

# 字母移位

## 人员

初锦阳、赵牧之、王馨琪、刘宸熙、苑钊、温郝冬、倪炜艺、柳力玮、田心一、姜皓轩、谢梓轩、李知朔、韩昱辰、燕润石、李瑞涵 到课, 栾婷婷、刘子轩 线上

## 上周作业检查

上周作业链接: <https://www.luogu.com.cn/contest/240403>

2025-0406周日15:30

报名

编辑比赛

题目数6 | 报名人数21

比赛说明 | 题目列表 | 排行榜

名次	参赛者	总分	A	B	C	D	E	F
#1	柳力玮	600 (10.61h)	100 (700ms)	100 (374ms)	100 (126ms)	100 (1.45s)	100 (5.47h)	100 (5.15h)
#2	栾婷婷	600 (5.23d)	100 (643ms)	100 (293ms)	100 (124ms)	100 (1.55s)	100 (2.05h)	100 (5.14d)
#3	苑钊	600 (41.71d)	100 (6.87d)	100 (6.93d)	100 (6.95d)	100 (6.96d)	100 (6.99d)	100 (7.00d)
#4	赵牧之	530 (6.68d)	100 (657ms)	100 (293ms)	100 (5.05h)	100 (1.60s)	100 (5.56h)	30 (6.24d)
#5	田心一	530 (13.00d)	100 (655ms)	100 (316ms)	100 (125ms)	100 (1.34s)	100 (6.82d)	30 (6.17d)
#6	谢梓轩	500 (20.40d)	100 (653ms)	100 (5.43s)	100 (6.78d)	100 (6.79d)	100 (6.82d)	
#7	温郝冬	500 (25.77d)	100 (6.23d)	100 (286ms)	100 (6.23d)	100 (6.24d)	100 (7.07d)	
#8	燕润石	500 (34.85d)	100 (6.93d)		100 (6.95d)	100 (6.97d)	100 (6.99d)	100 (7.00d)
#9	初锦阳	410 (5.62h)	100 (656ms)	100 (306ms)	100 (123ms)	100 (1.45s)		10 (5.62h)
#10	纪博涵	400 (2.58s)	100 (652ms)	100 (311ms)	100 (122ms)	100 (1.50s)		
#11	李知朔	400 (6.23d)	100 (662ms)	100 (310ms)	100 (134ms)	100 (6.23d)		
#12	刘宸熙	400 (9.21d)	100 (659ms)	100 (317ms)	100 (6.98d)	100 (2.23d)		
#13	王馨琪	400 (19.20d)	100 (655ms)	100 (6.16d)	100 (6.19d)	100 (6.85d)		
#14	姜皓轩	274 (7.32d)	100 (655ms)	100 (6.34h)	14 (1.23d)			60 (5.82d)
#15	倪炜艺	210 (793ms)	100 (655ms)	10 (0ms)	100 (138ms)			
#16	韩昱辰	200 (782ms)	100 (658ms)		100 (124ms)			
#17	刘梓勋	110 (14.34d)	100 (7.07d)	10 (7.27d)				
#18	刘子轩	100 (643ms)	100 (643ms)					
#19	Blue_wonders	100 (7.00d)						100 (7.00d)
#20	李瑞涵	72 (6.93d)	72 (6.93d)					

## 作业

<https://www.luogu.com.cn/contest/241023> (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

同学们上课听讲都比较认真, 不过有一部分同学课上很多题目都没有完成, 课后要把课上没做的题目都补一补。

## 课堂内容

### T504032 克隆机 (clone)

先是 1 遍每个字符, 再是 2 遍每个字符, 再 4 遍, 再 8 遍, 16 遍, .....

嵌套循环维护即可, 要维护已经遍历过多少字母了, 看什么时候遍历能超过第 n 个字符即可

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

using namespace std;

typedef long long LL;

int main()
{
    int k; LL n; cin >> k >> n;
    LL len = 0;
    for (LL i = 1; ; i *= 2) {
        for (char j = 'A'; j <= 'A'+k-1; ++j) {
            LL len2 = len + i;
            if (n <= len2) {
                cout << j << endl;
                return 0;
            }
            len = len2;
        }
    }
    return 0;
}
```

### U552396 找整数

进制转换, 把 1 换成 2 即可

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

using namespace std;

typedef long long LL;

int main()
{
    LL n; cin >> n;
```

```

string s;
while (n) {
    int t = n%2; n /= 2;
    if (t == 0) s += "0";
    else s += "2";
}
reverse(s.begin(), s.end());
cout << s << 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;
}

```

### B4038 [GESP202409 三级] 平衡序列

枚举  $i$ , 看是否有一个  $1 \sim i$  的区间和是否等于  $i+1 \sim n$  的区间和即可

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 10000 + 5;
int w[maxn], p[maxn];

int get_sum(int l, int r) { return p[r] - p[l-1]; }

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

    for (int i = 1; i <= n-1; ++i) {
        if (get_sum(1,i) == get_sum(i+1,n)) {
            cout << "Yes" << endl; return;
        }
    }
    cout << "No" << endl;
}

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

```