

第一题：当 a,b,c 为 10, 5, 1 时，输出结果为 5

```
# 报告当 a = 10, b = 5, c = 1 时的输出  
a, b, c = 10, 5, 1  
inputs(a, b, c)
```

The result is: 5

第二题（我从 CSDN 上学习到了 math.ceil 的用法和含义）

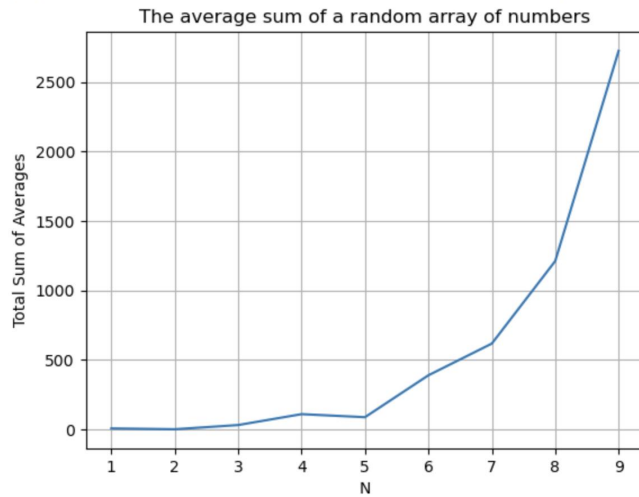
第三题：输出和从 10-60 的所有方法数，产生最多方法数的和为 35，最多的方法数为 4395456
（通过阅读 CSDN 上的关于投骰子的动态规划的文章，确定了用数组储存方法数，进行动态规划的思路）

```
和为 10 的方法数: 1  
和为 11 的方法数: 10  
和为 12 的方法数: 55  
和为 13 的方法数: 220  
和为 14 的方法数: 715  
和为 15 的方法数: 2002  
和为 16 的方法数: 4995  
和为 17 的方法数: 11340  
和为 18 的方法数: 23760  
和为 19 的方法数: 46420  
和为 20 的方法数: 85228  
和为 21 的方法数: 147940  
和为 22 的方法数: 243925  
和为 23 的方法数: 383470  
和为 24 的方法数: 576565  
和为 25 的方法数: 831204  
和为 26 的方法数: 1151370  
和为 27 的方法数: 1535040  
和为 28 的方法数: 1972630  
和为 29 的方法数: 2446300  
和为 30 的方法数: 2930455  
和为 31 的方法数: 3393610  
和为 32 的方法数: 3801535  
和为 33 的方法数: 4121260  
和为 34 的方法数: 4325310  
和为 35 的方法数: 4395456  
和为 36 的方法数: 4325310  
和为 37 的方法数: 4121260  
和为 38 的方法数: 3801535  
和为 39 的方法数: 3393610  
和为 40 的方法数: 2930455  
和为 41 的方法数: 2446300  
和为 42 的方法数: 1972630  
和为 43 的方法数: 1535040  
和为 44 的方法数: 1151370  
和为 45 的方法数: 831204  
和为 46 的方法数: 576565  
和为 47 的方法数: 383470  
和为 48 的方法数: 243925  
和为 49 的方法数: 147940  
和为 50 的方法数: 85228
```

```
和为 51 的方法数: 46420  
和为 52 的方法数: 23760  
和为 53 的方法数: 11340  
和为 54 的方法数: 4995  
和为 55 的方法数: 2002  
和为 56 的方法数: 715  
和为 57 的方法数: 220  
和为 58 的方法数: 55  
和为 59 的方法数: 10  
和为 60 的方法数: 1  
产生最多方法的和是: 35  
最多的方法数是: 4395456
```

第四题：由于 N(1, 101) 数值太大，运行时间长，因此为检验代码的正确性，修改一下 N 的范围，输出当 N(1, 10) 时的计算结果和图像（上网查询了如何生成子集的代码，查询如何运用 plt 生成图像）

[6.0, 0.0, 30.333333333333332, 108.74999999999999, 86.80000000000001, 388.5, 616.8571428571431, 1211.2499999999995, 2725.333333333331]



第五题：运行 1000 次时，平均路径数为 0.432（阅读文章《python 如何初始化 m 行 n 列的矩阵》，从中学习作业第一问的矩阵的建立，阅读文章《python 实现从二维矩阵左上角到右下角的出路数寻找》，从中学习第二问的路径个数的寻找）

```
print(f"Average number of paths {runs}: {average}")
```

Average number of paths 1000: 0.432