# 2017年校招全国统一模拟笔试(第四场)编程题集合--Python

**摘要：** 本文讲的是2017年校招全国统一模拟笔试(第四场)编程题集合--Python， 注：试题来源于牛客网 1. 牛牛有n张卡片排成一个序列.每张卡片一面是黑色的,另一面是白色的。初始状态的时候有些卡片是黑色朝上,有些卡片是白色朝上。牛牛现在想要把一些卡片翻过来,得到一种交替排列的形

[教程](https://www.aliyun.com/jiaocheng/) [云栖大会](https://yunqi.aliyun.com/) [Mysql](https://yq.aliyun.com/sqlarticle) [备案](https://beian.aliyun.com/) [文档](https://help.aliyun.com/) [域名](https://wanwang.aliyun.com/) [whois查询](https://whois.aliyun.com/) [PHP教程](https://yq.aliyun.com/php) [备份](https://www.aliyun.com/jiaocheng/178488.html) [互联网大学](https://edu.aliyun.com/academy/VirtualAcademy) [云教程](https://wanwang.aliyun.com/info/)

注：试题来源于牛客网 ****1.**** 牛牛有n张卡片排成一个序列.每张卡片一面是黑色的,另一面是白色的。初始状态的时候有些卡片是黑色朝上,有些卡片是白色朝上。牛牛现在想要把一些卡片翻过来,得到一种交替排列的形式,即每对相邻卡片的颜色都是不一样的。牛牛想知道最少需要翻转多少张卡片可以变成交替排列的形式。    
****输入描述:****

输入包括一个字符串S,字符串长度length(3 ≤ length ≤ 50),其中只包含'W'和'B'两种字符串,分别表示白色和黑色。整个字符串表示卡片序列的初始状态。

****输出描述:****

输出一个整数,表示牛牛最多需要翻转的次数。

****输入例子1:****

BBBW

****输出例子1:****

1

****个人思路：**** 有两个思路 1.根据第一位，把正确排列弄出来。然后与输入的字符串遍历比较，得出结果   
2.设一个 flag = BW and flag = WB,依次用切片取两个字符，比较，不相同的话，count += 1。最后取两个中的最小值 ****正确答案（本人&&牛客网id：liberty\_zheng）：****

python">flag1, flag2, flag3, flag4, flag5, flag6 = 'BW', 'WB', 'BB', 'WW', 'B', 'W' def bw\_start(arr): i, j = 0, 2 bw = 0 while j <= len(arr)+2: temp = ''.join(arr[i:j]) if temp == flag1: i += 2 j += 2 continue elif temp == flag2: bw += 2 elif temp == flag3 or temp == flag4: bw += 1 elif temp == flag6: bw += 1 i += 2 j += 2 return bw def wb\_start(arr): i, j = 0, 2 wb = while j < len(arr): temp = ''.join(arr[i:j]) if temp == flag2: i += 2 j += 2 continue elif temp == flag1: wb += 2 elif temp == flag3 or temp == flag4: wb += 1 elif temp == flag5: wb += 1 i += 2 j += 2 return wb string = raw\_input() arr = list(string) result = (bw\_start(arr), wb\_start(arr)) print min(result)

我用了第二种思路，就是 flag 立的有点多啦

data=raw\_input() counts1=0 counts2=0 for i in range(len(data)): if i%2==0: if data[i]=='B': continue else: counts1+=1 else: if data[i]=='W': continue else: counts1+=1 print min(counts1,len(data)-counts1)

网上同学是采用相当于第一种思路   
  
  
****2.**** 牛牛变得黑化了,想要摧毁掉地球。但他忘记了开启地球毁灭器的密码。牛牛手里有一个字符串S,牛牛还记得从S中去掉一个字符就恰好是正确的密码,请你帮牛牛求出他最多需要尝试多少次密码。   
如样例所示S = "ABA",3个可能的密码是"BA", "AA", "AB".   
当S = "A", 牛牛唯一可以尝试的密码是一个空的密码,所以输出1.    
****输入描述:****

