

<div>题目描述：</div> <p>某部门计划通过结队编程来进行项目开发，已知该部门有 N 名员工，每个员工有独一无二的职级，每三个员工形成一个小组进行结队编程，结队分组规则如下：</p> <p>从部门中选出序号分别为 i、j、k 的 3 名员工，他们的职级分别为 $level[i]$、$level[j]$、$level[k]$</p> <p>结队小组需满足：$level[i] < level[j] < level[k]$ 或者 $level[i] > level[j] > level[k]$，其中 $0 \leq i < j < k < n$</p> <p>请你按上述条件计算可能组合的小组数量。同一员工可以参加多个小组。</p> <div>输入描述：</div> <p>第一行输入：员工总数 n</p> <p>第二行输入：按序号依次排列的员工的职级 $level$，中间用空格隔开</p> <p>限制：</p> <p>$1 \leq n \leq 6000$</p> <p>$1 \leq level[i] \leq 10^5$</p> <div>输出描述：</div> <p>可能组合的小组数量</p> <div>补充说明：</div> <p>示例 1</p> <div>输入：</div> <div>4</div> <div>1 2 3 4</div> <div>输出：</div> <div>4</div> <div>说明：</div> <p>可能结队成的组合 (1, 2, 3)、(1, 2, 4)、(1, 3, 4)、(2, 3, 4)</p> <p>示例 2</p> <div>输入：</div>
--

3

5 4 7

输出:

0

说明:

根据结队条件，我们无法为该部门组建小组

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

void calc1(int level[], int count[], int n){
    memset(count, 0, sizeof(count));
    for (int i = 0; i < n; i++) {
        int c = 0;
        for (int j = i+1; j < n; j++) {
            if (level[j] > level[i])
                c++;
        }
        count[i] = c;
    }
}

void calc2(int level[], int count[], int n){
    memset(count, 0, sizeof(count));
    for (int i = 0; i < n; i++) {
        int c = 0;
        for (int j = i+1; j < n; j++) {
            if (level[j] < level[i])
                c++;
        }
        count[i] = c;
    }
}

int main() {
    int n;
    int level[6000], count[6000];
    while (cin >> n) { // 注意 while 处理多个 case
        for (int i = 0; i < n; i++)
            cin >> level[i];
        int ans = 0;
        calc1(level, count, n);
        for (int i = 0; i < n; i++)
            for (int j = i+1; j < n; j++) {
                if (level[i] < level[j])
```

```
        ans += count[j];
    }

    calc2(level, count, n);
    for (int i = 0; i < n; i++)
        for (int j = i+1; j < n; j++) {
            if (level[i] > level[j])
                ans += count[j];
        }
    cout << ans << endl;
}
}
// 64 位输出请用 printf("%lld")
```