

杂题练习2

人员

左子毅、朱奕鸣、杨洋、刘子淇、赵清航、王崇宇、于珈浩、刘佳赫 到课

作业检查

左子毅 未完成

朱奕鸣 未完成

杨洋 未完成

刘子淇 已完成

赵清航 未完成

王崇宇 上周请假

于珈浩 未完成

刘佳赫 上周请假

作业

<https://www.luogu.com.cn/contest/178178>, 课上讲的题目, 要求同学们课下补题

<https://www.luogu.com.cn/contest/176989>, 课上没讲的题目, 要求同学们课下思考完成

课堂表现

刘子淇同学课上做题比较好, 提出表扬!!

左子毅、杨洋 2位同学细节没考虑好, A 题都出错了, 以后要注意, 做题要细心。

课堂内容

CF1858A Buttons

```
#include <bits/stdc++.h>

using namespace std;

void solve() {
    int a, b, c; cin >> a >> b >> c;
    a += (c+1)/2, b += c/2;
    if (a > b) cout << "First" << endl;
    else cout << "Second" << endl;
}
```

```
int main()
{
    int T; cin >> T;
    while (T -- ) solve();
    return 0;
}
```

CF1913B Swap and Delete

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;
char str[maxn];

void solve() {
    cin >> (str+1);
    int n = strlen(str+1);

    int cnt0 = 0, cnt1 = 0;
    for (int i = 1; i <= n; ++i) {
        if (str[i] == '1') cnt1++;
        else cnt0++;
    }

    for (int i = 1; i <= n; ++i) {
        if (str[i] == '1') cnt0--;
        else cnt1--;
        if (cnt0 < 0 || cnt1 < 0) {
            // i-1 没问题
            // 保留 i-1 个数
            // 删除 n - (i-1) 个数
            cout << n - (i-1) << endl;
            return;
        }
    }
    cout << 0 << endl;
}

int main()
{
    int T; cin >> T;
    while (T -- ) { solve(); }
    return 0;
}
```

CF1884B Haunted House

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
char str[maxn];

void solve() {
    int n; cin >> n;
    cin >> (str+1);
    LL res = 0;

    int pos = n;
    for (int i = n; i >= 1; --i) {
        if (str[i] == '0') {
            res += pos-i;
            cout << res << " ";
            pos--;
        }
    }

    for (int i = pos; i >= 1; --i) cout << -1 << " ";
    cout << endl;
}

int main()
{
    int T; cin >> T;
    while (T -- ) solve();
    return 0;
}

```

CF1856C To Become Max

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1000 + 5;
int w[maxn];

bool dfs(int pos, int m, int mid, int n) {
    // 使用 m 次操作，把 w[pos] 这个数变成 mid，数组一共有 n 个数
    if (m < 0) return false;
    if (w[pos] >= mid) return true;
    if (pos == n) return false;

    // w[pos] 要加多少次? -> d
    int d = mid - w[pos];

```

```
    if (dfs(pos+1, m-d, mid-1, n)) return true;
    return false;
}

bool check(int n, int m, int mid) {
    for (int i = 1; i <= n; ++i) {
        if (dfs(i, m, mid, n)) return true;
    }
    return false;
}

void solve() {
    int n, m; cin >> n >> m;
    for (int i = 1; i <= n; ++i) cin >> w[i];

    int l = 1, r = 2e8;
    while (l <= r) {
        int mid = (l + r) / 2;
        if (check(n, m, mid)) l = mid+1;
        else r = mid-1;
    }

    cout << r << endl;
}

int main()
{
    int T; cin >> T;
    while (T -- ) { solve(); }
    return 0;
}
```