地址：https://codeforces.com/contest/1371

# A Magical Sticks

- 题意：有t组数$(1 \leq t \leq 100)$，每组数有 $n$ 个数，第$i$ 个数值为$i$ , $(1 \leq i \leq n)$ , $(1 \leq n \leq 10^9 )$ ,从中取出俩数$a,b$ 然后将其合成一个新的数$a+b$,把$a+b$再插入到原来的数中，求$n$个数中最多能有几个相同的数
- 题解：签到题，对$n$分奇偶看待，若$n$为偶数，最多有$\frac{x}{2}$ 个数，若$n$为奇数，最多有 $\frac{n}{2} + 1$

# B Magical Calendar

- 题意: 规定一个日历，每周有至多$r$天，在这个日历上连续画$n$天，问有多少种可能性
- 题解： 思考，如果连续的天数$n$比每周的天数$r$多,那么只需要计算$1 \leq k \leq r$ 的情况，换句话说，目前的可能情况只与$r$有关，且可以发现，（记每种$k$下可能情况为$x$) $k = 1, x = 1; k = 2, x = 2; k = 3, x = 3 \ldots$ 最后就是一个等差数列求和,结果就是 $\frac{r*(1+r)}{2}$ 第二种情况，就是 $n r $ 现在的话，在$ 1 k n-1$ 的情况下，上面的计算方法仍然满足，但当$k == n$ 时，此时刚好能填满一周，只有一种情况。 再往后都不能填满一周了，不用考虑。 但需要注意的一点，这题数据比较大，需要开long long

# C A Cookie for You

- 题意：两种客人来Anna家做客，有两种饼干，vanilla cookies(用v代替)有$a$块，chocolate cookies，用c代替) 有$b$块，有这样一个规定，第一种客人在$v > c$情况下吃 $v$饼干，否则吃$c$饼干。第二种客人在$v > c$ 情况下吃$c$饼干，否则吃$v$饼干。 但是如果他们不能吃到自己想要的饼干就会生气，现给出$a,b$值，并给出第一种客人数量$n$,第二种客人数量$m$，问是否存在一种序列，使得客人按序列取饼干后都能满意，没有人会生气。
- 题解：虽然题意有点绕，但仔细思考后发现：我们只需要考虑第二类客人和比较客人与饼干数量就行了。具体就是饼干数中的最小值要比第二类客人多，其次饼干总数要比客人数多， ## D Grid-00100
- 题意：题目太长不想打了，贴一个原题的图
- 题解：思路大概是这样的：答案只有$0$和$2$，而我们需要尽可能的把$1$均匀的放置在行上与列上。现在先将$1$放置在主对角线上，然后再放置在右上角的副对角线上，再放在主对角左下放的副对角线上，依次交替，直到$k$为$0$，$k$能够被$n$整除，结果为$0$,反之结果为2， ## E1 Asterism (Easy Version)
- 题意，Yuzu有$x$个糖，他的对手有$n$个，$1 \leq i \leq n$ 第$i$个敌人有$a$$_{i}$个糖。为了战胜对手，Yuzu必须确定一个排列$P$，规定，如果Yuzu的糖果比敌人的$P$$_{i}$多或者相等的话，该局获胜并且赢得$1$个糖果(可在下一局使用)。为了使得每一局都获胜。问：有多少种可能的排列？ 但是又定义$f(x)$ 是$x$的排列数。已知$n$ $a$和质数$p \leq n$，且如果$f(x)$不能被$p$整除,那么正整数$x$是好的，找出所有好的整数$x$，
- 题解：定义 $m$ = $max($ $a$$_{1}$,$a$$_{2}$,$a$$_{3}$,$\ldots$,$a$$_{n}$ $)$.为了赢得比赛，Yuzu起初最少得有$m-n+1$个糖，那么在$x < m - n + 1$时，$f(x)=0$ 此时$f(x)$可以被$p$整除。如果Yuzu起初有不少于$m$颗糖的话，在$x \geq m$时，$f(x) = n!$,此时$f(x)$也可以被$p$整除。现在只需要在$m-n+1 \leq x \leq m$中找答案就行。现对传入的数组排序，枚举每个位置上的$i$可能的情况$v[i]$,最后可以知道总方案数是 $\prod_{i=0}^n v[i]$
- 代码：

```
#include <bits/stdc++.h>
```

```
using namespace std;

int main() {
     int n, p;
    cin >> n >> p;
    vector<int> a(n, 0);
    int ma = 0;
    for(int i = 0; i < n; i++) {
        cin >> a[i];
        ma = max(ma, a[i]);
    }
    sort(a.begin(),a.end());//进行一个排序
    vector<int> res;
    for(int i = 1; i <= ma; i++) {
        int pnt = 0;
        ll ans = 1;
        //枚举位置,在该位置上可以有多少个数可以选择。
        for(int j = 0; j < n; j++) {
            while(pnt < n && i + j >= a[pnt])
                pnt++;
            ans = (ans * (pnt - j) % p); //在该位置可以选择排列的个数，需要-j(前面
已经用掉的数)
        }
        if(ans != 0) //说明没有被p整除
            res.push_back(i);
    }
    cout << res.size() << endl;
    for(auto i : res) {
        cout << i << " ";
    }
    cout << endl;
}
```

# E2 Asterism (Hard Version)

-题意：先咕咕，会补的，别催了 -题解 ## F Raging Thunder -题意：先咕咕，会补的，别催了 -题解：

From:
https://wiki.cvbbacm.com/ - **CVBB ACM Team**

Permanent link:
**https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:manespace:codeforces_round_654_div._2&rev=1593840198**

Last update: **2020/07/04 13:23**