

# Assignment #1: 摸底考试

Updated 1749 GMT+8 Sep 14, 2023

2023 fall, Compiled by Hongfei Yan== (请改为同学的姓名、院系) ==

Markdown (用 <https://typoraio.cn> 编辑) 格式文件在, <https://github.com/GMyhf/2023fall-cs101>

课程号: 04831410 课程名: 计算概论(B)	班号: 12
上课时间: 1-16周 每周 周二 7-9节	地点: 理教208
上机时间: 2-15周 每周 周四 7-8节 期末机考时间: 第16周 周四 7-8节	地点: 理科1号楼计算中心, 二层楼的6号和三层楼的7号机房
助教: 张哲瑞、张以宁、彭亦男、涂程颖、陈威宇	在课程微信群中的名字是“TA-”开始, 地点: 理科1号楼1220

为了尽量拉齐大家的学习基础, 尤其是零基础同学的强化学习, 第一周就进行了语法摸底考试。鉴于Canvas 9月22日开始使用, 这之前大家写好作业, 先自己保存, 之后通知提交到canvas.

摸底考试: AC6

请把每个题目解题思路(可选), 源码Python, 或者C++/C (已经在Codeforces/Openjudge上AC), 截图, 填写到下面作业模版中(例如使用: typora <https://typoraio.cn>, 或者word) 中提交。如果不能按时完成, 请写明大致花费时间, 原因。

教材、题解等课程相关材料, 请在课程微信群索要。

## 1. 编程环境

操作系统: Windows 11

Python编程环境: Spyder IDE 5.2.2

## 2. 必做题目

### OJ02750: 鸡兔同笼

math, <http://cs101.openjudge.cn/practice/02750>

一个笼子里面关了鸡和兔子(鸡有2只脚, 兔子有4只脚, 没有例外)。已经知道了笼子里面脚的总数 $a$ , 问笼子里面至少有多少只动物, 至多有多少只动物。

输入

一行, 一个正整数 $a$  ( $a < 32768$ ).

输出

一行，包含两个正整数，第一个是最少的动物数，第二个是最多的动物数，两个正整数用一个空格分开。

如果没有满足要求的答案，则输出两个0，中间用一个空格分开。

样例输入

20

样例输出

5 10

【苏王捷，工学院，2023年秋】

思路：求最大鸡数量和最大兔数量

### Python3 代码

```
um=int(input())
if num%2==0:
    maxi=num//2
    mini=num//4+num%4//2
else:
    maxi=0
    mini=0
print(str(mini)+" "+str(maxi))
```

Python代码运行截图 ==

状态: Accepted

源代码

```
num=int(input())
if num%2==0:
    maxi=num//2
    mini=num//4+num%4//2
else:
    maxi=0
    mini=0
print(str(mini)+" "+str(maxi))
```

==

## OJ02733：判断闰年

math, <http://cs101.openjudge.cn/practice/02733>

判断某年是否是闰年。

### 输入

输入只有一行，包含一个整数 $a$  ( $0 < a < 3000$ )

### 输出

一行，如果公元 $a$ 年是闰年输出Y，否则输出N

样例输入

```
2006
```

样例输出

```
N
```

### 提示

公历纪年法中，能被4整除的大多是闰年，但能被100整除而不能被400整除的年份不是闰年，能被3200整除的也不是闰年，如1900年是平年，2000年是闰年，3200年不是闰年。

【苏王捷，工学院，2023年秋】

思路：闰年判断

### Python3 代码

```
year=int(input())
if year%4==0:
    if year%100==0:
        if year%400==0:
            print("Y")
        else:
            print("N")
    else:
        print("Y")
else:
    print("N")
```

Python代码运行截图 ==

状态: **Accepted**

源代码

```
year=int(input())
if year%4==0:
    if year%100==0:
        if year%400==0:
            print("Y")
        else:
            print("N")
    else:
        print("Y")
else:
    print("N")
```

==

## OJ01218:THE DRUNK JAILER

<http://cs101.openjudge.cn/practice/01218/>

A certain prison contains a long hall of  $n$  cells, each right next to each other. Each cell has a prisoner in it, and each cell is locked.

