

# C++方向编程题答案

## 答案说明:

大家如果对本次题目或者答案有问题，可以联系下方的出题老师答疑。

### 出题老师:

选择题: 张文超 qq: 3627274478

代码题: 时亮益 qq: 569334855

## 第七周

### day37

题目ID: 797 --数据库连接池

链接: <https://www.nowcoder.com/questionTerminal/05f97d9b29944c018578d98d7f0ce56e>

### [题目解析]

1. 服务器后台访问数据库时，需要先连上数据库，而为了连上数据库，需要连接方和数据库服务器上各自都要维护链接，而链接本身的创建维护是有成本的
2. 要减少开销，这里的解决方案就是将以前的mysql链接复用起来，从而节省了创建和销毁的时间成本，该题目就是想让你根据算法，结合日志，计算当前的业务需要预先创建多少链接，来满足正常使用

### [解题思路]

循环接收每组用例，对于每组用例进行如下操作：

1. 依次获取每个状态，如果该状态是"connect"，则将其id插入到set中，否则删除该id
2. 获取从刚开始到目前set中的最大值
3. 输出结果

```
#include <iostream>
using namespace std;

#include <string>
#include <set>

int main()
{
    int n;
    while(cin>>n)
    {
        string id, op;
        set<string> s;
        size_t maxCon = 0;
        while(n-->0)
        {
            cin>>id>>op;
            if(op == "connect")
                s.insert(id);
            else
```

```

        s.erase(id);

        maxCon = max(maxCon, s.size());
    }

    cout<<maxCon<<endl;
}
return 0;
}

```

835-mkdir

<https://www.nowcoder.com/questionTerminal/433c0c6a1e604a4795291d9cd7a60c7a>

### [题目解析]

1. 该题目需要你了解 `mkdir -p` 命令，该命令可以递归创建目录
2. 但是该题目其实想让我们解决的问题是，给你一组数据（详情见题目），怎么用最少的 `mkdir -p` 创建出我们想要的多个目录

### [解题思路]

只要按照题面，将前缀字符串路径和相同路径剔除，只 `mkdir -p` 最长或者唯一路径就可以解决这个问题

循环接收每组用例，对于每组用户进行如下操作

1. 对改组用例按照字典顺序排序
2. 将改组用例中相等的或者前一个是后一个子串的路径剔除
3. 输入结果并换行

```

#include <iostream>
using namespace std;
#include <string>
#include <vector>
#include <algorithm>

int main()
{
    int n;
    while(cin>>n)
    {
        vector<string> path(n);

        // 接收所有的路径
        for(int i = 0; i < n; ++i)
            cin>>path[i];

        // 对所有路径按照字典序列进行排序
        sort(path.begin(), path.end());

        vector<bool> flag(n, true);
        // 将所有相等或者前一个是后一个子串的字符串剔除掉：此处使用标记剔除
        for(int i = 0; i < n-1; ++i)
        {
            if(path[i] == path[i+1]) // 两个字符串相等
            {
                flag[i] = false;
            }
        }
    }
}

```

```
    }
    // 前一个字符串是后一个字符串的子串
    if(path[i].size() < path[i+1].size() &&
        (path[i+1].substr(0, path[i].size()) == path[i] && path[i+1]
[path[i].size()] == '/'))
    {
        flag[i] = false;
    }
}

// 输出结果
for(int i = 0; i < n; ++i)
{
    if(flag[i])
    {
        cout<<"mkdir -p "<<path[i]<<endl;
    }
}
cout<<endl;
}
return 0;
}
```

比特就业课