Print all pairs with given sum

Difficulty Level : Easy • Last Updated : 01 Jul, 2021

Given an array of integers, and a number 'sum', print all pairs in the array whose sum is equal to 'sum'.

```
Examples :
Input  :  arr[] = {1, 5, 7, -1, 5},
          sum = 6
Output : (1, 5) (7, -1) (1, 5)

Input  :  arr[] = {2, 5, 17, -1},
          sum = 7
Output :  (2, 5)
```

Recommended: Please solve it on "***PRACTICE***" first, before moving on to the solution.

A **simple solution** is to traverse each element and check if there's another number in the array which can be added to it to give sum.

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

// Returns number of pairs in arr[0..n-1]
// with sum equal to 'sum'
int printPairs(int arr[], int n, int sum)
{
    int count = 0; // Initialize result

    // Consider all possible pairs and check
    // their sums
    for (int i = 0; i < n; i++)
        for (int j = i + 1; j < n; j++)
            if (arr[i] + arr[j] == sum)
                cout << "(" << arr[i] << ", "
                     << arr[j] << ")" << endl;
}

// Driver function to test the above function
int main()
{
    int arr[] = { 1, 5, 7, -1, 5 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int sum = 6;
    printPairs(arr, n, sum);
    return 0;
}
```

```
// print pairs with given sum.

class GFG {

    // Returns number of pairs
    // in arr[0..n-1] with sum
    // equal to 'sum'
    static void printPairs(int arr[],
                           int n, int sum)
    {
        // int count = 0;

        // Consider all possible pairs
        // and check their sums
        for (int i = 0; i < n; i++)
            for (int j = i + 1; j < n; j++)
                if (arr[i] + arr[j] == sum)
                    System.out.println("(" + arr[i] + ", " + arr[j] + ")");
    }

    // Driver Code
    public static void main(String[] arg)
    {
        int arr[] = { 1, 5, 7, -1, 5 };
        int n = arr.length;
        int sum = 6;
        printPairs(arr, n, sum);
    }
}
```

```python
# Python 3 implementation
# of simple method to find
# print pairs with given sum.

# Returns number of pairs
# in arr[0..n-1] with sum
# equal to 'sum'
def printPairs(arr, n, sum):

    # count = 0

    # Consider all possible
    # pairs and check their sums
    for i in range(0, n ):
        for j in range(i + 1, n ):
            if (arr[i] + arr[j] == sum):
                print("(", arr[i],
                      ", ", arr[j],
                      ")", sep = "")


# Driver Code
arr = [1, 5, 7, -1, 5]
n = len(arr)
sum = 6
printPairs(arr, n, sum)

# This code is contributed
```

```
// C# implementation of simple
// method to find print pairs
// with given sum.
using System;

class GFG {
    // Returns number of pairs
    // in arr[0..n-1] with sum
    // equal to 'sum'
    static void printPairs(int[] arr,
                           int n, int sum)
    {
        // int count = 0;

        // Consider all possible pairs
        // and check their sums
        for (int i = 0; i < n; i++)
            for (int j = i + 1; j < n; j++)
                if (arr[i] + arr[j] == sum)
                    Console.Write("(" + arr[i] + ", " + arr[j] + ")"
                                  + "\n");
    }

    // Driver Code
    public static void Main()
    {
        int[] arr = { 1, 5, 7, -1, 5 };
        int n = arr.Length;
        int sum = 6;
```

PHP

```php
<?php
// PHP implementation of simple
// method to find print pairs
// with given sum.

// Returns number of pairs in
// arr[0..n-1] with sum equal
// to 'sum'
function printPairs($arr, $n, $sum)
{
    // Initialize result
    $count = 0;

    // Consider all possible
    // pairs and check their sums
    for ($i = 0; $i < $n; $i++)
        for ( $j = $i + 1; $j < $n; $j++)
            if ($arr[$i] + $arr[$j] == $sum)
                echo "(", $arr[$i], ", ",
                           $arr[$j], ")", "\n";
}

// Driver Code
$arr = array (1, 5, 7, -1, 5);
```

Javascript

```
<script>

// JavaScript implementation of simple method to
// find print pairs with given sum.

