C++方向编程题答案

第四周

day19

> **题目ID：36846-汽水瓶**
>
> 链接：https://www.nowcoder.com/practice/fe298c55694f4ed39e256170ff2c205f?tpId=37&&tqId=21245&rp=1&ru=/activity/oj&qru=/ta/huawei/question-ranking
>
> **【题目解析】：**
>
> 本题题意明确
>
> **【解题思路】：**
>
> 本题题意简单，每次空瓶的数量除以2，直到最后空瓶的数量少于两瓶，就累加到了课兑换的数量。
>
> **【实例代码】**

```
#include<iostream>
#include<string>

using namespace std;

int calculateNum(int num)
{
    //总兑换数
    int sum = 0;
    while (num > 1)
    {
        //每三瓶换一瓶
        int result = num / 3;
        //剩余不足三瓶的先保留
        int reminder = num % 3;
        sum = sum + result;
        //下一次可以兑换的空瓶
        num = result + reminder;

        if (num == 2)
        {
            ++sum;
            break;
        }
    }
    return sum;
}

int main(){


 int number;
```

```
 while (cin >> number)
 {
     cout << calculateNum(number) << endl;
 }
 return 0;
}
```

题目ID:36889-查找两个字符串a,b中的最长公共子串

链接：https://www.nowcoder.com/practice/181a1a71c7574266ad07f9739f791506?tpId=37&&tqId=21288&rp=1&ru=/activity/oj&qru=/ta/huawei/question-ranking

【题目解析】：

本题题意明确

【解题思路】：

本题需要用动态规划求解，MCS[i][j]记录短字符串 s1 前 i 个字符和长字符串 s2 前 j 个字符的最长子串的长度，初始化所有值为 0。当 s1[i-1] = s2[j-1]时，MCS[i][j] = MCS[i - 1][j - 1] + 1，这里使用一个额外的值 start 来记录最长子串在短字符串 s1 中出现的起始位置，maxlen记录当前最长子串的长度，当MCS[i][j] > maxlen 时，maxlen = MCS[i][j]， 则start = i - maxlen ； 档s1[i-1] != s2[j-1]时不需要任何操作，最后获取substr(start, maxlen)即为所求。

【示例代码】

```
#include<iostream>
#include<string>
#include<algorithm>
#include<vector>
using namespace std;
int main()
{
    string str1, str2;
    while (cin >> str1 >> str2)
    {
        //以最短的字符串作为s1
        if (str1.size() > str2.size())
            swap(str1, str2);

        int len1 = str1.size(), len2 = str2.size();
        int i, j, start = 0, max = 0;

        vector<vector<int>> MCS(len1 + 1, vector<int>(len2 + 1, 0));
        for (i = 1; i <= len1; i++)
            for (j = 1; j <= len2; j++)
            {
                if (str1[i - 1] == str2[j - 1])
                    MCS[i][j] = MCS[i - 1][j - 1] + 1;
                //如果有更长的公共子串，更新长度
                if (MCS[i][j] > max)
                {

                    max = MCS[i][j];
```

```
                    //以i结尾的最大长度为max，则子串的起始位置为i - max
                    start = i - max;
                }
            }
        cout << str1.substr(start, max) << endl;
    }
    return 0;
}
```