

综合混练

人员

褚锦轩、许睿谦、王毅博、阮文璋、王承周、司云心 到课

上周作业检查

https://www.luogu.com.cn/contest/240400

2025-0406六队上课(综合混练)

报名

编辑比赛

题目数

4

报名人数

23

比赛说明

题目列表

排行榜

名次	参赛者	总分	A	B	C	D
#1	杨俊彦	400 (19.38h)	100 (2.38h)	100 (350ms)	100 (4.05h)	100 (12.96h)
#2	徐思远	400 (8.82d)	100 (2.19d)	100 (2.19d)	100 (2.20d)	100 (2.24d)
#3	阮文璋	400 (13.15d)	100 (2.54d)	100 (2.54d)	100 (3.55d)	100 (4.51d)
#4	刘奕辰	400 (14.06d)	100 (71ms)	100 (361ms)	100 (7.03d)	100 (7.04d)
#5	杨咏丞	400 (18.13d)	100 (2.56d)	100 (2.56d)	100 (5.53d)	100 (7.48d)
#6	袁晨峻	400 (26.27d)	100 (6.55d)	100 (6.55d)	100 (6.58d)	100 (6.59d)
#7	王承周	300 (7.31d)	100 (2.44d)	100 (2.43d)	100 (2.44d)	
#8	王毅博	280 (14.24d)	80 (0ms)		100 (7.12d)	100 (7.12d)
#9	李锦澍	210 (9.30h)	100 (70ms)		10 (0ms)	100 (9.30h)
#10	周治润	200 (150ms)	100 (68ms)	0 (0ms)	100 (82ms)	
#11	许睿谦	200 (3.11d)	100 (1.55d)	100 (1.55d)		
#12	SSJ司云心	200 (7.87d)	100 (1.49d)	100 (6.39d)		
#13	曹嫿	100 (66ms)	100 (66ms)			
#14	王陆文龙	100 (76ms)	100 (76ms)			
#15	龙沛轩	100 (78ms)	100 (78ms)			
#16	韩鸣蔚	100 (87ms)	100 (87ms)	0 (0ms)		
#17	潘俊伊	70 (0ms)	60 (0ms)	0 (0ms)	10 (0ms)	
#18	李雨谦	70 (0ms)	70 (0ms)	0 (0ms)	0	
#19	褚锦轩	40 (7.21d)	40 (7.21d)			
#20	王博涵	0 (0ms)	0 (0ms)			

作业

<https://www.luogu.com.cn/contest/241021> (课上讲了 A ~ C 题, 课后必做作业是 E 题, 选做作业是 D 题)

课堂表现

同学们课上听讲都比较认真, 普遍做题也做的还不错。

课堂内容

P10909 [蓝桥杯 2024 国 B] 立定跳远

二分, 判断当 $L = \text{mid}$ 的情况下, 能否满足只增加 $\leq m$ 个检查点就可以跳到终点

题目说的可以跳一次 $2L$ 的距离, 本质上等价于可以多增加一个检查点, 即: 判断能否在 $\leq m+1$ 个检查点的前提下到终点即可

从 $w[i]$ 位置跳到 $w[i+1]$ 位置, 需要增加检查点的数量: $w[i+1]-w[i]$ 除以 mid 的上取整 再 减一

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
int w[maxn];
int n, m;

int get_up(int a, int b) { return (a+b-1)/b; }

bool check(int mid) {
    int res = 0;
    for (int i = 0; i <= n-1; ++i) {
        res += get_up(w[i+1]-w[i], mid)-1;
        if (res > m) return false;
    }
    return true;
}

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

    int l = 1, r = 1e8+100;
    while (l <= r) {
        int mid = (l + r) / 2;
        if (check(mid)) r = mid-1;
        else l = mid+1;
    }

    cout << l << endl;
}
```

```
    return 0;
}
```

B4303 [蓝桥杯青少年组省赛 2024] 字母移位

考虑往左为正方向, 那么原问题本质上等价于:

- 第 1 个字符要变 $a_1 - a_2 + a_3 - a_4 + a_5 - a_6 + \dots$
- 第 2 个字符要变 $-a_2 + a_3 - a_4 + a_5 - a_6 + \dots$
- 第 3 个字符要变 $a_3 - a_4 + a_5 - a_6 + \dots$
- ...

所以, 可以维护一个数组, 数组的每一项分别为: $a_1, -a_2, a_3, -a_4, a_5, -a_6, \dots$

第 i 个字符要进行的变化: 其实就是第 i 项一直到第 n 项的后缀和, 因此, 可以维护一个后缀和数组来 $O(1)$ 求第 i 项字符的变化

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
char s[maxn];
int w[maxn], val[maxn];
LL suf[maxn];

char calc(char x, LL k) {
    k %= 26;
    if (k < 0) k += 26;

    x -= k;
    if (x < 'a') x += 26;
    return x;
}

int main()
{
    int n; cin >> n;
    cin >> (s+1);
    for (int i = 1; i <= n; ++i) {
        cin >> w[i];
        if (i&1) val[i] = w[i];
        else val[i] = -w[i];
    }

    for (int i = n; i >= 1; --i) suf[i] = suf[i+1] + val[i];

    for (int i = 1; i <= n; ++i) cout << calc(s[i], suf[i]);
}
```

```

    cout << endl;
    return 0;
}

```

U552394 买整数

比较直接的二分, 二分判断目前的 X 元能否买 mid 这个整数即可

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
LL a, b, c;

bool check(LL mid) {
    LL value = mid*a;
    int cnt = 0;
    while (mid) mid /= 10, cnt++;
    value += cnt*b;
    return value <= c;
}

int main()
{
    cin >> a >> b >> c;
    LL l = 1, r = 1000000000;
    while (l <= r) {
        LL mid = (l + r) / 2;
        if (check(mid)) l = mid+1;
        else r = mid-1;
    }
    cout << r << endl;
    return 0;
}

```

P3056 [USACO12NOV] Clumsy Cows S

用栈维护, 遍历字符串

- 遇到 '(', 将其入栈
- 遇到 ')', 看栈是否为空: 如果不空, 则删除一个左括号与其对应即可; 如果空, 则将该右括号改为左括号入栈

遍历结束, 如果栈里非空, 因为栈里面剩下的都是左括号, 则需要把一半的左括号改为右括号

```

#include <bits/stdc++.h>

using namespace std;

```

```
const int maxn = 1e5 + 5;
char s[maxn];

int main()
{
    cin >> (s+1);
    int n = strlen(s+1), res = 0;
    stack<char> stk;
    for (int i = 1; i <= n; ++i) {
        char x = s[i];
        if (x == '(') stk.push(x);
        else {
            if (!stk.empty()) stk.pop();
            else {
                stk.push('('), ++res;
            }
        }
    }
    res += stk.size()/2;
    cout << res << endl;
    return 0;
}
```