// Returns number of pairs in arr[0..n-1]
// with sum equal to 'sum'
function printPairs(arr, n, sum)
{
    let count = 0; // Initialize result

    // Consider all possible pairs and check
    // their sums
    for (let i = 0; i < n; i++)
        for (let j = i + 1; j < n; j++)
            if (arr[i] + arr[j] == sum)
                 document.write("(" + arr[i] + ", "
                    + arr[j] + ")" + "<br>");
}

// Driver function to test the above function

    let arr = [ 1, 5, 7, -1, 5 ];
    let n = arr.length;
```

```
</script>
```

Output :

```
(1, 5)
(1, 5)
(7, -1)
```

Method 2 (Use hashing).
We create an empty hash table. Now we traverse through the array and check for pairs in the hash table. If a matching element is found, we print the pair number of times equal to the number of occurrences of the matching element.
Note that the worst case of time complexity of this solution is **O(c + n)** where c is the count of pairs with a given sum.

C++

```
// C++ implementation of simple method to
// find count of pairs with given sum.
#include <bits/stdc++.h>
using namespace std;
```

```
    // Traverse through all elements
    for (int i = 0; i < n; i++) {

        // Search if a pair can be formed with
        // arr[i].
        int rem = sum - arr[i];
        if (m.find(rem) != m.end()) {
            int count = m[rem];
            for (int j = 0; j < count; j++)
                cout << "(" << rem << ", "
                     << arr[i] << ")" << endl;
        }
        m[arr[i]]++;
    }
}

// Driver function to test the above function
int main()
{
    int arr[] = { 1, 5, 7, -1, 5 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int sum = 6;
    printPairs(arr, n, sum);
    return 0;
}
```

```
import java.util.*;

class GFG{

// Returns number of pairs in arr[0..n-1]
// with sum equal to 'sum'
static void printPairs(int arr[], int n,
                       int sum)
{

    // Store counts of all elements in map m
    HashMap<Integer,
            Integer> mp = new HashMap<Integer,
                                      Integer>();

    // Traverse through all elements
    for(int i = 0; i < n; i++)
    {

        // Search if a pair can be formed with
        // arr[i].
        int rem = sum - arr[i];
        if (mp.containsKey(rem))
        {
            int count = mp.get(rem);

            for(int j = 0; j < count; j++)
                System.out.print("(" + rem +
                                 ", " + arr[i] +
                                  ")" + "\n");
```

```
        {
            mp.put(arr[i], 1);
        }
    }
}

// Driver code
public static void main(String[] args)
{
    int arr[] = { 1, 5, 7, -1, 5 };
    int n = arr.length;
    int sum = 6;

    printPairs(arr, n, sum);
}
}

// This code is contributed by Princi Singh
```

Python3

```
# Python3 implementation of simple method to
# find count of pairs with given sum.

# Returns number of pairs in arr[0..n-1]
# with sum equal to 'sum'
def printPairs(arr, n, sum):
```

```
    for i in range(n):

        # Search if a pair can be
        # formed with arr[i]
        temp = sum - arr[i]

        if temp in mydict:
            count = mydict[temp]
            for j in range(count):
                print("(", temp, ", ", arr[i],
                      ")", sep = "", end = '\n')

        if arr[i] in mydict:
            mydict[arr[i]] += 1
        else:
            mydict[arr[i]] = 1

# Driver code
if __name__ == '__main__':

    arr = [ 1, 5, 7, -1, 5 ]
    n = len(arr)
    sum = 6

    printPairs(arr, n, sum)

# This code is contributed by MuskanKalra1
```

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

class GFG{

// Returns number of pairs in arr[0..n-1]
// with sum equal to 'sum'
static void printPairs(int []arr, int n, int sum)
{

    // Store counts of all elements in map m
    Dictionary<int,
               int> m = new Dictionary<int,
                                       int>();

    // Traverse through all elements
    for(int i = 0; i < n; i++)
    {

        // Search if a pair can be formed with
        // arr[i].
        int rem = sum - arr[i];

        if (m.ContainsKey(rem))
        {
            int count = m[rem];

            for(int j = 0; j < count; j++)
            {
```

```
        {
            m[arr[i]]++;
        }
        else
        {
            m[arr[i]] = 1;
        }
    }
}

// Driver code
public static void Main(string[] args)
{
    int []arr = { 1, 5, 7, -1, 5 };
    int n = arr.Length;
    int sum = 6;

    printPairs(arr, n, sum);
}
}

// This code is contributed by rutvik_56
```

Javascript

```
<script>
      // JavaScript implementation of simple method to
```

```
  // Traverse through all elements
  for (var i = 0; i < n; i++) {
    // Search if a pair can be formed with
    // arr[i].
    var rem = sum - arr[i];

    if (m.hasOwnProperty(rem)) {
      var count = m[rem];

      for (var j = 0; j < count; j++) {
        document.write("(" + rem + ", " + arr[i] + ")" + "<br>");
      }
    }

    if (m.hasOwnProperty(arr[i])) {
      m[arr[i]]++;
    } else {
      m[arr[i]] = 1;
    }
  }
}

// Driver code
var arr = [1, 5, 7, -1, 5];
var n = arr.length;
var sum = 6;

printPairs(arr, n, sum);
```

```
(1, 5)
(7, -1)
(1, 5)
```

Method 3.

