



PRACTICE

CERTIFICATION

COMPETE

JOBS

LEADERBOARD

Search



shashank_sank...

Practice > Python > Basic Data Types > List Comprehensions

List Comprehensions ★

41/115 challenges solved

Rank: 57858 | Points: 565



Problem

Submissions

Leaderboard

Discussions

Editorial

Tutorial

Let's learn about list comprehensions! You are given three integers x , y and z representing the dimensions of a cuboid along with an integer n . Print a list of all possible coordinates given by (i, j, k) on a 3D grid where the sum of $i + j + k$ is not equal to n . Here, $0 \leq i \leq x$; $0 \leq j \leq y$; $0 \leq k \leq z$. Please use list comprehensions rather than multiple loops, as a learning exercise.

Example

 $x = 1$ $y = 1$ $z = 2$ $n = 3$ All permutations of $[i, j, k]$ are: $[[0, 0, 0], [0, 0, 1], [0, 0, 2], [0, 1, 0], [0, 1, 1], [0, 1, 2], [1, 0, 0], [1, 0, 1], [1, 0, 2], [1, 1, 0], [1, 1, 1], [1, 1, 2]]$.Print an array of the elements that do not sum to $n = 3$. $[[0, 0, 0], [0, 0, 1], [0, 0, 2], [0, 1, 0], [0, 1, 1], [1, 0, 0], [1, 0, 1], [1, 1, 0], [1, 1, 2]]$

Input Format

Author	harsh_beria93
Difficulty	Easy
Max Score	10
Submitted By	351320

NEED HELP?

[View tutorial](#)[View discussions](#)[View editorial](#)[View top submissions](#)

RATE THIS CHALLENGE



MORE DETAILS

[Download problem statement](#)

Change Theme

Python 3



```
1
2     x = int(input())
3     y = int(input())
4     z = int(input())
5     n = int(input())
6     l=[]
7     r=[]
8     for i in range(x+1):
9         for j in range(y+1):
10            for k in range(z+1):
11                s=[i,j,k]
12                l.append(s)
13     for i in range(len(l)):
14         if sum(l[i])!=n:
15             r.append(l[i])
16
17 print(r)
18
```

Line: 18 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code

```
Line: 18 Col: 1
```

[Upload Code as File](#) Test against custom input[Run Code](#)[Submit Code](#)[Test case 0](#)

Compiler Message

[Test case 1](#)

Success

[Test case 2](#)

Input (stdin)

[Download](#)

```
1 1
2 1
3 1
4 2
```

[Test case 3](#) [Test case 4](#) [Test case 5](#) [Test case 6](#)

Expected Output

[Download](#)

```
1 [[0, 0, 0], [0, 0, 1], [0, 1, 0], [1, 0, 0], [1, 1, 1]]
```