more elements with remainder 5, then surely we get a sum as 6. Here we do not consider (0,6) as the elements for the resultant pair should be less than 6. when it comes to (3,3) we have to check if we have two elements with remainder 3, then we can say that "There exists a pair whose sum is x".

Algorithm:

1. Create an array with size x.
2. Initialize all rem elements to zero.
3. Traverse the given array
 - Do the following if arr[i] is less than x:



Start Your Coding Journey Now!

[Login](#)[Register](#)

- If($\text{rem}[i] > 0$ and $\text{rem}[x-i] > 0$) then print "YES" and come out of the loop. This means that we have a pair that results in x upon doing.

5. Now when we reach at $x/2$ in the above loop

- If x is even, for getting a pair we should have two elements with remainder $x/2$.
 - If $\text{rem}[x/2] > 1$ then print "YES" else print "NO"
- If it is not satisfied that is x is odd, it will have a separate pair with $x-x/2$.
 - If $\text{rem}[x/2] > 1$ and $\text{rem}[x-x/2] > 1$, then print "Yes" else, print "No";

Implementation of the above algorithm:

C++

```
// Code in cpp to tell if there
// exists a pair in array whose
// sum results in x.
#include <iostream>
using namespace std;

// Function to print pairs
void printPairs(int a[], int n, int x)
{
    int i;
```


Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// Traverse the array
rem[i] = 0;
}
for (i = 0; i < n; i++)
{
    if (a[i] < x)
    {
        // Perform the remainder
        // operation only if the
        // element is x, as numbers
        // greater than x can't
        // be used to get a sum x.
        // Updating the count of remainders.
        rem[a[i] % x]++;
    }
}

// Traversing the remainder list
// from start to middle to
// find pairs
for (i = 1; i < x / 2; i++)
{
    if (rem[i] > 0 && rem[x - i] > 0)
    {
        // The elements with remainders
        // i and x-i will
        // result to a sum of x.
        // Once we get two
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
        },
    }
}

// Once we reach middle of
// remainder array, we have to
// do operations based on x.
if (i >= x / 2)
{
    if (x % 2 == 0)
    {
        if (rem[x / 2] > 1)
        {

            // if x is even and
            // we have more than 1
            // elements with remainder
            // x/2, then we will
            // have two distinct elements
            // which add up
            // to x. if we dont have
            // more than 1
            // element, print "No".
            cout << "Yes"
                << "\n";

        }
    }
    else
    {
        cout << "No"
            << "\n";
    }
}
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// the same process
// which we did in previous loop.
if (rem[x / 2] > 0 &&
    rem[x - x / 2] > 0)
{
    cout << "Yes"
        << "\n";
}
else
{
    cout << "No"
        << "\n";
}
}
}

/* Driver Code */
int main()
{
    int A[] = { 1, 4, 45, 6, 10, 8 };
    int n = 16;
    int arr_size = sizeof(A) / sizeof(A[0]);

    // Function calling
    printPairs(A, arr_size, n);

    return 0;
}
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// Code in Java to tell if there
// exists a pair in array whose
// sum results in x.
import java.util.*;
class GFG{

// Function to print pairs
static void printPairs(int a[], int n, int x)
{
    int i;
    int []rem = new int[x];
    for (i = 0; i < x; i++)
    {

        // initializing the rem
        // values with 0's.
        rem[i] = 0;
    }
    for (i = 0; i < n; i++)
    {
        if (a[i] < x)
        {

            // Perform the remainder
            // operation only if the
            // element is x, as numbers
            // greater than x can't
            // be used to get a sum x.
            // Updating the count of remainders.
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// ...  
// find pairs  
for (i = 1; i < x / 2; i++)  
{  
    if (rem[i] > 0 && rem[x - i] > 0)  
    {  
  
        // The elements with remainders  
        // i and x-i will  
        // result to a sum of x.  
        // Once we get two  
        // elements which add up to x ,  
        // we print x and  
        // break.  
        System.out.print("Yes"  
                        + "\n");  
  
        break;  
    }  
}  
  
// Once we reach middle of  
// remainder array, we have to  
// do operations based on x.  
if (i >= x / 2)  
{  
    if (x % 2 == 0)  
    {  
        if (rem[x / 2] > 1)  
        {
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// When x is even we continue the process to x. if we dont have
//more than 1
// element, print "No".
System.out.print("Yes"
                  + "\n");
}
else
{
    System.out.print("No"
                    + "\n");
}
}
else
{
    // When x is odd we continue
    // the same process
    // which we did in previous loop.
    if (rem[x / 2] > 0 &&
        rem[x - x / 2] > 0)
    {
        System.out.print("Yes"
                        + "\n");
    }
    else
    {
        System.out.print("No"
                        + "\n");
    }
}
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// C++ program to find pairs in an array whose sum is equal to x
{
    int A[] = { 1, 4, 45, 6, 10, 8 };
    int n = 16;
    int arr_size = A.length;

