



[Data Structures](#) [Algorithms](#) [Interview Preparation](#) [Topic-wise Practice](#) [C++](#) [Java](#) [Python](#) [Competitive Programming](#) [Machi](#)

Print sums of all subsets of a given set

Difficulty Level : Medium • Last Updated : 24 Jan, 2022

Given an array of integers, print sums of all subsets in it. Output sums can be printed in any order.

Examples :

Input : arr[] = {2, 3}

Output: 0 2 3 5

Input : arr[] = {2, 4, 5}

Output : 0 2 4 5 6 7 9 11

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



Method 1 (Recursive)

We can recursively solve this problem. There are total 2^n subsets. For every element, we consider two

Start Your Coding Journey Now!

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

```
// C++ program to print sums of all possible
// subsets.
#include <bits/stdc++.h>
using namespace std;

// Prints sums of all subsets of arr[l..r]
void subsetSums(int arr[], int l, int r, int sum = 0)
{
    // Print current subset
    if (l > r) {
        cout << sum << " ";
        return;
    }

    // Subset including arr[l]
    subsetSums(arr, l + 1, r, sum + arr[l]);

    // Subset excluding arr[l]
    subsetSums(arr, l + 1, r, sum);
}

// Driver code
int main()
{
    int arr[] = { 5, 4, 3 };
    int n = sizeof(arr) / sizeof(arr[0]);

    subsetSums(arr, 0, n - 1);
}
```



Start Your Coding Journey Now!

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

```
// Java program to print sums
// of all possible subsets.
import java.io.*;

class GFG {

    // Prints sums of all
    // subsets of arr[l..r]
    static void subsetSums(int[] arr, int l, int r, int sum)
    {

        // Print current subset
        if (l > r) {
            System.out.print(sum + " ");
            return;
        }

        // Subset including arr[l]
        subsetSums(arr, l + 1, r, sum + arr[l]);

        // Subset excluding arr[l]
        subsetSums(arr, l + 1, r, sum);
    }

    // Driver code
    public static void main(String[] args)
    {
        int[] arr = { 5, 4, 3 };
    }
}
```



Start Your Coding Journey Now!

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

// This code is contributed by anuj_67

Python3

```
# Python3 program to print sums of
# all possible subsets.

# Prints sums of all subsets of arr[l..r]

def subsetSums(arr, l, r, sum=0):

    # Print current subset
    if l > r:
        print(sum, end=" ")
        return

    # Subset including arr[l]
    subsetSums(arr, l + 1, r, sum + arr[l])

    # Subset excluding arr[l]
    subsetSums(arr, l + 1, r, sum)

# Driver code
arr = [5, 4, 3]
n = len(arr)
```

Start Your Coding Journey Now!

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

```
// C# program to print sums of all possible
// subsets.
using System;

class GFG {

    // Prints sums of all subsets of
    // arr[l..r]
    static void subsetSums(int[] arr, int l, int r, int sum)
    {

        // Print current subset
        if (l > r) {
            Console.Write(sum + " ");
            return;
        }

        // Subset including arr[l]
        subsetSums(arr, l + 1, r, sum + arr[l]);

        // Subset excluding arr[l]
        subsetSums(arr, l + 1, r, sum);
    }

    // Driver code
    public static void Main()
    {
```

Start Your Coding Journey Now!

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

// This code is contributed by anuj_67

PHP

```
<?php
// PHP program to print sums
// of all possible subsets.

// Prints sums of all
// subsets of arr[l..r]
function subsetSums($arr, $l,
                    $r, $sum = 0)
{
    // Print current subset
    if ($l > $r)
    {
        echo $sum , " ";
        return;
    }

    // Subset including arr[l]
    subsetSums($arr, $l + 1, $r,
                $sum + $arr[$l]);

    // Subset excluding arr[l]
    subsetSums($arr, $l + 1, $r, $sum);
}
```

Start Your Coding Journey Now!

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

```
subsetSums($arr, 0, $n - 1);
```

```
// This code is contributed by anuj_67.
```

```
?>
```

Javascript

```
<script>
```

```
// Javascript program to program to print
```

```
// sums of all possible subsets.
```

```
// Prints sums of all
```

```
// subsets of arr[l..r]
```

```
function subsetSums(arr, l, r, sum)
```

```
{
```

```
    // Print current subset
```

```
    if (l > r)
```

```
    {
```

```
        document.write(sum + " ");
```

```
        return;
```

```
    }
```

```
    // Subset including arr[l]
```

```
    subsetSums(arr, l + 1, r,
```

```
        sum + arr[l]);
```



Start Your Coding Journey Now!

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

```
let n = arr.length;

subsetSums(arr, 0, n - 1, 0);

// This code is contributed by code_hunt

</script>
```

Output :

12 9 8 5 7 4 3 0

Time complexity of this solution is $O(2^n)$ and space complexity is $O(2^n)$.

Method 2 (Iterative)

As discussed above, there are total 2^n subsets. The idea is generate loop from 0 to $2^n - 1$. For every number, pick all array elements which correspond to 1s in binary representation of current number.

C++

```
// Iterative C++ program to print sums of all
// possible subsets.
#include <bits/stdc++.h>
using namespace std;
```



Start Your Coding Journey Now!

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

```
// Consider all numbers from 0 to 2^n - 1
for (long long i = 0; i < total; i++) {
    long long sum = 0;

    // Consider binary representation of
    // current i to decide which elements
    // to pick.
    for (int j = 0; j < n; j++)
        if (i & (1 << j))
            sum += arr[j];

    // Print sum of picked elements.
    cout << sum << " ";
}

// Driver code
int main()
{
    int arr[] = { 5, 4, 3 };
    int n = sizeof(arr) / sizeof(arr[0]);

    subsetSums(arr, n);
    return 0;
}
```



Java

Start Your Coding Journey Now!

