

有趣的跳跃

人员

温郝冬、李沛都、罗启宸、吴青汉、王奕皓、周苇杰、陶汇笙、张昱霖、康佳、锁喜栋、郭栩睿到课

作业检查

温郝冬 未打卡

李沛都 未打卡

罗启宸 已完成

吴青汉 已完成

王奕皓 未打卡

周苇杰 已完成

陶汇笙 已完成

张昱霖 未打卡

康佳 已完成

锁喜栋 未打卡

郭栩睿 已完成

作业

noi 1.6 07 有趣的跳跃	必做题
东方博宜 1472 - 找筷子	选做题

课堂表现

课堂整体纪律不是很好，同学们以后上课要注意自己的纪律问题，不要让老师一遍一遍的强调

课堂内容

东方博宜 1334 - 扑克牌组合

定义数组 `f[k]`: `k`这个数出现的次数

两两匹配, 统计哪些数出现过, 让`f[k]++`

```
for (int i = 1; i <= n; i++) {  
    for (int j = i+1; j <= n; j++) {  
        int t = a[i] + a[j];  
        f[t]++;  
    }  
}
```

计数器统计一共有多少个数出现过

```
int sum = 0;  
for (int i = 1; i <= 100; i++) {  
    if (f[i] >= 1) {  
        sum++;  
    }  
}  
cout << sum << endl;
```

输出所有出现过的数

```
for (int i = 1; i <= 100; i++) {  
    if (f[i] >= 1) {  
        cout << i;  
    }  
}
```

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 52 + 5;
int w[maxn], vis[maxn];

int main() {
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) {
        cin >> w[i];
    }

    for (int i = 1; i <= n; ++i) {
        for (int j = i+1; j <= n; ++j) {
            int t = w[i] + w[j];
            vis[t]++;
        }
    }

    int sum = 0;
    for (int i = 1; i <= 50; ++i) {
        if (vis[i] > 0) {
            ++sum;
        }
    }
    cout << sum << endl;

    for (int i = 1; i <= 50; ++i) {
        if (vis[i] > 0) {
            cout << i << " ";
        }
    }
    cout << endl;
    return 0;
}

```

东方博宜 1260 - 三色球问题

一共3个红球，3个白球，6个黑球

所以可以选择 0/1/2/3 个红球，0/1/2/3 个白球，0/1/2/3/4/5/6 个黑球

总共有 $4 \times 4 \times 7 = 112$ 种方案

接下来，要在这112种方案中，找出来有多少种方案有 8 个球

可以通过三重for循环枚举所有方案，判断有多少满足条件(即有8个球)的方案

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int ans = 0;
    for (int i = 0; i <= 3; ++i) {
        for (int j = 0; j <= 3; ++j) {
            for (int k = 0; k <= 6; ++k) {
                if (i+j+k == 8) {
                    ans++;
                }
            }
        }
    }
    cout << ans << endl;
    return 0;
}

```

noi 1.6 07:有趣的跳跃

1. 输入n
2. 输入n个数 a[1]~a[n]
3. 求n个数的n-1个差的绝对值 -> b[1]~b[n-1]
4. 判断 b[1]~b[n-1] 是否正好出现过 1~n-1
5. 开一个数组f, f[i]代表 i 这个数出现了多少次
6. 只需要统计 1 ~ n-1 这个范围里的数
7. 判断f数组是否满足 f[1]==1, f[2]==1, ..., f[n-1]==1 这个条件
 - > 满足: Jolly
 - > 不满足: Not jolly
8. 只要存在一个 f[i]!=1 的话, 一定是 Not jolly, 程序结束
9. 开一个计数器, 判断 f[1], f[2], ..., f[n-1] 中有几个数满足等于 1
10. 计数器==n-1 -> Jolly
 - 计数器!=n-1 -> Not jolly

```

#include <bits/stdc++.h>

using namespace std;

int a[3005], b[3005];
int f[3005];

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
    }
    for (int i = 1; i <= n-1; i++) {
        b[i] = a[i] - a[i+1];
        if (b[i] < 0) {
            b[i] = -b[i];
        }
    }

    // f[i]: i 这个数出现的次数 1~n-1
    for (int i=1; i<=n-1; i++) {
        int t=b[i];
        if (t>=1 && t<=n-1) {
            f[t]++;
        }
    }

    int sum = 0;
    for (int i = 1; i <= n-1; i++) {
        if (f[i] == 1) {
            sum++;
        }
    }
    if (sum == n-1) {
        cout << "Jolly";
    }
    else {
        cout << "Not jolly";
    }
    return 0;
}

```