

Assignment #1: 摸底考试

Updated 1323 GMT+8 Sep 18, 2023

2023 fall, Compiled by 刘思瑞 2100017810

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 |

说明:

1) 为了尽量拉齐大家的学习基础, 尤其是督促零基础同学的强化学习, 第一周就进行了语法摸底考试。鉴于教学管理平台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。

2) 摸底考试: 未参加。摸底题目都在“练习”里面, 按照数字题号能找到, 可以重新提交。作业中提交自己最满意版本的代码和截图。

3) 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(包含Accepted, 学号, 或者姓名+学号), 填写到下面作业模版中(推荐使用 typora <https://typoraio.cn>, 或者用word)。AC或者没有AC, 都请标上每个题目大致花费时间。

4) 提交时候先提交pdf文件, 再把md或者doc文件上传到右侧“作业评论”。提交后, 助教看到画面如下, 有同学清晰头像、提交文件有pdf、作业评论有md或者doc。

Assignment #3: Programming Practice
截止：2022年9月27日 23:59 - 04831410

212/212
已评分

3.92 / 4 (98%)
平均分

1/212

← 恒环学院 →

第 1 页, 共 2 页

282A. Bit++, implementation
<https://codeforces.com/problemset/problem/282/A>
t=int(input());c=0
while(t):
 strn=input()
 i=0
 while(i<len(strn)):
 if strn[i]=='+':
 c+=1;i+=2
 elif strn[i]=='-':
 c-=1;i+=2
 elif strn[i]=='X':
 c+=0;i+=1
 # c=count+c

 t-=1
print(c)

已提交：2022年 9月26日 13:27

字数统计: 124 个字

提交的文件: (点击加载)
[Assignment 3.pdf](#)

测验
评分 共4
4

作业评论
[查看附件文件](#)
[Assignment 3.md](#)
恒城市与环境学院, 2022年 9月26日 13:28
[下载提交项评论](#)

5) 同学完成作业的时候, 就是这个模版文件中修改补充好。为便于助教批改作业, 请尽量不要删除其他文字。

6) 如果不能在截止前提交作业, 请写明原因。

编程环境

操作系统: Windows 11 22H2 22621.2283

Python编程环境: Visual Studio (1.82.2); python 3.11.3

C/C++编程环境: 无

1. 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

样例输出

```
1 | 5 10
```

【刘思瑞，元培学院物理方向，2023年秋】

思路：

首先保证一定会有解，判断是否是偶数；如果不是的话，最大值就直接除以2就可以，最小值先对四整除再除以2（又可能还有2的余数）。当然这样操作是基于题目中2和4的数字特征进行的，如果改成3和5的话还是要直接去遍历。

Python3 代码

```
1  """
2  刘思瑞 2100017810
3  """
4  num = int(input())
5  if num%2 != 0:
6      print("0 0")
7  else:
8      a = num//4 + (num%4)/2
9      b = num/2
10     print(str(int(a))+ " "+str(int(b)))
```

Python代码运行截图

状态: Accepted

源代码

```
"""
刘思瑞 2100017810
"""

num = int(input())
if num%2 != 0:
    print("0 0")
else:
    a = num//4 + (num%4)/2
    b = num/2
    print(str(int(a))+" "+str(int(b)))
```

©2002-2022 POJ 京ICP备20010980号-1

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

```
1 | 2006
```

样例输出

```
1 | 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。

【刘思瑞，元培学院物理方向，2023年秋】

思路：

直接判断就好。

Python3 代码

```
1  """
2  刘思瑞 2100017810
3  """
4  year = int(input())
5  if year%4 == 0:
6      if year%100 == 0:
7          if year%400 != 0:
8              print("N")
9          else:
10             print("Y")
11     else:
12         print("Y")
13 else:
14     print("N")
```

Python代码运行截图

#41250920提交状态

状态: Accepted

源代码

```
"""
刘思瑞 2100017810
"""
year = int(input())
if year%4 == 0:
    if year%100 == 0:
        if year%400 != 0:
            print("N")
        else:
            print("Y")
    else:
        print("Y")
else:
    print("N")
```

©2002-2022 POJ 京ICP备20010980号-1

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

