

# C++方向编程题答案

## 第八周

### day43

#### 844 电话号码

链接: <https://www.nowcoder.com/questionTerminal/ceb89f19187b4de3997d9cdef2d551e8>

#### 【题目解析】：

这个题目比较简单. 借助 hash 表完成字母和数字之间的转换即可. 注意大小写的情况

#### 【解题思路】：

1. 先用hash表存储字母和数字之间的映射关系
2. 每次读到一个字符, 去hash表中查找, 并进行处理即可.

// 借助 hash 表完成字母和数字之间的转换即可. 注意处理大小写的情况.

```
#include<iostream>
#include<vector>
#include<string>
#include<algorithm>
#include<functional>
#include <map>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <exception>
#include <iomanip>
#include <memory>
#include <sstream>

#define INF 1000000
using namespace std;

int main(int argc, char** argv)
{
    unordered_map<char, char> dic;
    dic.emplace('A', '2');
    dic.emplace('B', '2');
    dic.emplace('C', '2');
    dic.emplace('D', '3');
    dic.emplace('E', '3');
    dic.emplace('F', '3');
    dic.emplace('G', '4');
    dic.emplace('H', '4');
    dic.emplace('I', '4');
    dic.emplace('J', '5');
    dic.emplace('K', '5');
    dic.emplace('L', '5');
```

```

dic.emplace('M', '6');
dic.emplace('N', '6');
dic.emplace('O', '6');
dic.emplace('P', '7');
dic.emplace('Q', '7');
dic.emplace('R', '7');
dic.emplace('S', '7');
dic.emplace('T', '8');
dic.emplace('U', '8');
dic.emplace('V', '8');
dic.emplace('W', '9');
dic.emplace('X', '9');
dic.emplace('Y', '9');
dic.emplace('Z', '9');
int n;
while (cin >> n && n > 0)
{
    set<string> hs;
    string s;
    for (int i = 0; i < n; ++i)
    {
        cin >> s;
        string p = "";
        for (auto& c : s)
        {
            if (isdigit(c)) p += c;
            else if (isupper(c)) p += dic[c];
        }
        if (p.size() != 7) continue;
        p = p.substr(0, 3) + "-" + p.substr(3);
        hs.emplace(p);
    }

    for (auto& phone : hs) cout << phone << endl;
    cout << endl;
}

return 0;
}

```

## 69391 求和

链接: <https://www.nowcoder.com/questionTerminal/11cc498832db489786f8a03c3b67d02c>

### 【题目解析】：

基于递归实现 dfs(深度优先搜索) 即可. 这是一个比较典型的背包问题

### 【解题思路】：

假设问题的解为 $F(n, m)$ , 可分解为两个子问题  $F(n-1, m-n)$ 和 $F(n-1, m)$ 。对这两个问题递归求解, 求解过程中, 如果找到了符合条件的数字组合, 则打印出来 例如 1, 2, 3, 4, 5, 求有多少中组合和为 5 对于 1 这个元素来说, 可能会放到结果中, 可能不放到结果中 如果放到结果中, 就相当于求 2...5 中取若干个数字和为 4. (即为  $F(n-1, m-n)$ ) 如果不放到结果中, 就相当于求 2...5 中取若干个数字和为 5. (即为 $F(n-1, m)$ )

```

#include<iostream>
#include<vector>
using namespace std;
void help(int n, int m, vector<int>& v, int beg) {
    // m == 0 为递归结束条件. 此时 v 中可能已经包含了若干个元素了. 并且 v 中的内容就是一组结果.
    if (m == 0) {
        for (int i = 0; i < v.size(); i++) {
            // 这个 ? : 只是为了让结果的格式能够和要求一样.
            i == 0 ? cout << v[i] : cout << " " << v[i];
        }
        cout << endl;
    }
    for (int i = beg; i <= n && i <= m; i++) {
        // 以下几行代码是该题目的关键. 问题的转换.
        // 为了求 i -> n 有多少种情况和为 m, 则把问题转换为
        // i + 1 -> n 有多少种情况和为 m - i
        v.push_back(i);
        help(n, m - i, v, i + 1);
        v.pop_back();
    }
}
int main() {
    int n, m;
    while (cin >> n >> m) {
        vector<int> v;
        help(n, m, v, 1);
    }
}

```