

T1 智慧食堂

思路

- 读入三个数 a, b, c
- 读入支付方式 op
- 如果 $op == 2$ (现金)
→ 输出 `Fail to pay.`
- 否则
→ 输出 $a + b + c$

本题考察

- 基础输入输出
- `if-else` 条件判断
- 简单加法运算

T2 朱猪时刻

思路

关键只看 **分钟数 mm** 。

“朱猪时刻”的条件是：

$$mm \% 5 == 3$$

做法

1. 读入 hh 和 mm
2. 判断：
 - 如果 $mm \% 5 == 3$
→ 输出 0
 - 否则
→ 每次把分钟 $+1$ ，统计加了多少次
→ 直到满足 $mm \% 5 == 3$
→ 输出增加的次数

因为最多只会加 4 分钟，所以可以直接循环 0~4 次判断。

也可以更快一点直接算：

$$\text{delta} = (3 - \text{mm} \% 5 + 5) \% 5$$

输出 `delta` 即可。

本题考察知识点

1. 取模运算 `%`
2. 时间格式读入（字符串或两个整数）
3. 简单数学规律观察
4. 小范围循环或公式推导
5. 边界理解（如果本身满足要输出 0）

T3 训练记录

思路

这是一道**模拟题**，按天顺序处理即可。

一、需要维护两个变量

1. `h` —— 当前愉悦值（初始为 0）
2. `cnt` —— 已经进行的模拟赛场数（初始为 0）

二、每天分类讨论

读入当天状态 `a_i`，分情况处理：

① 如果是休息日 (`a_i == 0`)

- 愉悦值 `+100`

② 如果是工作日

- 如果 `a_i == 2`（没有模拟赛）
→ 愉悦值不变
- 如果 `a_i == 1`（有模拟赛）
→ 模拟赛次数 `cnt++`

→ 判断奇偶：

- 如果 cnt 是奇数
→ 愉悦值 +50
- 如果 cnt 是偶数
→ 愉悦值 -30

三、每天处理完后输出当前愉悦值

按顺序模拟 d 天即可。

本题考察知识点

1. 模拟思想（按顺序逐天处理）
2. 计数器的使用（记录第几场模拟赛）
3. 奇偶判断（ $\text{cnt} \% 2$ ）
4. 条件分支结构
5. 状态维护能力

T4 你这题至少 *3000

思路

每道题 **最多能放两个位置**：

- 第 a_i 个位置
- 第 $a_i + 1$ 个位置
- 如果 $a_i = m$ ，则 **只能放第 m 个位置**

我们只需要判断：**每个位置有没有题目可以放。**

做法

1. 开一个数组 $\text{vis}[1..m]$ ，表示这个位置是否有题可以放。
2. 遍历每道题：
 - 标记 $\text{vis}[a_i] = \text{true}$
 - 如果 $a_i < m$ ，再标记 $\text{vis}[a_i + 1] = \text{true}$
3. 遍历 $1..m$ ：
 - 如果 $\text{vis}[i] == \text{false}$
→ 说明没有任何题适合这个位置

→ 这个位置一定空缺

本题考察知识点

1. 数组标记思想 (vis 标记法)
2. 区间覆盖理解 (每题覆盖 a_i 和 a_{i+1})
3. 模拟统计
4. 线性扫描
5. 时间复杂度优化

时间复杂度:

- 标记: $O(n)$
- 扫描: $O(m)$
- 总复杂度: $O(n + m)$ (可以处理 10^5 数据)

T5 蓝玫瑰

思路

设选择的花数为:

- 红色: a'
- 粉色: b'
- 白色: c'

我们只需要**枚举可能的组合**并判断是否满足条件即可。

一、枚举范围

根据题目限制:

- $0 \leq a' \leq \min(a/2, 10)$ (红色最多 10)
- $0 \leq b' \leq b$
- $c/2 \leq c' \leq c$

因为要求**至少一支花**, 所以 $(a' + b' + c' > 0)$ 。

二、判断条件

每次枚举 (a', b', c') 时检查：

1 粉色数量最多

$$a' \leq b'$$

$$c' \leq b'$$

2 颜色比例限制

如果 $b' == 0$ 或 $c' == 0$

→ 条件自动满足

否则要求：

$$a'/c' \leq c'/b'$$

为了避免浮点数，可以写成：

$$a' * b' \leq c' * c'$$

3 红色限制

$$a' \leq 10$$

4 至少一朵花

$$a' + b' + c' > 0$$

三、统计答案

只要 (a', b', c') 满足所有条件，就把答案 +1。

时间复杂度

由于

$$a' \leq 10$$

所以枚举规模约为：

$$10 \times 5000 \times 5000$$

但可以通过限制 $b' \geq \max(a', c')$ 等条件减少枚举，实际数据可以通过。

本题考察知识点

1. 三重枚举（暴力枚举）
2. 数学不等式转化（避免浮点数）
3. 条件判断与剪枝
4. 边界情况处理（ $b'=0$ 或 $c'=0$ ）
5. 计数类问题建模

核心代码

```
for (int i = 0; i <= min(a/2, 10); i++)
{
    for (int j = 0; j <= b; j++)
    {
        for (int k = c/2; k <= c; k++)
        {
            if (i+j+k < 1) continue;
            if (i > j || k > j) continue;
            if (!j || !k) cnt++;
            else if (i*j <= k*k) cnt++;
        }
    }
}
```

