Print all combinations of points that can compose a given number

You can win three kinds of basketball points, 1 point, 2 points, and 3 points. Given a total score n, print out all the combination to compose n.

Examples:
For n = 1, the program should print following:
1

For n = 2, the program should print following:
1 1
2

For n = 3, the program should print following:
1 1 1
1 2
2 1
3

For n = 4, the program should print following:
1 1 1 1
1 1 2
1 2 1
1 3
2 1 1
2 2
3 1

Google™ Custom Search

GeeksforGeeks

53,525 people like GeeksforGeeks.

Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

and so on …

Algorithm:
At first position we can have three numbers 1 or 2 or 3.
First put 1 at first position and recursively call for n-1.
Then put 2 at first position and recursively call for n-2.
Then put 3 at first position and recursively call for n-3.
If n becomes 0 then we have formed a combination that compose n, so print the current combination.

Below is a generalized implementation. In the below implementation, we can change MAX_POINT if there are higher points (more than 3) in the basketball game.

```
#define MAX_POINT 3
#define ARR_SIZE 100
#include<stdio.h>

/* Utility function to print array arr[] */
void printArray(int arr[], int arr_size);

/* The function prints all combinations of numbers 1, 2, ...MAX_POINT
   that sum up to n.
   i is used in recursion keep track of index in arr[] where next
   element is to be added. Initital value of i must be passed as 0 */
void printCompositions(int n, int i)
{

  /* array must be static as we want to keep track
   of values stored in arr[] using current calls of
   printCompositions() in function call stack*/
  static int arr[ARR_SIZE];

  if (n == 0)
  {
    printArray(arr, i);
  }
  else if(n > 0)
  {
    int k;
    for (k = 1; k <= MAX_POINT; k++)
    {
      arr[i]= k;
      printCompositions(n-k, i+1);
```

Popular Posts

All permutations of a given string
Memory Layout of C Programs
Understanding “extern” keyword in C
Median of two sorted arrays
Tree traversal without recursion and without stack!
Structure Member Alignment, Padding and Data Packing
Intersection point of two Linked Lists
Lowest Common Ancestor in a BST.
Check if a binary tree is BST or not
Sorted Linked List to Balanced BST

```
    }
  }
}

/* UTILITY FUNCTIONS */
/* Utility function to print array arr[] */
void printArray(int arr[], int arr_size)
{
  int i;
  for (i = 0; i < arr_size; i++)
    printf("%d ", arr[i]);
  printf("\n");
}

/* Driver function to test above functions */
int main()
{
  int n = 5;
  printf("Differnt compositions formed by 1, 2 and 3 of %d are\n", n);
  printCompositions(n, 0);
  getchar();
  return 0;
}
```

Asked by Aloe

Please write comments if you find any bug in above code/algorithm, or find other ways to solve the same problem.

705 Subscribe

Related Tpoics:

- Backtracking | Set 8 (Solving Cryptarithmetic Puzzles)
- Tail Recursion
- Find if two rectangles overlap
- Analysis of Algorithm | Set 4 (Solving Recurrences)
- Print all possible paths from top left to bottom right of a mXn matrix
- Generate all unique partitions of an integer
- Russian Peasant Multiplication
- Closest Pair of Points | O(nlogn) Implementation

0 Tweet 0 0

Writing code in comment? Please use **ideone.com** and share the link here.

17 Comments GeeksforGeeks

Sort by Newest ▾

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 17 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 57 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 1 hour ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

Join the discussion…

ashish • 2 months ago

Can someone please explain how recursion is working here or give a dry run c

I am not able to understand when k=2, printCompositions(-1,2) then how it is p

puneetbestpat • 9 months ago

I was wondering if this method can be applied if the MAX_POINT is 10^9. woul

```
/* Paste your code here (You may delete these lines if not writing co
```

puneetbestpat • 9 months ago

I was wondering if this method can be applied if the MAX_POINT is 10^9. woul

abhishek08aug • a year ago

Intelligent :D

satya • 3 years ago

@all Can You Explain The Time Complexity of The Code.??
is it n*2^n or simply 2^n can some one prove it

Reply ASAP.

prakhar ➦ satya • 2 years ago

May be a loose upper bound is O(MAX_POINT^n).

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

Can someone explain a tighter upper bound ?

WgpShashank • 3 years ago

```
 class PrintAllCombinations
{


       public static void main(String[] args)
       {
               int s = 9;
               //int[] a = new int[] {2,3,6,7,8};
               int[] a = new int[] {2,3,6,7,8,0,10,66,45,3,56,89};
               getCombinatsions(a, 0, s, 0, "");
       }

       static void getCombinatsions(int[] a, int j, int desiredSum,
       {
               if(desiredSum == currentSum) {
                       System.out.println(st);
                       return;
               }
```

see more

WgpShashank • 3 years ago

Hi Geeks, I Just Modified The Above Program to Printb Unique Combination th
know if anything wrong in this code

```
  #define MAX_POINT 4
#define ARR_SIZE 100
#include <stdio.h>
```

```
/* Utility function to print array arr[] */
void printArray(int arr[], int arr_size);

/* The function prints all combinations of numbers 1, 2, ...MAX_POINT
   that sum up to n.
   i is used in recursion keep track of index in arr[] where next
   element is to be added. Initital value of i must be passed as 0 */
void printCompositions(int arr[ARR_SIZE],int n, int i)
{

  /* array must be static as we want to keep track
```

see more

coder WgpShashank • 2 years ago

for (i = 0; i arr[i+1]) flag=0;

i guess u wrote in hurry..please update it

WgpShashank • 3 years ago

for Explanation See here

http://1.bp.blogspot.com/-CnDx...

Shashank Mani Narayan • 3 years ago

@geeksforGeeks this will print duplicate values as well we have to modify it..

like for 3

it will print
12
21
but we wants only one (unique) way ??

raman Shashank Mani Narayan • 3 years ago

what modification we have to do in given program so that it can print th
number....ltes wait for GeeksforGeeks How they approach for this ques

Aravind_Sen • 4 years ago

no. of calls to functions can be optimized
and the size of array should be MAX_POINT

```
 void printCompositions(int n, int i)
{

  /* array must be static as we want to keep track
   of values stored in arr[] using current calls of
   printCompositions() in function call stack*/

    static int arr<strong>[MAX_POINT];
    int k;
    for (k = 1; k <= MAX_POINT; k++)
    {
      arr[i]= k;
      if(n-k >0 )
         printCompositions(n-k, i+1);
      else if(n-k==0)
         printArray(arr, i+1);
}
```

atul ➦ Aravind_Sen • 2 years ago

ok sorry this will work, i skipped i+1 you were passing to printArray.

atul ➦ Aravind_Sen • 2 years ago

your optmized code wont works:-

this will work;

```
for(k=1;k<=max;k++)
               {
                       if(num > 0)
                       {
                               arr[arrlen]=k;

                               if(num-k > 0)
                               {
                                 combination(num-k,max,arrlen-
                               }
                               else if(num-k == 0)
                               {
                                        for(p=0;p<arrlen+1;p+-
                                        {
```

see more

Yes, true. We could check for the value of (n-k) to see if it is greater tha
Will reduce the recursion overhead.

sharat04 ➦ reg_frenzy • 3 years ago

agreed. I had the same concern

Subscribe Add Disqus to your site