Another method to Print all pairs with the given sum is given as follows:

C++

```
// C++ code to implement
// the above approach
#include<bits/stdc++.h>
using namespace std;

void pairedElements(int arr[],
                    int sum, int n)
{
  int low = 0;
  int high = n - 1;

  while (low < high)
  {
    if (arr[low] + arr[high] == sum)
    {
```

```
      high--;
    }
    else
    {
      low++;
    }
  }
}

// Driver code
int  main()
{
  int arr[] = {2, 3, 4, -2,
               6, 8, 9, 11};
  int n = sizeof(arr) / sizeof(arr[0]);
  sort(arr, arr + n);
  pairedElements(arr, 6, n);
}

// This code is contributed by Rajput-Ji
```

Java

```
import java.util.Arrays;

/**
 * Created by sampat.
```

```
        int high = arr.length - 1;

        while (low < high) {
            if (arr[low] + arr[high] == sum) {
                System.out.println("The pair is : ("
                                   + arr[low] + ", " + arr[high] + ")");
            }
            if (arr[low] + arr[high] > sum) {
                high--;
            }
            else {
                low++;
            }
        }
    }

    public static void main(String[] args)
    {
        int arr[] = { 2, 3, 4, -2, 6, 8, 9, 11 };
        Arrays.sort(arr);

        SumOfPairs sp = new SumOfPairs();
        sp.pairedElements(arr, 6);
    }
}
```

Python3

```
    high = len(arr) - 1;

    while (low < high):
        if (arr[low] +
            arr[high] == sum):
            print("The pair is : (", arr[low],
                  ", ", arr[high], ")");
        if (arr[low] + arr[high] > sum):
            high -= 1;
        else:
            low += 1;

# Driver code
if __name__ == '__main__':

    arr = [2, 3, 4, -2,
           6, 8, 9, 11];
    arr.sort();
    pairedElements(arr, 6);

# This code contributed by shikhasingrajput
```

C#

```
// C# program to find triplets in a given
// array whose sum is equal to given sum.
using System;
```

```
    int high = arr.Length - 1;

    while (low < high)
    {
        if (arr[low] + arr[high] == sum)
        {
            Console.WriteLine("The pair is : ("
                              + arr[low] + ", " + arr[high] + ")");
        }
        if (arr[low] + arr[high] > sum)
        {
            high--;
        }
        else
        {
            low++;
        }
    }
}

// Driver code
public static void Main(String[] args)
{
    int []arr = { 2, 3, 4, -2, 6, 8, 9, 11 };
    Array.Sort(arr);

    SumOfPairs sp = new SumOfPairs();
    sp.pairedElements(arr, 6);
}
```

Javascript

```
<script>

        // Javascript code to implement
        // the above approach
        function pairedElements(arr, sum, n) {
            var low = 0;
            var high = n - 1;

            while (low < high) {
                if (arr[low] + arr[high] == sum) {
                    document.write("The pair is : (" +
                        arr[low] + ", " +
                        arr[high] + ")<br>");
                }
                if (arr[low] + arr[high] > sum) {
                    high--;
                }
                else {
                    low++;
                }
            }
        }

        // Driver code
        var arr = [ 2, 3, 4, -2, 6, 8, 9, 11]
        var n = arr.length;
        arr.sort(function(a,b){return a-b;});
```

Output :

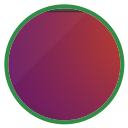
```
The pair is : (-2, 8)
The pair is : (2, 4)
```


Previous Next

RECOMMENDED ARTICLES

Page :

Article Contributed By :



kartik
@kartik

Vote for difficulty

Current difficulty : Easy

Easy | Normal | Medium | Hard | Expert

Improved By : jit_t, Smitha Dinesh Semwal, SampatKumar1, princi singh, rutvik_56, Rajput-Ji,

GeeksforGeeks
5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org