

欢乐的跳

人员

初锦阳、田心一、吴念远、董岱诚、赵牧之、周沁言、李知朔、李瑞涵、柳力玮、刘子轩、王馨琪、苑钊、韩昱辰、亓骏泽 到课

上周作业检查

上周作业链接: <https://www.luogu.com.cn/contest/217655>

2024-1201周日15:30

报名

编辑比赛

题目数4 | 报名人数17

比赛说明

题目列表

排行榜

名次	参赛者	总分	A	B	C	D
#1	柳力玮	400 (7.07h)	100 (54.05min)	100 (2.15h)	100 (2.79h)	100 (1.23h)
#2	宋云朗	400 (12.07d)	100 (2.98d)	100 (3.01d)	100 (3.04d)	100 (3.05d)
#3	田心一	400 (12.13d)	100 (1.36h)	100 (2.17h)	100 (5.98d)	100 (6.00d)
#4	李瑞涵	368 (9.61h)	100 (1.49h)	100 (2.29h)	68 (2.39h)	100 (3.45h)
#5	刘子轩	300 (5.29h)	100 (1.42h)	100 (2.19h)		100 (1.68h)
#6	赵牧之	300 (3.43d)	100 (1.45h)	100 (1.22d)		100 (2.15d)
#7	李知朔	290 (5.47d)	100 (1.55h)	100 (2.29h)	68 (2.46h)	22 (5.21d)
#8	初锦阳	290 (6.27d)	100 (1.68h)	100 (2.17h)	68 (2.40h)	22 (6.01d)
#9	吴念远	268 (5.93h)	100 (1.32h)	100 (2.23h)	68 (2.38h)	
#10	亓骏泽	268 (6.93d)	100 (1.67h)	100 (3.34h)	68 (6.72d)	
#11	周沁言	268 (12.54d)	100 (1.44h)	100 (6.24d)	68 (6.24d)	
#12	纪博涵	262 (8.85h)	100 (1.56h)	100 (2.21h)	62 (5.08h)	
#13	王馨琪	204 (5.33h)	100 (1.45h)	100 (2.23h)		4 (1.64h)
#14	Blue_wonders	100 (3.02d)			100 (3.02d)	
#15	董岱诚	0 (0ms)				

作业

<https://www.luogu.com.cn/contest/218964> (F 题选做)

课堂表现

有些同学课上着急自己做题, 在老师讲课时不集中注意力听。以后要进行改正, 一定以听讲为主。

课堂内容

U498481 字符串最大跨距

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

using namespace std;

int main()
{
    string ss;
    cin >> ss;
    int n = ss.size();
    int lpos, rpos;
    for (int i = 0; i < n; i++) {
        if (ss[i] == ',') {
            lpos = i;
            break;
        }
    }
    for (int i = n-1; i >= 0; i--) {
        if (ss[i] == ',') {
            rpos = i;
            break;
        }
    }

    string s = ss.substr(0, lpos);
    string s1 = ss.substr(lpos+1, rpos-lpos-1);
    string s2 = ss.substr(rpos+1);

    int l = s.find(s1), r = s.rfind(s2);
    if (l==-1 || r==-1) {
        cout << -1 << endl;
        return 0;
    }

    int t = r - l - s1.size();
    if (t > 0) {
        cout << t << endl;
    } else {
        cout << -1 << endl;
    }
    return 0;
}
```

U510530 珍珠链

-> 找所有稳定区间

-> 找所有区间，判断其是否稳定

```
int maxx = 0;
for (int i = 1; i <= n; i++) {
    for (int j = i; j <= n; j++) {
        // 计算 i~j 中有多少 G, R
        int sumg = 0, sumr = 0;
        for (int k = i; k <= j; k++) {
            if (s[k] == 'G') sumg++;
            else sumr++;
        }
        if (sumg == sumr) {
            maxx = max(maxx, j-i+1);
        }
    }
}
```

B2096 直方图

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

using namespace std;

const int maxn = 1e5 + 5;
int f[maxn];

int main()
{
    int n; cin >> n;
    int maxx = 0;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x; f[x]++; maxx = max(maxx, x);
    }

    for (int i = 0; i <= maxx; ++i) cout << f[i] << endl;
    return 0;
}
```

B2097 最长平台

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

using namespace std;

int a[105];
```

```

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

    int cnt = 1; // 第一个数出现了一次
    int maxx = 1;

    for (int i = 2; i <= n; i++) {
        if (a[i] == a[i-1]) {
            cnt++;
        } else {
            cnt = 1;
        }
        maxx = max(maxx, cnt);
    }

    cout << maxx << endl;
    return 0;
}

```

B2127 求正整数 2 和 n 之间的完全数

```

for (int i = 1; i <= n; i++) {
    if (i 是完全数) {
        cout << i << endl;
    }
}

i 的因数怎么找

for (int j = 1; j <= i-1; j++) {
    if (i%j == 0) {
        sum += j;
    }
}

if (sum == i)

```

P1152 欢乐的跳

```

// 方法一
a[1]      a[2]      a[3]      ....      a[n-1]      a[n]

```

```
b[1] = abs(a[1]-a[2])
b[2] = abs(a[2]-a[3])
...
b[n-1] = abs(a[n-1]-a[n])
```

构造 b 数组:

```
for (int i = 1; i <= n-1; i++) {
    b[i] = abs(a[i] - a[i+1]);
}
```

如果 $b[1] \sim b[n-1]$ 这里面正好 $1 \sim n-1$ 全都只出现一次的话

那么对 b 数组排序之后, 一定满足:

```
b[1] = 1
b[2] = 2
b[3] = 3
...
b[n-1] = n-1
```

// 方法二

```
a[1]          a[2]          a[3]          ....          a[n-1]          a[n]
```

```
abs(a[1]-a[2])
abs(a[2]-a[3])
...
abs(a[n-1]-a[n])
```

要确保上面这些值在 $1 \sim n-1$ 之间的每个数出现一次

```
int cnt[1005];
cnt[abs(a[1]-a[2])]++;
cnt[abs(a[2]-a[3])]++;
...
cnt[abs(a[n-1]-a[n])]++;
```

如果要满足题意, 必须要满足:

```
cnt[1] = 1
cnt[2] = 1
cnt[3] = 1
...
cnt[n-1] = 1
```

如果 $a[1]$ 是 100000000, $a[2]$ 是 0
但凡出现 $\text{abs}(a[i] - a[i+1])$ 一旦超过 $n-1$, 一定是错的

U498485 回文子串

```
string s;  
cin >> s;  
  
for (int len = 2; len <= n; len++) {  
    for (int i = 0; i <= n-len; i++) {  
        string ss = s.substr(i, len);  
        string t = ss;  
        reverse(t.begin(), t.end());  
        if (t == ss) {  
            }  
        }  
    }  
}
```