

C++方向编程题答案

第八周

day44

846 骆驼命名法

链接: <https://www.nowcoder.com/questionTerminal/aed1c7bbc2604e7c9661a2348b0541b8?pos=46&mutiTagIds=579&orderByHotValue=1>

【题目解析】:

题目描述比较好理解, 仔细读题即可.

【解题思路】:

首先一个字符一个字符的读取内容

1. 遇到 _ 就直接跳过
2. 如果上一个字符是 _ 则下一个字符转大写字母

```
#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)
{
    string s;
    while (cin >> s)
    {
        for (int i = 0; i < s.size(); ++i)
        {
            // 1. 遇到 _ 就直接跳过
            if (s[i] == '_') continue;
            // 2. 如果上一个字符是 _ 则下一个字符转为大写字母
            if (i > 0 && s[i - 1] == '_') cout << (char)toupper(s[i]);
            else cout << s[i];
        }
    }
}
```

```
    cout << endl;
}

return 0;
}
```

36855 单词倒排

链接: <https://www.nowcoder.com/questionTerminal/81544a4989df4109b33c2d65037c5836>

【题目解析】:

这个问题是包含了字符串常见操作的 切分 和 合并 虽然曾经 C 语言课上讲过类似的问题, 但是这个问题的变化是对单词分割符不设限制, 只要非字母, 都需要当成分割符, 并且分割符的数目不受限制.

【解题思路】:

我们整体的思路仍然是

1. 先切分(切分前先对分割符做处理, 统一分割符), 此时可以使用 stringstream 进行切分
2. 再合并(对切分结果逆序合并) 直接字符串拼接合并即可.

```
#include<iostream>
#include<sstream>
#include<string>
#include<vector>
using namespace std;

void deal(vector<string>&res,string& str, char delim = ' ') {
    // 对字符串进行切分, 借助 stringstream 完成. 默认分割符是空格.
    // stringstream 不光能完成切分, 还能完成字符串和数字之间的转换.
    stringstream ss;
    ss << str;
    string tmp;
    while (ss>>tmp) {
        res.push_back(tmp);
    }
}

int main() {
    string str;
    while (getline(cin, str)) {
        vector<string> res;
        for (int i = 0; i < str.size(); i++)
            // 先对分割符进行处理. 如果发现分割符, 就统一处理成空格.
            if (!isalnum(str[i]))
                str[i] = ' ';
        // 对字符串进行切分
        deal(res, str);
        cout << res[res.size() - 1];
        for (int i = res.size() - 2; i >= 0; i--)
            cout << " " << res[i];
        cout << endl;
    }

    return 0;
}
```

比特科技制作