

## A

找出三个数中最大的最小的和中间的点，以中间的点为基准，把最小的最大的点分别向两边移动，直到与中间点的距离大于等于 $d$ 为止，计算答案即可。

## B

输入每个人的出拳时统计出石头，剪刀，布的人数，如果三种都有人出，或者总共只出了一种就输出-1。如果输出有2种，判断出胜利的类型，遍历所有人的出拳结果，输出获胜者。

## C

三重for循环计算矩阵的每个元素

```
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++) {
        for (int h = 1; h <= n; h++) {
            c[i][j] += (a[i][h] * b[h][j]);
        }
    }
}
```

## D

答案为 $a^b$ ，注意 $a = 256$ ， $b = 8$ 时的答案超出`unsigned long long`的数据范围，特判一下输出 $18446744073709551616(2^{64})$ ，`unsigned long long`最大的数据范围 $2^{64} - 1$

## E

按照题意模拟即可

## F

找出元素最多的子集，使最大元素和最小元素之差不大于 $n$ 。双指针算法，排序后， $l, r$ 指针都是递增的，不会出现在前面，只需左指针从前向后扫一遍，找与它差 $n$ 的数所组成的最多子集。

## G

先考虑 $ans$ 的计算方法。先把需要在一起的两个人看成一个人，有 $n$ 种坐法，其中两人内部又有2种坐法，其余的人有 $(n-1)!$ 种坐法，又要考虑旋转的情况，所以 $\frac{2*n*(n-1)!}{n} (n > 2)$ ， $n = 2$ 时， $ans = 2 * 2 * (2-1)/2$ ，注意到此时的剩余的2是由于两人内部产生的，但 $n = 2$ ，无论怎么坐都会通过旋转而重合，所以 $ans = 1$ 。然后考虑到最后输出的要对1000000007取模，以及存在三个阶乘，所以只需找到 $n$ 为多大时， $(ans!)!$ 大于1000000007。经计算 $n$ 为4时，符合上述情况，所以 $n < 4$ 时，输出对应的结果，大于等于4时输出0。

## H

题意就是给你一个数，将其表示为三个素数 $a, b, c$ 之和，并且让 $a \leq b \leq c$ ，其中 $a, b$ 尽可能小。其实本题就是一个弱化的哥德巴赫猜想，题目让 $a, b$ 尽可能小，已经明示了要遍历，并且范围在3000内，首先用一个数组存下范围内素数，然后一个二维循环就可以完成。

## I

如果某节点的父节点有镜像节点，那么如果该节点为左儿子，则他的镜像为父节点镜像右儿子，反之同理。预先对根节点的左右儿子标记，然后 $dfs$ 。

## J

给定 $n$ 个字符串， $m$ 个询问，字符串仅包含大写与小写字母，询问时候输入 $l, r$ ，输出 $l, r$ 之间字典序最大的字符串

解法1：直接 $st$ 表上就可以，边界条件 $st[i][0] = (\text{原字符串})$ ，然后可以直接比较大小就可以，转移方程( $maxx$ 的意义类似 $max$ ):  $st[i][j] = maxx(st[i][j-1], st[i + (1 << (j-1))][j-1])$ ; 询问: 答案为  $maxx(st[l][\text{int}(\log_2(r-l+1))], st[r - (1 << \text{int}(\log_2(r-l+1))) + 1][\text{int}(\log_2(r-l+1))])$

解法2:  $trie$ 树存下所有的字符串，然后 $dfs$ 一遍跑字典序大小，然后就是个 $rmq$ ，之后可以线段树。

## K

将房间要求速度信息处理成递增。(如果当前房间的要求速度比前一个低，那么把当前房间要求速度改为与前一个房间要求相同也不会影响结果。)然后每个人从后往前找第一个被困住的房间。(后指的是当前能到达的最后，如果被锁住要从锁住的前一个开始判断)，然后存一下输出即可

## L

假设我们现在算出了合并前的答案，在合并 $x$ 和 $y$ 时，设 $sz[x]$ 为 $x$ 所在集合的集合大小， $sz[y]$ 同理。考虑这两个集合对答案的贡献。有三种情况：

1. 从 $x$ 所在集合中取一个人，然后再从其他非 $y$ 集合中挑选出两个互不在同一集合的人
2. 从 $y$ 所在集合中取一个人，然后再从其他非 $x$ 集合中挑选出两个互不在同一集合的人
3. 从 $x, y$ 所在集合中各取一个人，然后再从其他集合中挑选出两个互不在同一集合的人

可以发现合并之后 $x$ 和 $y$ 在同一集合，仔细观察上面说到的情况1, 2，它们对答案的贡献并没有因为合并操作而改变。只有情况3，在合并之后，该贡献被消灭，所以要用上一次的答案减去这个情况，就是合并之后的答案。用并查集维护集合元素的个数，每次合并时更新答案即可。