

# 综合混练

## 人员

牛晓晨 到课

## 上周作业检查

上周作业链接: <https://vjudge.net/contest/715752>

Begin: 2025-05-10 08:30 CST

☆👥 2025-0510 五队上课 (综合混练)

End: 2025-12-04 16:30 CST

Elapsed: 6:23:29:18

Running

Remaining: 201:08:30:41

OverviewProblemStatusRank (6:23:29:18)DiscussSettingCloneUpdateDelete

Rank	Team	Score	Penalty	A 18 / 19	B 16 / 18	C 9 / 21	D 2 / 21	E 2 / 4	F 0 / 3	G 1 / 6
1	☆👤 ikunTLE (方冠霖)	6	31963	7:05:25	8:18:07	2:12:55:50	6:06:30:04 (-6)	6:06:54:21 (-2)		6:08:20:08
2	☆👤 fbl123bc (付丙森)	4	5102	6:49:46	7:56:11	8:50:28 (-2)	(-6)	2:12:45:43	(-2)	
3	☆👤 ccx123bc (曹承贤)	4	19600	7:15:26	8:02:18	6:09:10:01 (-1)	6:13:52:27			
4	☆👤 niuxiaochen (牛晓晨)	3	449	0:44:22	1:33:25	5:11:16				
5	☆👤 zhn123bc (张皓宇)	3	1369	6:52:17	7:26:38	8:30:10	(-3)		(-1)	
6	☆👤 misaka16384 (黄诗琦)	3	1546	7:10:41	7:45:49	8:50:19 (-6)				
7	☆👤 Hanhj (韩鸿钜)	3	1782	7:36:55	8:14:56	13:50:17				
8	☆👤 WangYanzhen (王彦臻)	3	3176	6:55:54	8:39:41 (-1)	1:13:00:25				
9	☆👤 qp_an (赵广宇)	3	7718	7:07:58	2:12:36:49	2:12:53:59				
10	☆👤 two_tiger (卢炫佑)	2	909	7:03:18	8:06:33	(-2)	(-1)			
11	☆👤 chx123bc (陈瀚霄)	2	926	7:21:30	8:05:14		(-3)			(-5)
12	☆👤 lzy1031 (李政毅)	2	940	7:16:17	8:24:05					
13	☆👤 dana230513 (金一航)	2	954	7:36:09	8:18:08					
14	☆👤 lzy123bc (刘智予)	2	961	7:41:25	8:19:46					
15	☆👤 FeatherCrow (许岩)	2	972	7:37:59 (-1)	8:15:00	(-1)				
16	☆👤 longlong_int (刘锦轩)	2	2155	0:47:52	1:11:07:57					
17	☆👤 lxr123bc (刘新睿)	1	45	0:45:52						
18	☆👤 fj123bc (范家郡)	1	50	0:50:48	(-1)					

## 作业

<https://vjudge.net/contest/717355> (课上讲了上周比赛的 D E F G, 本周比赛的 A B 这些题, 课后作业是本周比赛的 C D E F 题)

## 课堂表现

课堂表现很不错, 听课很认真

## 课堂内容

### CF1886C Decreasing String

可以用枚举或二分确定最后一段是删了几个字符, 然后用单调栈删掉这些数量的字符即可

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

using namespace std;

typedef long long LL;

LL get_sum(int l, int r) { return ((LL)l+r)*(r-l+1)/2; }

bool check(int n, int mid, LL pos) {
    LL sum = get_sum(n-mid+1, n);
    return pos > sum;
}

string get_str(string s, int k) {
    vector<char> vec;
    for (char i : s) {
        while (!vec.empty() && i < vec.back() && k) vec.pop_back(), --k;
        vec.push_back(i);
    }
    while (k) vec.pop_back(), --k;

    string res;
    for (char i : vec) res += i;
    return res;
}

void solve() {
    string s; LL pos; cin >> s >> pos;
    int n = s.size();

    int l = 0, r = n-1;
    while (l <= r) {
        int mid = (l + r) / 2;
        if (check(n, mid, pos)) l = mid+1;
        else r = mid-1;
    }

    LL sum = get_sum(n-r+1, n);
    pos -= sum;

    string s2 = get_str(s, r);
```

```
    cout << s2[pos-1];
}

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

## CF1883E Look Back

先求每个数跟前一个数的差距, 差几次 乘2 或者 除2 的操作, 然后从前往后做累加

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

using namespace std;

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

int calc(int a, int b) {
    int res = 0;
    while (a*2 <= b) a *= 2, --res;
    while (a > b) b *= 2, ++res;
    return res;
}

void solve() {
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i];
    for (int i = 2; i <= n; ++i) h[i] = calc(w[i-1], w[i]);

    LL t = 0, res = 0;
    for (int i = 2; i <= n; ++i) {
        t += h[i];
        if (t < 0) t = 0;
        res += t;
    }

