

编程作业: 从现实问题到计算机程序

✓ 通过但未认证 · 100/100 分

截止时间 The assignment was due on 一月 15, 11:59 晚上 PST
您仍可以在课程结束前完成此作业。

说明

我提交的作业

讨论

编程题 # 1: 晶晶赴约会

来源: POJ (Coursera声明: 在POJ上完成的习题将不会计入Coursera的最后成绩。)

注意: 总时间限制: 1000ms 内存限制: 65536kB

描述

晶晶的朋友贝贝约晶晶下周一起去看展览，但晶晶每周的1、3、5有课必须上课，请帮晶晶判断她能否接受贝贝的邀请，如果能输出YES；如果不能则输出NO。

输入

输入有一行，贝贝邀请晶晶去看展览的日期，用数字1到7表示从星期一到星期日。

输出

输出有一行，如果晶晶可以接受贝贝的邀请，输出YES，否则，输出NO。注意**YES**和**NO**都是大写字母！

样例输入

```
1 第一组
2 1
3 第二组
4 2
5 第三组
6 3
```

样例输出

```
1  第一组
2  NO
3  第二组
4  YES
5  第三组
6  NO
```

参考答案

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int a;
5      cin >> a;
6      if (a == 1 || a == 3 || a == 5)
7          cout << "NO" << endl;
8      else
9          cout << "YES" << endl;
10     return 0;
11 }
12 //这道题比较简单啦，和A+B是一个级别的，看懂应该没什么问题。
```

编程题 # 2：奇数求和

来源: POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

注意： 总时间限制: **1000ms** 内存限制: **65536kB**

描述

计算正整数 m 到 n （包括 m 和 n ）之间的所有奇数的和，其中， m 不大于 n ，且 n 不大于 300。例如 $m=3, n=12$, 其和则为： $3+5+7+9+11=35$

输入

两个数 m 和 n ，两个数以空格间隔，其中 $0 \leq m \leq n \leq 300$ 。

输出

奇数之和

样例输入

```
1  第一组
2  7 15
3  第二组
4  0 1
5  第三组
6  3 3
7  第四组
8  100 100
```

样例输出

```
1  第一组
2  55
3  第二组
4  1
5  第三组
6  3
7  第四组
8  0
```

参考答案

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int m, n, result = 0;
5      cin >> m >> n;
6      while (m <= n) {
7          //对于m和n之间的每一个数，
              如果它是奇数，那么就加入到我们的结果里。如果不是就跳过。
8          if (m % 2 == 1)
9              result += m;
10         m++;
11     }
12     //最后输出
13     cout << result << endl;
14     return 0;
15 }
16 //其实还有更快的算法你能想到吗？
```

编程题 # 3：整数的个数

来源: POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

注意： 总时间限制: **1000ms** 内存限制: **65536kB**

描述

给定k ($1 < k < 100$) 个正整数，其中每个数都是大于等于1，小于等于10的数。写程序计算给定的k个正整数中，1，5和10出现的次数。

输入

输入有两行：第一行包含一个正整数k，第二行包含k个正整数，每两个正整数用一个空格分开。

输出

输出有三行，第一行为1出现的次数，，第二行为5出现的次数，第三行为10出现的次数。

样例输入

```
1  第一组
2  5
3  1 5 8 10 5
4  第二组
5  5
6  2 2 2 2 2
```

样例输出

```
1  第一组
2  1
3  2
4  1
5  第二组
6  0
7  0
8  0
```

参考答案

```
1  #include <iostream>
2  using namespace std;
3  int main(){
4      int k;
5      cin>>k;
6      int n1=0, n5=0, n10=0;
7      for (int i=0;i<k;i++){
8          int n;
9          cin>>n;
10         if (n == 1) n1++;
11         else if (n == 5) n5++;
12         else if (n == 10) n10++;
13     }
14     cout<<n1<<endl;
15     cout<<n5<<endl;
16     cout<<n10<<endl;
17     return 0;
18 }
```

编程题 # 4: 1的个数

来源: POJ (Coursera声明: 在POJ上完成的习题将不会计入Coursera的最后成绩。)

注意: 总时间限制: **1000ms** 内存限制: **65536kB**

描述

给定一个十进制整数N，求其对应2进制数中1的个数

输入

第一个整数表示有N组测试数据，其后N行是对应的测试数据，每行为一个整数。

输出

N行，每行输出对应一个输入。

样例输入

1	5
2	2
3	100
4	1000
5	66
6	0

样例输出

1	1
2	3
3	6
4	2
5	0

提示

这道题有一个特点，那就是输入的数据的个数不止一组了。但是我们知道数组的总数量，所以可以用计数循环来读入数据。

那么输出应该怎么办呢？在我们的评分系统上，输入和输出是分别放在两个地方处理的，所以可以提早把输出结果打印出来，不会干扰后面的输入。

参考答案

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cin >> n;
7     for (int i = 0; i < n; i++) {
8         int x, ans = 0;
9         cin >> x;
10        while (x > 0) {
11            ans += x % 2;
12            x /= 2;
13        }
14        cout << ans << endl;
15    }
16    return 0;
17 }
18 //这道题的解法就是反复地除以2，看最低位是1还是0。有些知道位运算
    的同学喜欢用位运算，但其实没必要，因为编译器优化之后的程序
    其实效率是一样的。
```

编程题 # 5：数组逆序重放

来源: POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

注意： 总时间限制: **1000ms** 内存限制: **65536kB**

描述

将一个数组中的值按逆序重新存放。例如，原来的顺序为8,6,5,4,1。要求改为1,4,5,6,8。

输入

输入为两行：第一行数组中元素的个数n ($1 < n < 100$)，第二行是n个整数，每两个整数之间用空格分隔。

输出

输出为一行：输出逆序后数组的整数，每两个整数之间用空格分隔。

样例输入

```
1 5
2 8 6 5 4 1
```

样例输出

```
1 1 4 5 6 8
```

参考答案

```
1  #include <iostream>
2  using namespace std;
3  int a[100];
4  int main() {
5      int n;
6      cin >> n;
7      for (int i = 0; i<n; i++)
8          cin >> a[i];
9      while (n--) { //常用的倒序计数循环，等价于while(n-->0)
10         cout << a[n];
11         if (n > 0) cout << " ";
12         //如果不是最后一个数那么就要用空格分隔开
13     }
14     return 0;
15 } //顺序输入倒序输出，但是如果我们要求通过修改数组本身实现逆序重
    放应该怎么做呢？
```

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.