输入包括一个字符串S,字符串长度length(1 ≤ length ≤ 50),其中都是从'A'到'Z'的大写字母。

****输出描述:****

输出一个整数,表示牛牛最多需要尝试的密码次数。

****输入例子1:****

ABA

****输出例子1:****

3

****个人思路：**** 我的思路太简单，对题的理解就是，一个字符串去掉一个字符后的所有排列组合。 对于又看到排列组合，我分分钟就用了 Python 强大的库 itertools.permutations(arr, num) 。 但结果，超内存啦 再看同学思路，如果相邻的两个字符串相同，那么去除任意一个，效果都是一样的 没有重复的，次数就是字符串长度。这句才是重点。 ****正确答案（牛客网id：liberty\_zheng）：****

data=raw\_input() count=len(data) for i in range(1,len(data)): if data[i]==data[i-1]: count-=1 print count

只能说我太依赖有些东西了。   
  
  
****3.**** 牛牛以草料为食。牛牛有一天依次遇到n堆被施展了魔法的草料,牛牛只要遇到一堆跟他当前相同大小的草料,它就会把草料吃完,而使自己的大小膨胀一倍。一开始牛牛的大小的是A,然后给出牛牛依次遇到的n堆草料的大小。请[计算](https://www.aliyun.com/)牛牛最后的大小。    
****输入描述:****

输入包括两行,第一行包含两个整数n和A(1 ≤ n ≤ 200, 1 ≤ A ≤ 1,000,000,000) 第二行包括n个整数,表示牛牛依次遇到的草料堆大小a\_i(1 ≤ a\_i ≤ 1,000,000,000)

****输出描述:****

输出一个整数,表示牛牛最后的大小。

****输入例子1:****

5 1 2 1 3 1 2

****输出例子1:****

4

****个人思路：**** 很简单的一题，遇见相同的翻倍即可。 ****正确答案（本人）：****

ab = [int(i) for i in raw\_input().strip().split()] arr = [int(i) for i in raw\_input().strip().split()] A = ab[1] for i in range(ab[0]): if arr[i] == A: A = A \* 2 print A

****4.**** 牛牛有一个长度为n的整数序列s,羊羊要在牛牛的序列中选择不同的两个位置,然后交换这两个位置上的元素。现在需要求出羊羊交换后可以得到的不同的序列个数。(注意被交换的两元素值可能相同)。   
如序列{1, 47},输出1.羊羊必须交换仅有的两个元素,得到序列{47, 1}。羊羊必须交换,不能保留原有的序列。   
{1, 2, 1},输出3.羊羊通过交换可以得到{2, 1, 1},{1, 1, 2},{1, 2, 1}这三个序列。    
****输入描述:****

输入包括两行,第一行为一个整数n(2 ≤ n ≤ 50),即序列的长度。 第二行n个整数,表示序列的每个元素a\_i(1 ≤ a\_i ≤ 50),以空格分割。

****输出描述:****

输出一个整数,表示羊羊可以得到的不同的序列个数

****输入例子1:****

3 1 2 1

****输出例子1:****

3

****个人思路：**** 难得这道题我居然用了数学中的找规律，想也没想就直接放上了找出的规律，40% case

n = int(raw\_input()) arr = [int(i) for i in raw\_input().strip().split()] result = n \* (n - 1) / 2

然后一看，我是写到重复的问题。但是，我是考虑了重复的问题，而且还列了规律。只是被规律和测试冲昏了头脑   
****正确答案（牛客网id：Saharayu）：****

n = int(raw\_input()) a\_i = map(int,raw\_input().split()) m = 0 if len(set(a\_i)) == 1: print 1 else: for i in range(n): for j in range(i+1,n): if a\_i[i] == a\_i[j]: m = m + 1 if m != 0: print n\*(n-1)/2 - m + 1 else: print n\*(n-1)/2