```
1 | 2
2 | 5
3 | 100
```

样例输出

```
1 | 2
2 | 10
```

来源

Greater New York 2002

【刘思瑞，元培学院物理方向，2023年秋】

思路：

对于每一个牢房进行判断，只需要判断所有比牢房数字小的情况是否可以整除即可，利用bool类型变量来判断牢房是否被打开

Python3 代码

```
1  """
2  刘思瑞 2100017810
3  """
4  def prison(t):
5      n = 0
6      for i in range(1,t+1):
7          m = 0
8          for j in range(1,i+1):
9              if not(i%j):
10                 m = not(m)
11             if m :
12                 n += 1
13         return n
14  a = []
15  for i in range(int(input())):
16      a.append(prison(int(input())))
17  for i in a:
18      print(i)
```

状态: Accepted

源代码

```
"""
刘思瑞 2100017810
"""
def prison(t):
    n = 0
    for i in range(1, t+1):
        m = 0
        for j in range(1, i+1):
            if not(i%j):
                m = not(m)
        if m:
            n += 1
    return n
a = []
for i in range(int(input())):
    a.append(prison(int(input())))
for i in a:
    print(i)
```

©2002-2022 POJ 京ICP备20010980号-1

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

样例输出

```
1 | if so, you already have a google account. you can sign in on the right.
```

来源

计算概论05

【刘思瑞，元培学院物理方向，2023年秋】

思路：

利用字符串切片功能，根据ASCII码进行大小写的转换

Python3 代码

```
1  """
2  刘思瑞 2100017810
3  """
4  sen = input()
5  sen_o = ''
6  for i in sen:
7      if ord(i) >= 97 and ord(i) <= 122:
8          i = chr(ord(i) - 32)
9      elif ord(i) >= 65 and ord(i) <= 90:
10         i = chr(ord(i) + 32)
11     sen_o = sen_o + i
12 print(sen_o)
```

Python代码运行截图

状态: Accepted

源代码

```
"""
刘思瑞 2100017810
"""
sen = input()
sen_o = ''
for i in sen:
    if ord(i) >= 97 and ord(i) <= 122:
        i = chr(ord(i) - 32)
    elif ord(i) >= 65 and ord(i) <= 90:
        i = chr(ord(i) + 32)
    sen_o = sen_o + i
print(sen_o)
```

©2002-2022 POJ 京ICP备20010980号-1

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 ($1 \leq L \leq 10000$) 和 M ($1 \leq M \leq 100$)， 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

```
1 500 3
2 150 300
3 100 200
4 470 471
```

样例输出

1 | 298

来源：noip2005普及组

【刘思瑞，元培学院物理方向，2023年秋】

思路：

遍历所有的需要操作的元素即可，数据量较小不会超时

Python3 代码

```
1  """
2  刘思瑞 2100017810
3  """
4  L , M = map(int, input().split())
5  tree = [1]*(L+1)
6  for i in range(M):
7      origin , determination = map(int, input().split())
8      for j in range(origin , determination+1):
9          tree[j] = 0
10 n = 0
11 for i in tree:
12     n += i
13 print(n)
```

Python代码运行截图

状态: Accepted

源代码

```
"""
刘思瑞 2100017810
"""
L, M = map(int, input().split())
tree = [1]*(L+1)
for i in range(M):
    origin, determination = map(int, input().split())
    for j in range(origin, determination+1):
        tree[j] = 0
n = 0
for i in tree:
    n += i
print(n)
```

©2002-2022 POJ 京ICP备20010980号-1

2. 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 ($1 \leq h_i \leq 10^9$) 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