One night, the jailer gets bored and decides to play a game. For round 1 of the game, he takes a drink of whiskey, and then runs down the hall unlocking each cell. For round 2, he takes a drink of whiskey, and then runs down the hall locking every other cell (cells 2, 4, 6, ...). For round 3, he takes a drink of whiskey, and then runs down the hall. He visits every third cell (cells 3, 6, 9, ...). If the cell is locked, he unlocks it; if it is unlocked, he locks it. He

repeats this for  $n$  rounds, takes a final drink, and passes out.

Some number of prisoners, possibly zero, realizes that their cells are unlocked and the jailer is incapacitated. They immediately escape.

Given the number of cells, determine how many prisoners escape jail.

### 输入

The first line of input contains a single positive integer. This is the number of lines that follow. Each of the following lines contains a single integer between 5 and 100, inclusive, which is the number of cells  $n$ .

### 输出

For each line, you must print out the number of prisoners that escape when the prison has  $n$  cells.

样例输入

2  
5  
100

样例输出

2  
10

来源

Greater New York 2002

【苏王捷，工学院，2023年秋】

思路：猪b循环，其实可以直接判断完全平方数

### Python3 代码

```
n=int(input())
for i in range(n):
    cell=int(input())
    prison=[0]*cell
    for j in range(cell):
        for _ in range(cell):
            if (_+1)%(j+1)==0:
                prison[_]=(prison[_]+1)%2
    print(sum(prison))
```

Python代码运行截图 ==

状态: Accepted

源代码

```
n=int(input())
for i in range(n):
    cell=int(input())
    prison=[0]*cell
    for j in range(cell):
        for _ in range(cell):
            if (_+1)%(j+1)==0:
                prison[_]=(prison[_]+1)%2
    print(sum(prison))
```

==

## OJ02689: 大小写字母互换

<http://cs101.openjudge.cn/practice/02689>

把一个字符串中所有出现的大写字母都替换成小写字母，同时把小写字母替换成大写字母。

### 输入

输入一行：待互换的字符串。

### 输出

输出一行：完成互换的字符串（字符串长度小于80）。

样例输入

```
If so, you already have a Google Account. You can sign in on the right.
```

样例输出

```
iF SO, YOU ALREADY HAVE A gOOGLe aCCOUNT. yOU CAN SIGN IN ON THE RIGHT.
```

来源

计算概论05

【苏王捷，工学院，2023年秋】

思路：用ASCII码转换

### Python3 代码

```
word=input()
word_new=""
for i in word:
    if 97<=ord(i)<=122:
        word_new+=chr(ord(i)-32)
    elif 65<=ord(i)<=90:
        word_new+=chr(ord(i)+32)
    else:
        word_new+=i
print(word_new)
```

Python代码运行截图 ==

状态: Accepted

源代码

```
word=input()
word_new=""
for i in word:
    if 97<=ord(i)<=122:
        word_new+=chr(ord(i)-32)
    elif 65<=ord(i)<=90:
        word_new+=chr(ord(i)+32)
    else:
        word_new+=i
print(word_new)
```

==

C++ 代码

```
// 请改为同学的代码
```

C++ 代码运行截图 == (请粘贴同学的截图) ==

## OJ02808: 校门外的树

implementation, <http://cs101.openjudge.cn/practice/02808>

某校大门外长度为L的马路上有一排树，每两棵相邻的树之间的间隔都是1米。我们可以把马路看成一个数轴，马路的一端在数轴0的位置，另一端在L的位置；数轴上的每个整数点，即0, 1, 2, ....., L, 都种有一棵树。

马路上有一些区域要用来建地铁，这些区域用它们在数轴上的起始点和终止点表示。已知任一区域的起始点和终止点的坐标都是整数，区域之间可能有重合的部分。现在要把这些区域中的树（包括区域端点处的两棵树）移走。你的任务是计算将这些树都移走后，马路上还有多少棵树。

### 输入

输入的第一行有两个整数L (1 ≤ L ≤ 10000) 和 M (1 ≤ M ≤ 100)，L代表马路的长度，M代表区域的数目，L和M之间用一个空格隔开。接下来的M行每行包含两个不同的整数，用一个空格隔开，表示一个区域的起始点和终止点的坐标。