T6 逃避讲题的最好方法是

思路

关键目标：

小 S 要避免在任何一道题成为最高分或最低分。

如果在某题中：

- 是 **最高分**（包括并列）
- 或 **最低分**（包括并列）

就有可能被选中上台讲题。

一、逐题判断

对于每一道题 j ：

1. 统计这道题所有学生的
 - \max_j （最高分）
 - \min_j （最低分）
2. 看小 S 的分数 $a[x][j]$

如果：

$$a[x][j] == \max_j \quad \text{或} \quad a[x][j] == \min_j$$

说明：

- 小 S 在这道题 **一定会触发风险**
- 必须 **修改这一题的分数**

否则：

- 这题已经安全，不需要改。

二、统计需要修改的题数

设需要修改的题数为 $need$ 。

- 如果 $need > c$
 - 修改次数不够
 - 小 S **仍然可能被选中**
 - 输出 up
- 如果 $need \leq c$
 - 可以全部修改掉
 - 小 S **不可能被选中**
 - 输出 down

考虑特殊情况，如果当前最高分和最低分差值小于1，那么就一定会上台

时间复杂度

每组数据：

$$O(n \times m)$$

最大：

$$50 \times 300 \times 300$$

可以轻松通过。

本题考察知识点

1. 矩阵数据处理
2. 最大值 / 最小值统计
3. 模拟与计数
4. 逻辑判断
5. 多组测试处理

参考代码

```
#include <bits/stdc++.h>
using namespace std;
int t, a[305][305];
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cin >> t;
    while (t--)
    {
        int n, m, x, c, f = 0, s = 0;
        cin >> n >> m >> x;
        for (int i = 1; i <= n; i++)
            for (int j = 1; j <= m; j++) cin >> a[i][j];
        cin >> c;
        for (int j = 1; j <= m; j++) // 枚举m道题
        {
            int minx = 100, maxx = 0;
            for (int i = 1; i <= n; i++) // 枚举n个人
            {
                if (x == i) continue;
                minx = min(minx, a[i][j]);
                maxx = max(maxx, a[i][j]);
            }
            if (maxx - minx <= 1)
            {
                f = 1; break;
            }
            if (a[x][j] >= maxx || a[x][j] <= minx) // s的分值>=maxx, 或者<=minx, 统计
            {
                s++;
            }
        }
        if (f || s > c) cout << "up\n";
        else cout << "down\n";
    }
    return 0;
}
```

T7 2014

思路

关键点：可以交换任意两个字符。

这意味着：

字符串的顺序不重要，只要字符数量能拼出来即可。

一、观察 2014 串的结构

一个 2014 包含：

- 2 ×1
- 0 ×1
- 1 ×1
- 4 ×1

如果有 k 个 2014：

- 2 有 k 个
- 0 有 k 个
- 1 有 k 个
- 4 有 k 个

总长度：

$$n = 4k$$

二、需要满足的条件

1 长度必须是 4 的倍数

$$n \% 4 == 0$$

2 四种数字数量必须相同

统计字符串中：

- cnt2
- cnt0
- cnt1
- cnt4

必须满足：

$$\text{cnt2} == \text{cnt0} == \text{cnt1} == \text{cnt4} == n/4$$

3 不能有其他数字

如果出现 3,5,6,7,8,9 等数字
→ 一定无法组成 2014 串。

时间复杂度

只需要 统计一次字符串：

$$O(n)$$

可以轻松处理 $n \leq 10^6$ 。

本题考察知识点

1. 字符串统计
2. 贪心思想（顺序可忽略）
3. 字符频率统计
4. 数学规律观察
5. 大数据规模处理 $O(n)$

T8 物品分组

思路

目标是：

把连续的物品分成 k 组，让 每组价值和的最大值尽量小。

这是典型的 **二分答案 + 贪心检查** 问题。

一、为什么可以二分答案

设答案为 x ：

表示 **每一组的价值和都不能超过 x** 。

如果 x 可以做到分成 $\leq k$ 组，那么：

- 更大的 x 一定也可以做到。

如果 x 做不到：

- 更小的 x 也一定做不到。

因此具有 **单调性**，可以 **二分答案**。

二、二分范围

最小值： $\max\{a_i\}$

因为一组至少要包含一个物品。

最大值： $\sum a_i$

全部放一组。

三、如何判断某个 x 是否可行

贪心扫描数组：

- 当前组累加价值
- 如果加入下一个物品会 **超过 x**
 - 就 **新开一组**

统计需要多少组 cnt 。

最后判断： $cnt \leq k$ ，说明 x 可行。

四、算法流程

1. 二分 x
2. 每次用贪心检查需要多少组

3. 如果 $\leq k$
→ 尝试更小答案
4. 否则
→ 增大答案

最终得到最小可行值。

五、时间复杂度

$$O(n \log sum)$$

在 $n \leq 1000$ 下非常快。

六、本题考察知识点

1. 二分答案
2. 贪心分组
3. 单调性判断
4. 连续区间划分问题
5. 数组扫描

参考代码

```
#include <bits/stdc++.h>
using namespace std;
const int N = 1e3+5;
int n, a[N], k, ans;
bool check(int x)
{
    int cnt = 1, s = 0;
    for (int i = 1; i <= n; i++)
    {
        if (s + a[i] <= x) s += a[i];
        else cnt++, s = a[i];
    }
    return cnt <= k;
}
int main()
{
    cin >> n;
    int l = 0, r = 0;
    for (int i = 1; i <= n; i++) cin >> a[i], l = max(l, a[i]), r += a[i];
    cin >> k;
    // cout << l << " " << r << "\n";
    while (l <= r)
    {
        int mid = (l + r) >> 1;
        if (check(mid)) ans = mid, r = mid-1;
        else l = mid + 1;
    }
    cout << ans;
    return 0;
}
```