    // cout << "----- ";
    cout << res << endl;
}

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

```
    return 0;
}
```

## CF1280A Cut and Paste

模拟, 维护  $n$  以内的字符串,  $n$  后面的字符不需要具体维护, 只维护长度即可

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

using namespace std;

typedef long long LL;
const int mod = 1e9 + 7;

void solve() {
    int n; string s; cin >> n >> s;
    s = " " + s;

    int len = (int)s.size() - 1;
    for (int i = 1; i <= n; ++i) {
        int x = s[i] - '0' - 1;
        string t;
        if (x >= 1 && (int)s.size() < n + 10) t = s.substr(i + 1);
        for (int j = 0; j < x; ++j) s += t;

        int t_len = (len - i + mod) % mod;
        len = (len + t_len * x % mod) % mod;
    }

    // cout << "----- ";
    cout << len << endl;
}

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

## CF14E Camels

dp,  $f(i, j, k, x, y)$ : 第  $i$  位是  $j$ ,  $i-1$  位是  $k$ , 有  $x$  个高峰,  $y$  个低峰时的方案数

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

using namespace std;

const int N = 20 + 5, M = 10 + 5;
```

```

int f[N][5][5][M][M];
// f(i,j,k,x,y): 第i位是j, i-1位是k, 有x个高峰,y个低峰时 的方案数

int main()
{
    int n, t; cin >> n >> t;
    for (int i = 1; i <= 4; ++i) {
        for (int j = 1; j <= 4; ++j) {
            if (i == j) continue;
            f[2][i][j][0][0] = 1;
        }
    }

    for (int i = 3; i <= n; ++i) {
        for (int j = 1; j <= 4; ++j) {
            for (int k = 1; k <= 4; ++k) {
                if (j == k) continue;
                for (int x = 0; x <= t; ++x) {
                    for (int y = 0; y <= t-1; ++y) {
                        // f[i][j][k][x][y] <- f[i-1][k][l][?][?]
                        // l, k, j
                        for (int l = 1; l <= 4; ++l) {
                            if (k == l) continue;
                            if (l > k && k < j) {
                                if (y >= 1) f[i][j][k][x][y] += f[i-1][k][l][x][y-1];
                            } else if (l < k && k > j) {
                                if (x >= 1) f[i][j][k][x][y] += f[i-1][k][l][x-1][y];
                            } else {
                                f[i][j][k][x][y] += f[i-1][k][l][x][y];
                            }
                        }
                    }
                }
            }
        }
    }

    int res = 0;
    for (int i = 1; i <= 4; ++i) {
        for (int j = 1; j <= 4; ++j) res += f[n][i][j][t][t-1];
    }
    cout << res << endl;
    return 0;
}

```

### CF1485D Multiples and Power Differences

用  $720720$  和  $720720+a[i][j]^4$  来交替构造这个二维数组即可

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

```

using namespace std;

const int maxn = 500 + 5;
int w[maxn][maxn];

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

    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            if ((i+j)&1) cout << 720720 << " ";
            else cout << 720720 - w[i][j]*w[i][j]*w[i][j]*w[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}

```

### CF1473D Program

输入 [l,r] 后, 可以查询 [1,l-1] 和 [r+1,n] 的最大值、最小值, 然后处理

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;
char s[maxn];
int _log2[maxn], p[maxn], f1[maxn][20], f2[maxn][20];
int n, m;

int get_max(int l, int r) {
    int len = r - l + 1;
    int k = _log2[len];
    return max(f1[l][k], f1[r-(1<<k)+1][k]);
}

int get_min(int l, int r) {
    int len = r - l + 1;
    int k = _log2[len];
    return min(f2[l][k], f2[r-(1<<k)+1][k]);
}

int calc(int l, int r) {
    int up = 0, down = 0;
    if (l-1 >= 1) up = max(up, get_max(1, l-1)), down = min(down, get_min(1, l-1));
}

```

```
    if (r+1 <= n) {
        up = max(up, get_max(r+1, n) - p[r] + p[l-1]);
        down = min(down, get_min(r+1, n) - p[r] + p[l-1]);
    }
    return up - down + 1;
}

void solve() {
    cin >> n >> m;
    cin >> (s+1);
    for (int i = 1; i <= n; ++i) {
        if (s[i] == '+') p[i] = p[i-1] + 1;
        else p[i] = p[i-1] - 1;
        f1[i][0] = f2[i][0] = p[i];
    }

    for (int k = 1; k < 20; ++k) {
        for (int i = 1; i+(1<<k)-1 <= n; ++i) {
            f1[i][k] = max(f1[i][k-1], f1[i+(1<<(k-1))][k-1]);
            f2[i][k] = min(f2[i][k-1], f2[i+(1<<(k-1))][k-1]);
        }
    }

    while (m -- ) {
        int l, r; cin >> l >> r;
        // cout << "----- ";
        cout << calc(l, r) << endl;
    }
}

int main()
{
    for (int i = 0; (1<<i)<maxn; ++i) _log2[1<<i] = i;
    for (int i = 1; i < maxn; ++i) {
        if (_log2[i]) continue;
        _log2[i] = _log2[i-1];
    }

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