```
1 | 5 3
2 | 7
3 | 7
4 | 3
5 | 6
6 | 2
```

样例输出

| | |
|---|---|
| 1 | 6 |
| 2 | 7 |
| 3 | 7 |
| 4 | 2 |
| 5 | 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 $1 \leq N \leq 10^5, 1 \leq D \leq 10^9, 1 \leq h_i \leq 10^9$ 。

【刘思瑞，元培学院物理方向，2023年秋】

思路：

首先是想到了一个最合理的操作过程：因为最后排在第一个的元素一定能够和他前面的所有元素交换（因为只有交换才能改变两个元素的先后顺序，如果不能够全部交换那么必然会有一个元素一直在这个元素的前面）。基于这个想法，我写了第一个代码，每次遍历数组，将不同元素的可交换区间取交集，接下来判断下一个元素是否在这个区间内，找到最小的一个或者当区间为空时break，接下来从数组里面pop掉这个元素再重新调用函数，进行迭代n次直到找出所有的元素。但是这个code在oj上大概跑了30s就超时了，感觉大概卡在12、13数据点左右。

接下来为了改进代码我想到了内存和算法两方面。因为数组开辟连续的内存空间，在数组里调用pop方法的复杂度很高，因此我尝试了链表，在时间上有一定的改进但是还是卡在了十多个数据点。之后我尝试改进算法，感谢@23 数院胡睿诚 同学告诉我可以一次性找到所有的可交换元素再进行排序，这样显著降低了迭代的次数。卡了一天终于在oj上过了。

Python3 代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 def exchange(t):
5     global d, num
6     if t.head:
7         list1 = [t.head.data]
8         temp = t.head
9         minnum = t.head.data - d
10        maxnum = t.head.data + d
11        while True:
12            try:
13                if temp.data - d > minnum:
14                    minnum = temp.data - d
15                if temp.data + d < maxnum:
16                    maxnum = temp.data + d
17                if maxnum < minnum:
18                    break
```

```

19         if (temp.next.data >= minnum) and (temp.next.data <=
maxnum):
20             list1.append(temp.next.data)
21             if temp.next.data - d > minnum:
22                 minnum = temp.next.data -d
23             if temp.next.data + d < maxnum:
24                 maxnum = temp.next.data + d
25             if maxnum < minnum:
26                 break
27             temp.delete_by_element()
28         else:
29             temp = temp.next
30         except AttributeError:
31             break
32     t.head = t.head.next
33     return list1
34 else:
35     return None
36
37 class LNode(object):
38     def __init__(self, data, next=None):
39         self.data = data
40         self.next = next
41
42     def delete_by_element(self):
43         if self.next.next:
44             self.next = self.next.next
45         else:
46             self.next = None
47
48 class LinkList(object):
49     def __init__(self):
50         self.head = None
51
52     def initList(self, data):
53         self.head = LNode(data[0])
54         p = self.head
55         for i in data[1:]:
56             node = LNode(i)
57             p.next = node
58             p = p.next
59
60     def delete(self, index):
61         q = self.head
62         if index == 0:
63             self.head = self.head.next
64         else:
65             for i in range(index-1):
66                 q = q.next
67             r = q
68             if (r.next).next:
69                 q.next = r.next.next
70
71 num , d = map(int, input().split())
72 hight = []
73 for i in range(num):

```

```

74     hight.append(int(input()))
75     height = LinkList()
76     height.initList(hight)
77     list_temp = [1]
78     while list_temp:
79         list_temp = exchange(height)
80         if not list_temp:
81             break
82         list_temp.sort()
83         for j in list_temp:
84             print(j)