找出重复的次数，最后 - m + 1 即可。   
  
  
****5.**** 牛牛喜欢字符串,但是他讨厌丑陋的字符串。对于牛牛来说,一个字符串的丑陋值是字符串中相同连续字符对的个数。比如字符串“ABABAABBB”的丑陋值是3,因为有一对"AA"和两对重叠的"BB"。现在给出一个字符串,字符串中包含字符'A'、'B'和'?'。牛牛现在可以把字符串中的问号改为'A'或者'B'。牛牛现在想让字符串的丑陋值最小,希望你能帮帮他。    
****输入描述:****

输入包括一个字符串s,字符串长度length(1 ≤ length ≤ 50),字符串只包含'A','B','?'三种字符。

****输出描述:****

输出一个整数,表示最小的丑陋值

****输入例子1:****

A?A

****输出例子1:****

0

****个人思路：**** 倒也想到了判断 。 前后的 AB，然后怎么怎么操作的。但自己又被自己阻碍了。最后没有再去想。 ****正确答案（牛客网id：大虫逗你玩）：****

import sys str = raw\_input() arr = list(str) length = len(arr) count = 0 for i in range(1,len(arr)): if arr[i] == "?": if arr[i-1] == "A": arr[i] = "B" elif arr[i-1] == "B": arr[i] = "A" for i in range(0,length-1): if arr[i] == arr[i+1] and arr[i]!="?": count = count + 1 print count

这个思路简单啊，就没有想太多。问号的前一个，来设置 。 的值，最后来判断丑陋值。但这是怎么确定最小的呢。或许根本不用考虑最小。   
****6.**** 牛家庄幼儿园为庆祝61儿童节举办庆祝活动,庆祝活动中有一个节目是小朋友们围成一个圆圈跳舞。牛老师挑选出n个小朋友参与跳舞节目,已知每个小朋友的身高h\_i。为了让舞蹈看起来和谐,牛老师需要让跳舞的圆圈队形中相邻小朋友的身高差的最大值最小,牛老师犯了难,希望你能帮帮他。   
如样例所示:   
当圆圈队伍按照100,98,103,105顺时针排列的时候最大身高差为5,其他排列不会得到更优的解    
****输入描述:****

输入包括两行,第一行为一个正整数n(3 ≤ n ≤ 20) 第二行为n个整数h\_i(80 ≤ h\_i ≤ 140),表示每个小朋友的身高。

****输出描述:****

输出一个整数,表示满足条件下的相邻小朋友身高差的最大值。

****输入例子1:****

4 100 103 98 105

****输出例子1:****

5

****个人思路：**** 想法过于简单，只想到数组的排列组合列举到明年的情况。没有去想，什么样的排列才可以得到要求。 排好序之后，一句话：由高向低的扩散，即为所求答案   
****正确答案（牛客网id：大虫逗你玩）：****

n = int(raw\_input()) a = [int(s) for s in raw\_input().split()] a.sort() max\_v = 0 for i in range(2, n): v = a[i] - a[i-2] if v > max\_v: max\_v = v print max\_v

所以，转化而知，就是求排序后间隔为2的最大的差。   
  
****7.**** 有一条无限长的纸带,分割成一系列的格子,最开始所有格子初始是白色。现在在一个格子上放上一个萌萌的机器人(放上的这个格子也会被染红),机器人一旦走到某个格子上,就会把这个格子涂成红色。现在给出一个整数n,机器人现在会在纸带上走n步。每一步,机器人都会向左或者向右走一个格子,两种情况概率相等。机器人做出的所有随机选择都是独立的。现在需要计算出最后纸带上红色格子的期望值。    
****输入描述:****