### 输出

输出包括一行，这一行只包含一个整数，表示马路上剩余的树的数目。

样例输入

```
500 3
150 300
100 200
470 471
```

样例输出

```
298
```

来源：noip2005普及组

【苏王捷，工学院，2023年秋】

思路：用1和0表示树的有无，直接替换

**Python3 代码**

```
tree,n=map(int,input().split())
line=[]
for i in range(n):
    line.append([int(i) for i in input().split()])
road=[1]*(tree+1)
for i in range(n):
    for j in range(line[i][0]-1,line[i][1]):
        road[j]=0
print(sum(road))
```

Python代码运行截图 ==

状态: **Accepted**

源代码

```
tree,n=map(int,input().split())
line=[]
for i in range(n):
    line.append([int(i) for i in input().split()])
road=[1]*(tree+1)
for i in range(n):
    for j in range(line[i][0]-1,line[i][1]):
        road[j]=0
print(sum(road))
```

==



## C++ 代码

```
// 请改为同学的代码
```

C++ 代码运行截图 == (请粘贴同学的截图) ==

## 3. 选做题目

### OJ25353: 排队

Greedy, <http://cs101.openjudge.cn/practice/25353/>

有  $N$  名同学从左到右排成一排，第  $i$  名同学的身高为  $h_i$ 。现在张老师想改变排队的顺序，他能进行任意多次（包括0次）如下操作：

- 如果两名同学相邻，并且他们的身高之差不超过  $D$ ，那么老师就能交换他俩的顺序。

请你帮张老师算一算，通过以上操作，字典序最小的所有同学（从左到右）身高序列是什么？

输入

第一行包含两个正整数  $N, D$  ( $1 \leq N \leq 10^5, 1 \leq D \leq 10^9$ )。

接下去  $N$  行，每行一个正整数  $h_i$  ( $1 \leq h_i \leq 109$ ) 表示从左到右每名同学的身高。

输出

输出  $N$  行，第  $i$  行表示答案中第  $i$  名同学的身高。

样例输入

```
5 3
7
7
3
6
2
```

样例输出

```
6
7
7
2
3
```

提示

【样例解释】

一种交换位置的过程如下：

```
7 7 3 6 2-> 7 7 6 3 2-> 7 7 6 2 3-> 7 6 7 2 3-> 6 7 7 2 3
```

### 【数据范围和约定】

对于 10% 的数据，满足  $N \leq 100$ ;

对于另外 20% 的数据，满足  $N \leq 5000$ ;

对于全部数据，满足  $1 \leq N \leq 10^5, 1 \leq D \leq 10^9, 1 \leq h_i \leq 10^9$ 。

【苏王捷，工学院，2023年秋】

思路：将所有能提前至第一位的数从大到小排序，循环添加直至排完

### Python3 代码

```
N,D=map(int,input().split())
height=[]
for i in range(N):
    height.append(int(input()))
height_new=[]
while True:
    try:
        tem=[]
        cap=height[0]
        maximun=height[0]
        minimun=height[0]
        for i in height[0::]:
            maximun=max(maximun,i)
            minimun=min(minimun,i)
            if abs(i-maximun)<=D and abs(i-minimun)<=D and 0<=cap-i<=D:
                tem.append(i)
        tem.sort()
        for i in range(len(tem)):
            height.remove(tem[i])
        height_new.extend(tem)
    except IndexError:
        break
for i in height_new:
    print(i)
```

Python代码运行截图 ==

状态: Accepted

源代码

```
N,D=map(int,input().split())
height=[]
for i in range(N):
    height.append(int(input()))
height_new=[]
while True:
    try:
        tem=[]
        cap=height[0]
        maximun=height[0]
        minimun=height[0]
        for i in height[0::]:
            maximun=max(maximun,i)
            minimun=min(minimun,i)
            if abs(i-maximun)<=D and abs(i-minimun)<=D and 0<=cap-i<=D:
                tem.append(i)
        tem.sort()
        for i in range(len(tem)):
            height.remove(tem[i])
        height_new.extend(tem)
    except IndexError:
        break
for i in height_new:
    print(i)
```

==

## 4. 学习总结和收获

代码优化可以不使用remove等操作而是改变列表字符，大大节省时间