A_ZZULI

这道题由于要计算连通块大小，可以使用并查集，关键在于如何去合并。
对于 a,b,c ，(a,b) 合并后 (b,c) 合并，与 (a,b) 合并再 (a,c) 合并是没有区别的，那么要合并的一组数就用第一个去合并一遍。
所以拿第一个 `Z` 向后合并一遍，遇见 `Z` 、`U`、 `L`、 `I` 就合并;
再拿第一个 `U` 按照规则向后合并一遍，再拿第一个 `L` 按照规则向后合并一遍，再拿第一个 `I` 按照规则向后合并一遍即可

B_大本营

如果大本营在其中一个守卫的侦察范围（圆）内，那么是不可行的。
考虑大本营在圆外的情况，不可行的情况就是大本营被几个圆围起来，所以把相交（切）的守卫的圆心连起来（建边）。
因为一个守卫最多与其他两个守卫相交，所以一个圆心的度最大值为2，那么建的边只能构成链或者是环，最后判断大本营是否在多边形内即可。

C_最大公因数

由于 $gcd(a,b)=x$ ，则 $gcd(\frac{a}{x},\frac{b}{x})=1$ ，那么这道题就是让我们找 $[l,r]$ 内两个数，其不仅是 x 的倍数，在除完 x 后也要互质。
两个相邻的整数是互质的，所以这里找 $[l,r]$ 内两个相邻的 x 的倍数就行
例：若存在，$gcd(r/x,r/x-1)=1$ ，$gcd(r/x*x,(r/x-1)*x)=x$

D_大盗

每个物品拿或者不拿，一个比较明显的01背包。
用数组维护到达每一个房间的背包状态，遇到藏品时所有状态可以加也可以不加，遇到三体人时保留对应状态即可。
使用数组暴力更新的复杂度为 $O(nk)$ ，会超时，考虑优化。
一个 `bool` 类型的 dp ，且只有加法和清空非保留的状态，可以对应到 01 串也就是 `bitset` ，如果其是用 `int` 维护的话时间为 $O(\frac{nk}{32})$
在遇到藏品，加为 $dp<<a_i$ ，不加为 dp ，合并就是 $dp<<a_i|dp$
遇到三体人时看一下这一位是 0 还是 1 就行了
最后暴力从 $k-1$ 向下找

E_睡大觉

因为是将其换算成最近的日期，所以我们记录一个当前的时刻，如果下一个给定的时间比当前的时间大我们就可以是为仍在这一天，否则就向后走一天。同时需要注意的是，因为他每次至少睡1s，所以如果与我们当前的时刻相同的话也要向后推一天，接下来就是算出当前是哪一天判断月与日是否就相同就可以了。

F_小欧拉

本题由于最终式子过于常见，解法多为打表找规律，下面是推导:

看累加符号右边，有一堆奇奇怪怪的东西，最奇怪的当然是 ϕ ，推一下

考虑欧拉函数的计算式，设 $p \in prime$ ，则 $\phi(x)=x\prod\limits_{p|x}\frac{p-1}{p}$ ，发现这里有个 x ，这和 $\frac{\phi(gcd(i,j))}{gcd(i,j)}$ 简直天造地设的一对，那么约分掉

先不看下取整符号

$$\frac{\phi(ij)\phi(gcd(i,j))}{gcd(i,j)}=ij(\prod_{p|ij}\frac{p-1}{p})(\prod_{p|gcd(i,j)}\frac{p-1}{p})$$

这里 i,j 共有的质因数前面的括号只被算了一次，而后面的括号是让它成功计算第二次，分配下来就是 i 的每个质因数参与计算一次，j 的每个质因数参与计算一次。

这就和 $(i\prod\limits_{p|i}\frac{p-1}{p})(j\prod\limits_{p|j}\frac{p-1}{p})$ 是一样的了，而这个式子就是 $\phi(i)\phi(j)$。

可推原式就是 $\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{M}\phi(i)\lfloor\frac{N}{i}\rfloor\phi(j)\left\lfloor\frac{M}{j}\right\rfloor=\sum\limits_{i=1}^{N}\phi(i)\lfloor\frac{N}{i}\rfloor\sum\limits_{j=1}^{M}\phi(j)\left\lfloor\frac{M}{j}\right\rfloor$ ，这两部分是一样的，那么只推导出来一侧就可以得出解了。

将 $\lfloor\frac{N}{i}\rfloor$ 放进计数循环，即前面设置一个累加为 $\sum\limits_{i=1}^{N}\sum\limits_{n=1}^{\lfloor\frac{N}{i}\rfloor}\phi(i)$

