

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：鲍松山 qq：365690203

代码题：鲍松山 qq：365690203

第四周

day22

题目ID：36487-小易的升级之路

链接：<https://www.nowcoder.com/practice/fe6c73cb899c4fe1bdd773f8d3b42c3d?tpId=49&&tqId=29329&rp=1&ru=/activity/oj&gru=/ta/2016test/question-ranking>

【题目解析】：

本题题意明确

【解题思路】：

本题的能力值的累加分两种情况，一种是直接相加 b_i ，一种是累加当前能力值于 b_i 的最大公约数。最大公约数可以通过辗转相除法求得： a 与 b 的最大公约数相当于 b 与 a, b 余数的最大公约数。如果求余结果为0，则 b 为所求结果

【示例代码】

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

//辗转相除法
int GCD(int a, int b)
{
    int c;
    while(c = a % b)
    {
        a = b;
        b = c;
    }
    return b;
}

int getPowerValue(int n, int power)
{
    vector<int> num(n);

    for(int i=0; i<n; ++i) //输入敌人的防御值
```

```

        cin>>num[i];

    for(int i=0; i<n; ++i)
    {
        if(power >= num[i])
            power += num[i];
        else
            power += GCD(power, num[i]);
    }
    return power;
}

int main()
{
    int n, a, power;
    while(cin >> n >> a)
    {
        power = getPowerValue(n, a);
        cout<<power<<endl;
    }
    return 0;
}

```

题目ID:36883-找出字符串中第一个只出现一次的字符

链接: <https://www.nowcoder.com/practice/e896d0f82f1246a3aa7b232ce38029d4?tpId=37& tqId=21282&rp=1&ru=/activity/oj&qr=/ta/huawei/question-ranking>

【题目解析】：

本题题意明确

【解题思路】：

用一个数组的每一个位置表示对应的位置。对应的字符位置存放字符出现的次数。统计完之后，遍历输入字符，遇到第一个只出现一次的字符就停止

【示例代码】

```

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

//暴力法 O(n^2)
char getFirstOneChar_1(const string &str)
{
    int j;
    for(int i=0; i<str.size(); ++i)
    {
        for(j=0; j<str.size(); ++j)
        {
            if(j == i) //排除自己跟自己比较
                continue;
            if(str[j] == str[i])

```

```

        break;
    }
    if(j >= str.size())
        return str[i];
    }
    return -1;
}

//hash法 O(n)
char getFirstOneChar_2(const string &str)
{
    int hash[256] = {0};
    for(int i=0; i<str.size(); ++i) //统计字符的次数
        hash[str[i]]++;

    for(int i=0; i<str.size(); ++i)
    {
        if(hash[str[i]] == 1)
            return str[i];
    }
    return -1;
}

//string类函数查找法
char getFirstOneChar_3(const string &str)
{
    for(int i=0; i<str.size(); ++i)
    {
        int index1 = str.find(str[i]);
        int index2 = str.rfind(str[i]);
        if(index1 == index2)
            return str[i];
    }
    return -1;
}

int main()
{
    string str;
    char res;
    while(getline(cin, str))
    {
        res = getFirstOneChar_3(str);
        if(res == -1)
            cout<<-1<<endl;
        else
            cout<<res<<endl;
    }
    return 0;
}

```