



第一题 算力 (power)

考察循环结构与分支结构。根据题意，枚举所有模型并统计符合条件的数量即可。时间复杂度 $O(n)$ 。

参考代码：

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

int n = 0, l = 0, r = 0, ans = 0;
int h[200005] = {};

int main() {
    cin >> n;
    for (int i = 1; i <= n; ++i) {
        cin >> h[i];
    }
    cin >> l >> r;
    for (int i = 1; i <= n; ++i) {
        if (l <= h[i] && h[i] <= r) {
            ++ans;
        }
    }
    cout << ans << endl;
    return 0;
}
```

第二题 数字 (digits)

考察数学知识与循环结构。用取模运算取出数字的个位数、用除法运算将数字除以十，不断重复即可。时间复杂度 $O(\log_{10} n)$ 。

参考代码：

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

int n = 0;

int main() {
    cin >> n;
    while (n > 0) {
        cout << n % 10;
        n /= 10;
    }
    cout << endl;
    return 0;
}
```

第三题 数独 (sudoku)

考察函数的定义与程序设计能力。根据题意判断每行、每列、每个九宫格中的数字是否符合条件即可。

参考代码：

```

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

char s[16] = "";
int a[10][10] = {};
bool vis[10] = {};

bool checkRow(int i) {
    fill(vis + 1, vis + 10, false);
    for (int j = 1; j <= 9; ++j) {
        if (vis[a[i][j]]) return false;
        vis[a[i][j]] = true;
    }
    return true;
}

bool checkCol(int j) {
    fill(vis + 1, vis + 10, false);
    for (int i = 1; i <= 9; ++i) {
        if (vis[a[i][j]]) return false;
        vis[a[i][j]] = true;
    }
    return true;
}

bool checkGrid(int i, int j) {
    fill(vis + 1, vis + 10, false);
    for (int x = 0; x <= 2; ++x) {
        for (int y = 0; y <= 2; ++y) {
            if (vis[a[i + x][j + y]]) return false;
            vis[a[i + x][j + y]] = true;
        }
    }
    return true;
}

bool check() {
    for (int i = 1; i <= 9; ++i) {
        if (!checkRow(i)) return false;
    }
}

```

```

}
for (int j = 1; j <= 9; ++j) {
    if (!checkCol(j)) return false;
}
for (int i = 1; i <= 7; i += 3) {
    for (int j = 1; j <= 7; j += 3) {
        if (!checkGrid(i, j)) return false;
    }
}
return true;
}

int main() {
    for (int i = 1; i <= 9; ++i) {
        cin >> s;
        for (int j = 1; j <= 9; ++j) {
            a[i][j] = (int)(s[j - 1] - '0');
        }
    }
    cout << (check() ? "Yes" : "No") << endl;
    return 0;
}

```

第四题 付费 (pay)

考察枚举法。本题的数据范围较小，直接穷举三种套餐的购买数量，并通过比较得到最优解即可。时间复杂度 $O(n^3)$ 。

参考代码：

```

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

int n = 0, a = 0, b = 0, c = 0, ans = 1e9;

int main() {
    cin >> n >> a >> b >> c;
    for (int i = 0; i <= n / 3 + 1; ++i) {
        for (int j = 0; j <= n / 5 + 1; ++j) {
            for (int k = 0; k <= n / 10 + 1; ++k) {
                if (i * 3 + j * 5 + k * 10 >= n) {
                    ans = min(ans, i * a + j * b + k * c);
                }
            }
        }
    }
    cout << ans << endl;
    return 0;
}

```

第五题 分割 (divide)

考察模拟法。如果一个数字可以通过一定的操作变成 1，则一定是先尝试除以 2，再尝试乘以 $\frac{2}{3}$ ，最后尝试乘以 $\frac{4}{5}$ 。模拟这个过程即可，时间复杂度 $O(\log x)$ 。

参考代码：

```

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

int main() {
    int T = 0;
    cin >> T;
    while (T--) {
        long long x = 0;
        int ans = 0;
        cin >> x;
        while (x != 1) {
            if (x % 2 == 0) {
                x = x / 2, ++ans;
            } else if (x % 3 == 0) {
                x = x / 3 * 2, ++ans;
            } else if (x % 5 == 0) {
                x = x / 5 * 4, ++ans;
            } else {
                ans = -1;
                break;
            }
        }
        cout << ans << endl;
    }
    return 0;
}

```

第六题 硬币 (coins)

考察枚举法与程序设计能力。用二进制集合枚举所有可能的真假组合，统计符合条件的数量即可。时间复杂度 $O(2^n \times m)$ 。

参考代码：

```

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

int n = 0, m = 0, k = 0, ans = 0;
int c[102] = {};
int a[102][16] = {};
bool flag[102] = {};

bool check(int s) {
    for (int i = 1; i <= m; ++i) {
        int cnt = 0;
        for (int j = 1; j <= c[i]; ++j) {
            if (s & (1 << (a[i][j] - 1))) ++cnt;
        }
        bool res = cnt >= k;
        if (res != flag[i]) return false;
    }
    return true;
}

int main() {
    cin >> n >> m >> k;
    for (int i = 1; i <= m; ++i) {
        char ch = 0;
        cin >> c[i];
        for (int j = 1; j <= c[i]; ++j) {
            cin >> a[i][j];
        }
        cin >> ch;
        flag[i] = ch == 'x';
    }
    for (int s = 0; s < (1 << n); ++s) {
        if (check(s)) ++ans;
    }
    cout << ans << endl;
    return 0;
}

```