转换为 n 枚举 i 的倍数为 $\sum\limits_{i=1}^{N}\sum\limits_{i|n}\phi(i)$

后面那个东西就是常见的卷积 $\phi*1=Id$ 形式（下面有证明）

所以 $\sum\limits_{i|n}\phi(n)=n$ ，$\sum\limits_{i=1}^{N}n=\frac{N(N+1)}{2}$

那么 $\sum\limits_{i=1}^{N}\phi(i)\lfloor\frac{N}{i}\rfloor\sum\limits_{j=1}^{M}\phi(j)\frac{M}{j}=\frac{N(N+1)}{2}\frac{M(M+1)}{2}$

结果已出，看数据量开个 `int128` 就可以了

> **证明** $\phi*1=Id$ ：
> $\forall d|m,1\le\alpha\le n$ ，有 $(\alpha,n)=d\Longrightarrow(\frac{\alpha}{d},\frac{n}{d})=1$
> 这样的 α 可以选出来 $\phi(\frac{m}{d})$ 个 （m 下有多少个数是 d 的倍数）
> 所以 $n=\sum\limits_{d|n}\phi(\frac{m}{d})$ ，即 $(\phi*1)(n)=Id(n)$
> 得证

G_迷宫

最短路问题，两两建边，如果目标点为花那么边权为 $1-1=0$ ，否则为 1

固定源点与汇点，最短路的各种算法都可以写

H_安全区

初始的位置并不固定，同时又因为趋于无穷步。那么此时小明在每一个无毒迷失的概率都是相同的，在进行到一定轮之后，其每一步的情况与概率都是相同的。

那么我们此时应计算出下一步的所有情况，以此来作为答案。

所以答案的表示应为$\frac{下一步能够到达安全区的步数}{下一步可以走的总步数}$，暴力的方法就是统计每一个无毒密室下一步的情况然后相加就可以计算出，但是我们n,m都很大无法计算出，所以我们就是假设每一个都是无毒密室，然后计算出相应的步数，随后遍历每一个有毒密室，我们只需将从这个有毒密室走出去的情况减去，并且更新一下从其他无毒密室走进来的情况（你可能在一在安全区的无

毒密室走进来又退回到了安全区）。

I_密集

过去的时间越久剩下的蜗牛的数量也会越少，存在单调性，所以我们使用二分。
对于原本顺序为 A，B，C 的三个蜗牛，如果 B 先超过了 C，然后 A 超过了 B 的话，那么 A 也一定能超过 C。
我们想一下如果想要知道在某一时刻那些蜗牛已经退赛的话，就是看有哪些蜗牛的相对顺序发生了变化。
那么 $check$ 函数中我们只需要求出每一个蜗牛按照此时的位置排序，从最远的位置向前遍历，更新一个此时的编号的最小值 Min，如果当前这个蜗牛的编号比 Min 大，就说明其被超过了，这样就可以统计出来被超过的数量。
同时需要注意的是此时刻两个蜗牛位于同一个位置不算超过。

J_苹方树

本题树上差分和树链剖分都可做，这里说一下树上差分的写法。
由于询问在所有操作之后，那么就操作完离线向上跑一下 `dfs` 处理下结果然后输出。
操作很多，一次操作一条路径，可以打一个标记。如果都乘起来最大可以达到 $100^{4\times10^{10}}$ 显然不可以直接乘，但由于平方数的判定方式是所有质因数的数量为偶数，所以将标记定为 w 的质因数与其对应的数量（$\sqrt{n}$ 分解就可以）。
对于每次操作路径 u,v ，在向上走到 $lca(u,v)$ 前，u 与 v 对于各自上面的祖先都起作用，所以在 u 与 v 这两个节点都存一下 w 的质因数与对应数量，但走到 $lca(u,v)$ 之后，如果两个子树的信息都保留会造成一条路被计算两次的情况，所以要存一下 w 的质因数与其对应数量的相反数来让这条路对完整包含这条路的子树根节点的贡献只有一次。
存完之后，将所有节点我们存入的信息进行向上合并，每个节点 i 在用自己的信息合并完所有的子节点 $son[i]$ 的信息之后，暴力判断一下是否存在出现次数为奇数的质因数，如果存在那么 v_i 就不是平方数，否则是平方数。

K_试香

a_i 是 2 的幂，判断 x 能不能由其中几个 a_i 构成即可。

L_固执

我们可以发现对于当前的字符串，如果出现次数最多的那个字符的数量大于等于 $n/2+1$，我们就无法构造出合法的字符串，又因为要保证字典序最小，所以我们对于每一位，我们从 `a` 到 `z` 去遍历，如果将这个字符位于这一位，验证剩下的字符时候是否对于下一个后缀是否还满足上面的关系就可以了。