2026年8月 SACCC 三级题解

苏州市人工智能学会 · 青少年人工智能核心算法素养考核（SACCC）

级别：三级 期次：2026年8月

题目清单

顺序	题目	主要考点
1	星灯编号统计	进制转换、累加
2	二进制灯牌开关	位运算、顺序模拟
3	训练温度记录	一维数组、连续上升段、最大值位置
4	签到码校验	字符串遍历、字符分类、字符转换
5	纪念品三件套	枚举、三数求和计数
6	巡展机器人	网格模拟、指令处理

第 1 题 星灯编号统计

主要考点： 进制转换、累加

题意概括

输入若干个用二进制、八进制、十进制或十六进制表示的非负整数。需要把它们统一转换成十进制，并输出所有数的总和。

核心思路

从高位到低位扫描每个编号。若当前已经得到前缀的十进制值 value，读入下一位数字 d 后，新值就是 value * base + d。十六进制中的字母需要先转换成 10 到 15，得到每个编号的值后累加即可。

参考代码

```
#include <bits/stdc++.h>
using namespace std;

int digitValue(char c) {
    if (c >= '0' && c <= '9') return c - '0';
    if (c >= 'A' && c <= 'F') return c - 'A' + 10;
    return c - 'a' + 10;
}

long long convertValue(const string &s, int base) {
    long long value = 0;
    // 按位使用“旧值乘进制再加当前位”的方式完成转换。
    for (char c : s) value = value * base + digitValue(c);
    return value;
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int n;
    cin >> n;
    long long sum = 0;
    for (int i = 0; i < n; ++i) {
        int base;
        string code;
        cin >> base >> code;
        sum += convertValue(code, base);
    }
    cout << sum << '\n';
    return 0;
}
```

第 2 题 二进制灯牌开关

主要考点： 位运算、顺序模拟

题意概括

给定灯牌的初始整数状态以及若干条位运算指令，要求严格按照输入顺序执行全部指令，并输出最终状态。

核心思路

用变量 x 保存当前状态。逐条读取操作名和参数，根据操作名执行对应的按位与、按位或、按位异或、左移或右移，并把结果继续保存在 x 中。后面的操作必须使用前一次更新后的值。

参考代码

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int n;
    long long x;
    cin >> n >> x;
    for (int i = 0; i < n; ++i) {
        string op;
        long long v;
        cin >> op >> v;
        // 每次都在最新的 x 上执行操作，不能打乱指令顺序。
        if (op == "AND") x &= v;
        else if (op == "OR") x |= v;
        else if (op == "XOR") x ^= v;
        else if (op == "SHL") x <<= v;
        else if (op == "SHR") x >>= v;
    }
    cout << x << '\n';
    return 0;
}
```

第 3 题 训练温度记录

主要考点： 一维数组、连续上升段、最大值位置

题意概括

给定连续若干天的温度，要求同时求出最长的严格连续升温天数，以及最高温第一次出现的位置和温度值。

核心思路

一次扫描即可完成两项统计。若当天温度高于前一天，就把当前连续升温段长度加一，否则重新从 1 开始；同时维护历史最长长度。最高温位置只在遇到更大温度时更新，因此温度相同时会自然保留第一次出现的位置。

参考代码

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int n;
    cin >> n;
    vector<int> a(n);
    for (int i = 0; i < n; ++i) cin >> a[i];
    int best = 1, cur = 1, maxPos = 0;
    for (int i = 1; i < n; ++i) {
        // cur 表示以第 i 天结尾的严格连续升温段长度。
        if (a[i] > a[i - 1]) ++cur;
        else cur = 1;
        best = max(best, cur);
        // 只在严格变大时更新，从而保留最高温第一次出现的位置。
        if (a[i] > a[maxPos]) maxPos = i;
    }
    cout << best << ' ' << maxPos + 1 << ' ' << a[maxPos] << '\n';
    return 0;
}
```