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

```
// Prints sums of all subsets of array
static void subsetSums(int arr[], int n)
{

    // There are total 2^n subsets
    int total = 1 << n;

    // Consider all numbers from 0 to 2^n - 1
    for (int i = 0; i < total; i++) {
        int sum = 0;

        // Consider binary representation of
        // current i to decide which elements
        // to pick.
        for (int j = 0; j < n; j++)
            if ((i & (1 << j)) != 0)
                sum += arr[j];

        // Print sum of picked elements.
        System.out.print(sum + " ");
    }
}

// Driver code
public static void main(String args[])
{
    int arr[] = new int[] { 5, 4, 3 };
    int n = arr.length;
```



Start Your Coding Journey Now!

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

Python3

Iterative Python3 program to print sums of all possible subsets

Prints sums of all subsets of array

```
def subsetSums(arr, n):
```

```
    # There are total 2^n subsets
```

```
    total = 1 << n
```

```
    # Consider all numbers from 0 to 2^n - 1
```

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

```
        Sum = 0
```

```
        # Consider binary representation of
```

```
        # current i to decide which elements
```

```
        # to pick.
```

```
        for j in range(n):
```

```
            if ((i & (1 << j)) != 0):
```

```
                Sum += arr[j]
```

```
        # Print sum of picked elements.
```

```
        print(Sum, "", end = "")
```

```
arr = [ 5, 4, 3 ]
```

```
n = len(arr)
```



Start Your Coding Journey Now!

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

```
// Iterative C# program to print sums of all
// possible subsets.
using System;
class GFG {

    // Prints sums of all subsets of array
    static void subsetSums(int[] arr, int n)
    {

        // There are total 2^n subsets
        int total = 1 << n;

        // Consider all numbers from 0 to 2^n - 1
        for (int i = 0; i < total; i++) {
            int sum = 0;

            // Consider binary representation of
            // current i to decide which elements
            // to pick.
            for (int j = 0; j < n; j++)
                if ((i & (1 << j)) != 0)
                    sum += arr[j];

            // Print sum of picked elements.
            Console.Write(sum + " ");
        }
    }
}
```

Start Your Coding Journey Now!

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

```
}  
}
```

// This code is contributed by divyesh072019.

PHP

```
<?php  
// Iterative PHP program to print  
// sums of all possible subsets.  
  
// Prints sums of all subsets of array  
function subsetSums($arr, $n)  
{  
  
    // There are total 2^n subsets  
    $total = 1 << $n;  
  
    // Consider all numbers  
    // from 0 to 2^n - 1  
    for ($i = 0; $i < $total; $i++)  
    {  
        $sum = 0;  
  
        // Consider binary representation of  
        // current i to decide which elements  
        // to pick.
```

Start Your Coding Journey Now!

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

```
}  
}  
  
// Driver code  
$arr = array(5, 4, 3);  
$n = sizeof($arr);  
subsetSums($arr, $n);  
  
// This Code is Contributed by ajit  
?>
```

Javascript

```
<script>  
  
// Iterative Javascript program to print sums of all  
// possible subsets.  
  
// Prints sums of all subsets of array  
function subsetSums(arr, n)  
{  
  
    // There are total 2^n subsets  
    let total = 1 << n;  
  
    // Consider all numbers from 0 to 2^n - 1  
    for(let i = 0; i < total; i++)
```

Start Your Coding Journey Now!

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

```
// ...  
for(let j = 0; j < n; j++)  
    if ((i & (1 << j)) != 0)  
        sum += arr[j];  
  
    // Print sum of picked elements.  
    document.write(sum + " ");  
}  
  
let arr = [ 5, 4, 3 ];  
let n = arr.length;  
  
subsetSums(arr, n);  
  
</script>
```

Output :

0 5 4 9 3 8 7 12

Time Complexity: $O(N * 2^N)$

Auxiliary Space: $O(1)$



Thanks to cfh for suggesting above iterative solution in a comment.

Note: We haven't actually created sub-sets to find their sums rather we have just used recursion to find sum of non-contiguous sub-sets of the given set.

Start Your Coding Journey Now!

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

can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

[Like](#)[Previous](#)[Next](#)

Recursive program to print all subsets with **Program to reverse a string (Iterative and**

Start Your Coding Journey Now!

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

RECOMMENDED ARTICLES

Page :

Article Contributed By :



GeeksforGeeks

Vote for difficulty

Current difficulty : [Medium](#)

[Easy](#)[Normal](#)[Medium](#)[Hard](#)[Expert](#)

Improved By : [vt_m](#), [jit_t](#), [spp_____](#), [code_hunt](#), [divyeshrabadiya07](#), [atulp99945](#), [rgarg2580](#),
[varshagumber28](#), [pankajsharmagfg](#), [divyesh072019](#), [mukesh07](#), [avtarkumar719](#), [as5853535](#)

Start Your Coding Journey Now!

[Login](#)[Register](#)[Report Issue](#)

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

[Load Comments](#)

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

feedback@geeksforgeeks.org

Company

[About Us](#)[Careers](#)[In Media](#)

Learn

[Algorithms](#)[Data Structures](#)[SDE Cheat Sheet](#)

Start Your Coding Journey Now!

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

News

- Top News
- Technology
- Work & Career
- Business
- Finance
- Lifestyle

Languages

- Python
- Java
- CPP
- Golang
- C#
- SQL

Web Development

- Web Tutorials
- Django Tutorial
- HTML
- CSS
- JavaScript
- Bootstrap

Contribute

- Write an Article
- Improve an Article
- Pick Topics to Write
- Write Interview Experience
- Internships
- Video Internship

@geeksforgeeks , Some rights reserved