输入包括一个整数n(0 ≤ n ≤ 500),即机器人行走的步数。

****输出描述:****

输出一个实数,表示红色格子的期望个数,保留一位小数。

****输入例子1:****

4

****输出例子1:****

3.4

****个人思路：**** 看到期望，还特地的去搜了一下期望怎么求。果然世界处处皆数学啊。请同学们自行复习期望。 ****正确答案（牛客网id：草莓冰淇淋）：****

def main(): n = int(raw\_input()) print round(trans(n), 1) def trans(n): if n == 0: return 1.0 elif n == 1: return 2.0; elif n%2 == 1: return trans(n-1)\*(2\*n)/(2\*n-1); elif n%2 == 0: return trans(n-1)\*(2\*n+1)/(2\*n); main()

用了递归。我现在的问题在于我不知道递归怎么结束啊。   
  
  
****8.**** 牛牛在农场饲养了n只奶牛,依次编号为0到n-1, 牛牛的好朋友羊羊帮牛牛照看着农场.有一天羊羊看到农场中逃走了k只奶牛,但是他只会告诉牛牛逃走的k只奶牛的编号之和能被n整除。你现在需要帮牛牛计算有多少种不同的逃走的奶牛群。因为结果可能很大,输出结果对1,000,000,007取模。   
例如n = 7 k = 4:   
7只奶牛依次编号为0到6, 逃走了4只   
编号和为7的有:{0, 1, 2, 4}   
编号和为14的有:{0, 3, 5, 6}, {1, 2, 5, 6}, {1, 3, 4, 6},{2, 3, 4, 5}   
4只牛的编号和不会大于18,所以输出5.    
****输入描述:****

输入包括一行,两个整数n和k(1 ≤ n ≤ 1000),(1 ≤ k ≤ 50),以空格分割。

****输出描述:****

输出一个整数表示题设所求的种数。

****输入例子1:****

7 4

****输出例子1:****

5

****个人思路：**** 题很好懂很好理解，又有大数据问题。然后看了下面同学们的思路，一把辛酸泪 ****正确答案（牛客网id：牛客4808019）：****

#include<iostream> using namespace std; const int mod=1e9+7; //看了楼上各位大神的代码和注释，写了一点粗鄙之见。。 //这个是通过状态压缩之后的形式了，最初的形式应该是dp[i][j][s],代表的含义是从[0,i]中取j个数，使他们的和模n的余数为s，这样的j个数的集合的个数 //状态转移方程为dp[i][j][s]=dp[i-1][j][s]+dp[i-1][j-1][(s-i+n)%n]; //转移的过程就是对于第i个数，取还是不取，如果不取，那么方案数就和i-1时的方案数相同 //如果取i的话，那么分两种情况分析，第一种情况是i<=s，此时我们需要求从前i-1个数中，取j-1个数，使得他们的和模n的余数为s-i,这样就能保证在加入i时，和模n为s //第二种情况是i>s，那么s-i为负数，注意到本题要求的是组成和为n的倍数，也即所有模n为0的数，因此这种情况下将(s-i)%n表示成(s-i+n)%n即可满足要求，因为((s-i+n)%n+i)%n=s; //这个式子对于第一种情况也成立，因此和在一起就好了 int dp[55][1005]; int main() { int n,k; while(cin>>n>>k) { dp[0][0]=1; for(int i=0;i<n;i++ { for(int j=k;j>=1;j--)//状态压缩，道理和01背包一样 { for(int s=0;s<n;s++) { dp[j][s]=(dp[j][s]+dp[j-1][(n+s-i)%n])%mod; } } cout<<dp[k][0]<<endl;//最后输出取k个使他们的和模n的余数为0的所有集合的数目 } return 0; }

我是一只小小小小鸟。 

以上是2017年校招全国统一模拟笔试(第四场)编程题集合--Python的全部内容，在云栖社区的博客、问答、公众号、人物、课程等栏目也有2017年校招全国统一模拟笔试(第四场)编程题集合--Python的相关内容，欢迎继续使用右上角搜索按钮进行搜索python ，以便于您获取更多的相关知识。