    // Function calling
    printPairs(A, arr_size, n);
}
}
```

// This code is contributed by aashish1995

Python3

```
# Code in Python3 to tell if there
# exists a pair in array whose
# sum results in x.
```

```
# Function to print pairs
```

```
def printPairs(a, n, x):
```

```
    rem = []
```

```
    for i in range(x):
```

```
        # Initializing the rem
        # values with 0's.
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
... # remove the remainder operation...
# only if the element is x, as
# numbers greater than x can't
# be used to get a sum x.Updating
# the count of remainders.
rem[a[i] % x] += 1

# Traversing the remainder list from
# start to middle to find pairs
for i in range(1, x // 2):
    if (rem[i] > 0 and rem[x - i] > 0):

        # The elements with remainders
        # i and x-i will result to a
        # sum of x. Once we get two
        # elements which add up to x,
        # we print x and break.
        print("Yes")
        break

# Once we reach middle of
# remainder array, we have to
# do operations based on x.
if (i >= x // 2):
    if (x % 2 == 0):
        if (rem[x // 2] > 1):

            # If x is even and we have more
            # than 1 elements with remainder
            # x/2, then we will have two
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
else:
```

```
    # When x is odd we continue  
    # the same process which we  
    # did in previous loop.
```

```
    if (rem[x // 2] > 0 and  
        rem[x - x // 2] > 0):  
        print("Yes")
```

```
    else:  
        print("No")
```

```
# Driver Code
```

```
A = [ 1, 4, 45, 6, 10, 8 ]
```

```
n = 16
```

```
arr_size = len(A)
```

```
# Function calling
```

```
printPairs(A, arr_size, n)
```

```
# This code is contributed by subhammahato348
```

C#

```
// C# Code in C# to tell if there  
// exists a pair in array whose  
// sum results in x.  
using System;
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
int i;
int []rem = new int[x];
for (i = 0; i < x; i++)
{

    // initializing the rem
    // values with 0's.
    rem[i] = 0;
}
for (i = 0; i < n; i++)
{
    if (a[i] < x)
    {

        // Perform the remainder
        // operation only if the
        // element is x, as numbers
        // greater than x can't
        // be used to get a sum x.
        // Updating the count of remainders.
        rem[a[i] % x]++;
    }
}

// Traversing the remainder list
// from start to middle to
// find pairs
for (i = 1; i < x / 2; i++)
{
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// Once we get two
// elements which add up to x ,
// we print x and
// break.
Console.WriteLine("Yes" + "\n");
break;
}
}

// Once we reach middle of
// remainder array, we have to
// do operations based on x.
if (i >= x / 2)
{
    if (x % 2 == 0)
    {
        if (rem[x / 2] > 1)

            // if x is even and
            // we have more than 1
            // elements with remainder
            // x/2, then we will
            // have two distinct elements
            // which add up
            // to x. if we dont have
            // more than 1
            // element, print "No".
        Console.WriteLine("Yes" + "\n");
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
,
}
else
{

    // When x is odd we continue
    // the same process
    // which we did in previous loop.
    if (rem[x / 2] > 0 &&
        rem[x - x / 2] > 0)
    {
        Console.Write("Yes"
                      + "\n");
    }
    else
    {
        Console.WriteLine("No"
                          + "\n");
    }
}
}
}

/* Driver Code */
public static void Main(string[] args)
{
    int[] A = { 1, 4, 45, 6, 10, 8 };
    int n = 16;
    int arr_size = A.Length;
```

Start Your Coding Journey Now!

[Login](#)[Register](#)

Javascript

```
<script>
```

```
// Code in Javascript to tell if there  
// exists a pair in array whose  
// sum results in x.
```

```
// Function to print pairs
```

```
function printPairs(a, n, x)
```

```
{
```

```
    let i;
```

```
    let rem = new Array(x);
```

```
    for(i = 0; i < x; i++)
```

```
    {
```

```
        // Initializing the rem
```

```
        // values with 0's.
```

```
        rem[i] = 0;
```

```
    }
```

```
    for(i = 0; i < n; i++)
```

```
    {
```

```
        if (a[i] < x)
```

```
        {
```

```
            // Perform the remainder
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
}  
}  
  
// Traversing the remainder list  
// from start to middle to  
// find pairs  
for(i = 1; i < x / 2; i++)  
{  
    if (rem[i] > 0 && rem[x - i] > 0)  
    {  
  
