# List Comprehensions

35 more points to get your next star!

Rank: 1417414 | Points: 75/110

Problem Submissions Leaderboard Editorial Tutorial

Let's learn about list comprehensions! You are given three integers and representing the dimensions of a cuboid along with an integer . Print a list of all possible coordinates given by on a 3D grid where the sum of is not equal to . Here, . Please use list comprehensions rather than multiple loops, as a learning exercise.

**Example**

All permutations of are:

.

Print an array of the elements that do not sum to .

**Input Format**

Four integers and , each on a separate line.

**Constraints**

Print the list in lexicographic increasing order.

**Sample Input 0**

```
1
1
1
2
```

**Sample Output 0**

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

**Explanation 0**

Each variable and will have values of or . All permutations of lists in the form .
Remove all arrays that sum to to leave only the valid permutations.

**Sample Input 1**

```
2
2
2
2
```

**Sample Output 1**

```
[[0, 0, 0], [0, 0, 1], [0, 1, 0], [0, 1, 2], [0, 2, 1], [0, 2, 2], [1, 0, 0], [1, 0, 2], [1, 1, 1], [1, 1, 2], [1, 2, 0], [1, 2, 1], [1, 2, 2], [2, 0, 1], [2, 0, 2], [2, 1, 0], [2, 1, 1], [2, 1, 2], [2, 2, 0], [2, 2, 1], [2, 2, 2]]
```

Change Theme Language Python 3

```python
if __name__ == '__main__':
    X = int(input())  # Read X
    Y = int(input())  # Read Y
    Z = int(input())  # Read Z
    N = int(input())  # Read N

    # List comprehension to generate the coordinates
    result = [[x, y, z] for x in range(X + 1) for y in range(Y + 1) for z in range(Z + 1) if x + y + z != N]

    # Print the result list
    print(result)

```

Line: 12 Col: 1

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

## Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Sample Test case 1

Input (stdin) Download

```
1
1
1
2
```

Your Output (stdout)

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

Expected Output Download

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