

小东所在公司要发年终奖，而小东恰好获得了最高福利，他要在公司年会上参与一个抽奖游戏，游戏在一个 **6\*6** 的棋盘上进行，上面放着 **36** 个价值不等的礼物，每个小的棋盘上面放置着一个礼物，他需要从左上角开始游戏，每次只能向下或者向右移动一步，到达右下角停止，一路上的格子里的礼物小东都能拿到，请设计一个算法使小东拿到价值最高的礼物。

给定一个 **6\*6** 的矩阵 **board**，其中每个元素为对应格子的礼物价值,左上角为**[0,0]**,请返回能获得的最大价值，保证每个礼物价值大于 **100** 小于 **1000**。

```
1  import java.util.*;
2
3  public class Bonus {
4      public int getMost(int[][] board) {
5          // write code here
6          int n = board.length;
7          int[][] dp = new int[n][n];
8          dp[0][0] = board[0][0];
9          for (int i = 1; i < n; i++) {
10             dp[0][i] = dp[0][i-1]+board[0][i];
11             dp[i][0] =dp[i-1][0]+board[i][0];
12
13         }
14         for (int i = 1; i < n; i++) {
15             for (int j = 1; j < n; j++) {
16                 dp[i][j] = Math.max(dp[i - 1][j], dp[i][j - 1]) + board[i][j];
17             }
18         }
19         return dp[n - 1][n - 1];
20     }
21 }
```

小东和三个朋友一起在楼上抛小球，他们站在楼房的不同层，假设小东站的楼层距离地面 **N** 米，球从他手里自由落下，每次落地后反跳回上次下落高度的一半，并以此类推知道全部落到地面不跳，求 **4** 个小球一共经过了多少米？(数字都为整数)

给定四个整数 **A,B,C,D**，请返回所求结果。

测试样例：

100,90,80,70

返回：1020

```
1
2
3 class Balls:
4
5     def calcDistance(self, A, B, C, D):
6
7         # write code here
8
9         sum = A+B+C+D
10
11         return sum*4- sum
12
13
```

就是等比数列求和

果园里有一堆苹果，一共  $n$  头( $n$  大于 1 小于 9)熊来分，第一头为小东，它把苹果均分  $n$  份后，多出了一个，它扔掉了这一个，拿走了自己的一份苹果，接着第二头熊重复这一过程，即先均分  $n$  份，扔掉一个然后拿走一份，以此类推直到最后一头熊都是这样(最后一头熊扔掉后可以拿走 0 个，也算是  $n$  份均分)。问最初这堆苹果最少有多少个。

给定一个整数  $n$ , 表示熊的个数，返回最初的苹果数。保证有解。

测试样例：

2

返回：3

```
1 public class Apple {
2     public static int getInitial(int n) {
3         for(int i=n+1;;i++) {
4             int temp=i;
5             int bear=n;
6             while(bear>0) {
7
8                 if(temp%n==1) {
9                     temp=temp-temp/n-1;
10                    bear--;
11                }else {
12                    break;
13                }
14            }
15            if(bear==0) {
16
```

```
17         return i;
18     }
19 }
20 }
21 public static void main(String[] args) {
22     System.out.println(getInitial(2));
23 }
24 }
```

无法从后往前推，因为最后一个熊拿到的个数是不确定的，所以就使用遍历，从  $n+1$  开始，判断在每一次的加减的过程中，该数是否对  $n$  取余等于 1，如果是则继续减，直到熊的个数等于 0 为止，否则就进入下一个数的判断。