        // The elements with remainders  
        // i and x-i will  
        // result to a sum of x.  
        // Once we get two  
        // elements which add up to x ,  
        // we print x and  
        // break.  
        document.write("Yes" + "</br>");  
        break;  
    }  
}  
  
// Once we reach middle of  
// remainder array, we have to  
// do operations based on x.  
if (i >= x / 2)  
{  
    if (x % 2 == 0)
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// elements with remainder
// x/2, then we will
// have two distinct elements
// which add up
// to x. if we dont have
// more than 1
// element, print "No".
document.write("Yes" + "</br>");
}
else
{
    document.write("No" + "</br>");
}
}
else
{

    // When x is odd we continue
    // the same process
    // which we did in previous loop.
    if (rem[x / 2] > 0 &&
        rem[x - x / 2] > 0)
    {
        document.write("Yes" + "</br>");
    }
    else
    {
        document.write("No" + "</br>");
    }
}
```



Start Your Coding Journey Now!

[Login](#)[Register](#)

```
let n = 16;  
let arr_size = A.length;  
  
// Function calling  
printPairs(A, arr_size, n);  
  
// This code is contributed by suresh07  
  
</script>
```

Output

Yes

Time Complexity: $O(n+x)$

Auxiliary Space: $O(x)$

Similarly the indices of a pair that add upto a given sum can also be calculated by unordered map. The only change here is that we also have to store indices of elements as value for each element as key.

C++14

```
#include <bits/stdc++.h>  
using namespace std;  
  
pair<int,int> findSum(int *arr,int& n,int& target)
```


Start Your Coding Journey Now!

[Login](#)[Register](#)

```
        findElement=target-arr[i];
        if(mp[findElement])
        {
            result.first=i-1;
            result.second=mp[findElement]-1;
            break;
        }
        else mp.insert({arr[i],i});
    }
    return result;
}

int main()
{
    int arr[]={1,5,4,3,7,9,2};
    int n=sizeof(arr)/sizeof(arr[0]);
    int search=7;
    pair<int,int>ans=findSum(arr,n,search);
    cout<<min(ans.first,ans.second)<<" "<<max(ans.first,ans.second);
    return 0;
}
```

Time Complexity: $O(n)$

Auxiliary Space: $O(n)$

Related Problems:



Start Your Coding Journey Now!

[Login](#)[Register](#)

... a number x, check for pair in A[] with sum as x | GeeksforGeeks



Please write comments if you find any of the above codes/algorithms incorrect, or find other ways to solve the same problem.



FREE CONTEST

Bi-Wizard 2.0

School Coding Contest

Exclusively for students of classes 8th-12th

Rewards upto **INR 5 Lacs**

 **26 Feb, 6 PM IST**

Start Your Coding Journey Now!

Login

Register

LINK

Previous

Union and Intersection of two linked lists |
Set-3 (Hashing)

Next

Graph and its representations

RECOMMENDED ARTICLES

Page :

Article Contributed By :



GeeksforGeeks

Start Your Coding Journey Now!

[Login](#)[Register](#)

Improved By : jit_t, manishshaw1, Mayuri Gupta 4, hellokoding, salonitayal, bidibaaz123, SaiSanjanaGudla, idkravitz, aashish1995, SoumikMondal, subhammahato348, shrey3108, suresh07, rrrtnx, sg275527109, sanjoy_62, rajeev0719singh, mdtauseef123, mkumar2789, AR_Gaurav, umadevi9616, rakeshsahni, surinderdawra388, samim2000, kk9826225, bml8ngqtq8nb1h971tf2wo5m68l4m0nmykp8nxmq, prophet1999

Article Tags : ABCO, Accolite, Amazon, Amazon-Question, CarWale, CarWale-Question, FactSet, Flipkart, Hike, Infosys, Microsoft, Morgan Stanley, SAP Labs, Wipro, Zoho, Arrays, Backtracking, Hash, Recursion

Practice Tags : Infosys, Zoho, Flipkart, Morgan Stanley, Accolite, Amazon, Microsoft, FactSet, Hike, ABCO, Wipro, SAP Labs, CarWale, Arrays, Hash, Recursion, Backtracking

[Report Issue](#)

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

[Load Comments](#)

Start Your Coding Journey Now!

Login

Register

GeeksforGeeks, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

About Us

Careers

Privacy Policy

Contact Us

Copyright Policy

Learn

Algorithms

Data Structures

Languages

CS Subjects

Video Tutorials

Web Development

Web Tutorials

HTML

CSS

JavaScript

Bootstrap

Contribute

Write an Article

Write Interview Experience

Internships

Videos

Start Your Coding Journey Now!

Login

Register