

# 综合练习

## 人员

赵熙羽、杨瑾硕、谢亚锴、陈洛冉、于子珈、隋天乙、司云心、刘闯速、牛同泽、程梓豪、秦显森、孙靖轲、葛真然 到课

## 上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/754>

### 2025-0907 周日10:30 (综合练习)

🔄 刷新

#	用户名	姓名	编程分	时间	A	B	C	D	E
1	qinxiansen	秦显森	500	872	100	100	100	100	100
2	yuzijia1	于子珈	500	1053	100	100	100	100	100
3	xieyakai	谢亚锴	500	1059	100	100	100	100	100
4	sunjingke	孙靖轲	500	1139	100	100	100	100	100
5	siyunxin	司云心	500	1443	100	100	100	100	100
6	zhaoxiyu	赵熙羽	500	1708	100	100	100	100	100
7	chengzihao	程梓豪	400	817	100	100		100	100
8	yangjinsuo	杨瑾硕	400	1289	100	100	100	100	
9	chenluoran	陈洛冉	300	1001	100	100	100		
10	liuchuangsu	刘闯速	260	449	100		100	60	
11	niutongze	牛同泽	200	343	100	100			0
12	sultianvi	隋天乙	197	100	100	0	37	60	

## 本周作业

<https://cppoj.kids123code.com/contest/793> (课上讲了 A ~ C 题, 课后作业是 D 题)

## 课堂表现

今天课上的前两题主要是一些 STL 嵌套的题目, 可以看出有一部分同学掌握的还是不太好的

主要有 pair不会用、结构体不会重载运算符、不会遍历map 等这几个问题

课上老师带大家全部过了一遍这些 STL 使用的方法, 课下同学们也要自己再好好复习复习

## 课堂内容

### [CSP-J 2024] 地图探险 (上周作业)

按照题意模拟即可, 循环 k 次, 能往前走就往前走, 不能往前走就转向

把到过的点打标记, 最后统计到过多少点即可

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

```

using namespace std;

const int maxn = 1000 + 5;
char s[maxn][maxn];
bool st[maxn][maxn];
int dx[] = {0, 1, 0, -1}, dy[] = {1, 0, -1, 0};

void solve() {
    memset(st, false, sizeof(st));

    int n, m, k; cin >> n >> m >> k;
    int x, y, id; cin >> x >> y >> id;

    for (int i = 1; i <= n; ++i) cin >> (s[i]+1);

    st[x][y] = true;
    for (int i = 1; i <= k; ++i) {
        int nx = x+dx[id], ny = y+dy[id];
        if (nx>=1 && nx<=n && ny>=1 && ny<=m && s[nx][ny]=='.') x = nx, y = ny, st[x]
[y] = true;
        else id = (id+1)%4;
    }

    int cnt = 0;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            if (s[i][j]=='.' && st[i][j]) ++cnt;
        }
    }
    cout << cnt << endl;
}

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

```

## Dash

set 套 pair 记录哪些位置有可以恢复生命的物品, 然后按照题意模拟走的过程即可

```

#include <bits/stdc++.h>

using namespace std;

set<pair<int,int>>>s;

int main()
{

```

```

int n, m, h, k; cin >> n >> m >> h >> k;
string str; cin >> str;
while (m -- ) { int xx, yy; cin >> xx >> yy; s.insert({xx,yy}); }

int x = 0, y = 0;
for (char i : str) {
    if (i == 'R') ++x;
    else if (i == 'L') --x;
    else if (i == 'U') ++y;
    else --y;

    --h;
    if (h < 0) { cout << "No" << endl; return 0; }
    if (s.count({x,y}) && h<k) { h = k; s.erase({x,y}); }
}
cout << "Yes" << endl;
return 0;
}

```

## A Piece of Cake

map 套 pair 维护

判断一个草莓在哪一块蛋糕上, 这个过程可以用 二分查找  $\log n$  来实现, 然后用 map 套 pair 记录这块蛋糕上草莓数量 +1

```

#include <bits/stdc++.h>
#define x first
#define y second

using namespace std;

typedef pair<int,int> PII;
const int maxn = 2e5 + 5;
int a[maxn], b[maxn];
struct node {
    int x, y;
} w[maxn];

int main()
{
    int W, H; cin >> W >> H;
    int T; cin >> T;
    for (int i = 1; i <= T; ++i) cin >> w[i].x >> w[i].y;

    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> a[i];
    int m; cin >> m;
    for (int i = 1; i <= m; ++i) cin >> b[i];

    map<PII, int> mp;

```

```

for (int i = 1; i <= T; ++i) {
    int x = w[i].x, y = w[i].y;
    int tx = lower_bound(a+1, a+n+1, x) - a, ty = lower_bound(b+1, b+m+1, y) - b;
    mp[{tx,ty}]++;
}

int minn = 1e9, maxx = 0;
if (mp.size() != (n+1)*(m+1)) minn = 0;
for (auto it : mp) minn = min(minn, it.y), maxx = max(maxx, it.y);
cout << minn << " " << maxx << endl;
return 0;
}

```

## NAND repeatedly

线性 dp

$f[i][0]$  代表以第  $i$  项结尾时有多少 0,  $f[i][1]$  代表以第  $i$  项结尾时有多少 1

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 1e6 + 5;
char s[maxn];
LL f[maxn][2];

int main()
{
    int n; cin >> n >> (s+1);

    LL res = 0;
    for (int i = 1; i <= n; ++i) {
        if (s[i] == '0') {
            f[i][0] = 1, f[i][1] = f[i-1][0] + f[i-1][1];
        } else {
            f[i][0] = f[i-1][1], f[i][1] = f[i-1][0]+1;
        }
        res += f[i][1];
    }
    cout << res << endl;
    return 0;
}

```