第 4 题 签到码校验

主要考点： 字符串遍历、字符分类、字符转换

题意概括

统计签到码中大写字母、小写字母和数字的数量，并按规定生成新字符串：字母变成下一个字母，数字 d 变成 9-d，其他字符不变。

核心思路

逐个字符扫描原字符串，一边判断字符类别并计数，一边调用转换函数生成新字符。字母转换时用取模处理 z 和 Z 回到 a 和 A；数字可以用 9-(d-0) 得到对应的新数字。

参考代码

```
#include <bits/stdc++.h>
using namespace std;

char changeChar(char c) {
    // 取模让 z、Z 分别回到 a、A；数字按 9-d 转换。
    if (c >= 'a' && c <= 'z') return char('a' + (c - 'a' + 1) % 26);
    if (c >= 'A' && c <= 'Z') return char('A' + (c - 'A' + 1) % 26);
    if (c >= '0' && c <= '9') return char('0' + (9 - (c - '0')));
    return c;
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    string s;
    getline(cin, s);
    int upper = 0, lower = 0, digit = 0;
    string changed;
    for (char c : s) {
        if (isupper((unsigned char)c)) ++upper;
        if (islower((unsigned char)c)) ++lower;
        if (isdigit((unsigned char)c)) ++digit;
        changed.push_back(changeChar(c));
    }
    cout << upper << ' ' << lower << ' ' << digit << '\n' << changed << '\n';
    return 0;
}
```

第 5 题 纪念品三件套

主要考点： 枚举、三数求和计数

题意概括

从 n 件纪念品中恰好选择 3 件，每件最多选择一次。不同下标视为不同纪念品，需要统计总价恰好等于预算的选择方案数。

核心思路

数据范围允许直接枚举三个下标。令 i < j < k，就能保证三件纪念品互不重复，并且每一种选择只统计一次；当三个价格之和等于预算时答案加一。

参考代码

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int n, target;
    cin >> n >> target;
    vector<int> a(n);
    for (int i = 0; i < n; ++i) cin >> a[i];
    int ans = 0;
    for (int i = 0; i < n; ++i)
        for (int j = i + 1; j < n; ++j)
            for (int k = j + 1; k < n; ++k)
                // i < j < k 保证不重复选取，也不会重复统计同一组合。
                if (a[i] + a[j] + a[k] == target) ++ans;
    cout << ans << '\n';
    return 0;
}
```

第 6 题 巡展机器人

主要考点： 网格模拟、指令处理

题意概括

机器人从给定位置出发，按照指令在网格中移动。越界或撞到障碍时本次移动无效；到达带星标的格子时收集星标，最后输出位置和收集数量。

核心思路

为四种指令建立行、列变化量，然后逐条模拟。每次先计算候选位置，只有候选位置在网格内且不是障碍时才真正移动。星标收集后立即把该格改为空地，避免机器人再次经过时重复计数；起点也要先检查一次。

参考代码

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int n, m, r, c, q;
    cin >> n >> m >> r >> c >> q;
    string cmd;
    cin >> cmd;
    vector<string> grid(n);
    for (int i = 0; i < n; ++i) cin >> grid[i];
    --r; --c;
    int coins = 0;
    // 起点上的星标也算被经过，需要先收集。
    if (grid[r][c] == '*') {
        ++coins;
        grid[r][c] = '.';
    }
    map<char, pair<int, int>> dir = {{'U', {-1, 0}}, {'D', {1, 0}}, {'L', {0, -1}}, {'R', {0, 1}}};
    for (char ch : cmd) {
        int nr = r + dir[ch].first, nc = c + dir[ch].second;
        // 越界或遇到障碍时忽略本条指令，当前位置不变。
        if (nr < 0 || nr >= n || nc < 0 || nc >= m || grid[nr][nc] == '#') continue;
        r = nr;
        c = nc;
        if (grid[r][c] == '*') {
            ++coins;
            // 收集后清空，防止重复经过时再次计数。
            grid[r][c] = '.';
        }
    }
    cout << r + 1 << ' ' << c + 1 << ' ' << coins << '\n';
    return 0;
}
```