```

Python代码运行截图

#41268134提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

'''
刘思瑞 2100017810
'''
def exchange(t):
    global d, num
    if t.head:
        list1 = [t.head.data]
        temp = t.head
        minnum = t.head.data - d
        maxnum = t.head.data + d
        while True:
            try:
                if temp.data - d > minnum:
                    minnum = temp.data - d
                if temp.data + d < maxnum:
                    maxnum = temp.data + d
                if maxnum < minnum:
                    break
            if (temp.next.data >= minnum) and (temp.next.data <= ma
                list1.append(temp.next.data)
                if temp.next.data - d > minnum:
                    minnum = temp.next.data - d
                if temp.next.data + d < maxnum:
                    maxnum = temp.next.data + d
                if maxnum < minnum:
                    break
            temp.delete_by_element()
        else:
            temp = temp.next

```

基本信息

#: 41268134
 题目: 25353
 提交人: 23n2100017810
 内存: 25344kB
 时间: 397ms
 语言: Python3
 提交时间: 2023-09-20 19:29:42

3. 学习总结和收获

在这周练习的作业中我感觉前五题还是比较easy的，也没有太大的坑，甚至时间限制也比较宽松。这周主要头疼的还是排队这题，在试了好多变之后终于过了，还了解到了内存空间的分配和链表的写法。之后就是自己练习的题目，我在oj上找了几个相对来说不算太难的题目，我感觉邮箱验证这个题彻底教会了我debug。还有就是正则表达式确实挺好玩的，就是不知道正则表达式实现起来复杂度到底怎么样。除此之外我还在cf上做了一些最基础的语法题，感觉实在太简单了我就不放在后面了，这周因为刚刚选上课所以还不是太能适应这个题目的进度，争取下周每天都看一个cs101的练习。

额外练习的题目：

OJ01064: 网线主管

<http://cs101.openjudge.cn/practice/01064>

仙境的居民们决定举办一场程序设计区域赛。裁判委员会完全由自愿组成，他们承诺要组织一次史上最公正的比赛。他们决定将选手的电脑用星形拓扑结构连接在一起，即将它们全部连到一个单一的中心服务器。为了组织这个完全公正的比赛，裁判委员会主席提出要将所有选手的电脑等距离地围绕在服务器周围放置。为购买网线，裁判委员会联系了当地的一个网络解决方案提供商，要求能够提供一定数量的等长网线。裁判委员会希望网线越长越好，这样选手们之间的距离可以尽可能远一些。该公司的网线主管承接了这个任务。他知道库存中每条网线的长度（精确到厘米），并且只要告诉他所需的网线长度（精确到厘米），他都能够完成对网线的切割工作。但是，这次，所需的网线长度并不知道，这让网线主管不知所措。你需要编写一个程序，帮助网线主管确定一个最长的网线长度，并且按此长度对库存中的网线进行切割，能够得到指定数量的网线。

输入

第一行包含两个整数N和K，以单个空格隔开。 N ($1 \leq N \leq 10000$) 是库存中的网线数， K ($1 \leq K \leq 10000$) 是需要的网线数量。接下来N行，每行一个数，为库存中每条网线的长度（单位：米）。所有网线的长度至少1m，至多100km。输入中的所有长度都精确到厘米，即保留到小数点后两位。

输出

网线主管能够从库存的网线中切出指定数量的网线的最长长度（单位：米）。必须精确到厘米，即保留到小数点后两位。若无法得到长度至少为1cm的指定数量的网线，则必须输出“0.00”（不包含引号）。

样例输入

```
1 4 11
2 8.02
3 7.43
4 4.57
5 5.39`
```

样例输出

```
1 2.00
```

来源：Northeastern Europe 2001

【刘思瑞，元培学院物理方向，2023年秋】

思路：

就是比较常规的二分法查找，如果不用二分法会超时，然后要注意判断什么时候输出0.00

Python3 代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 a,b=input().split()
5 a= int(a)
6 b= int(b)
7 c=0
8 l=[]
9 for i in range(a):
10     l.append(float(input())*100)
```



```
11     c+=l[i]
12     d = c//b
13     if c<b:
14         print('0.00')
15     else:
16         max=d
17         min=0
18         while max-min>1:
19             p=0
20             for i in l:
21                 p+=i//d
22             if p>=b:
23                 min=d
24                 d = (max+d)//2
25             elif p<b:
26                 max=d
27                 d=(d+min)//2
28         p=0
29         for i in l:
30             p+=i//max
31
32         if p>=b:
33             d=max
34         else:
35             d=min
36         e=d/100
37         print("%.2f" % e)
```

Python代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
a,b=input().split()
a= int(a)
b= int(b)
c=0
l=[]
for i in range(a):
    l.append(float(input())*100)
    c+=l[i]
d = c//b
if c<b:
    print('0.00')
else:
    max=d
    min=0
    while max-min>1:
        p=0
        for i in l:
            p+=i//d
        if p>=b:
            min=d
            d = (max+d)//2
        elif p<b:
            max=d
            d=(d+min)//2
    p=0
    for i in l:
        p+=i//max
    if p>=b:
        d=max
    else:
        d=min
    e=d/100
    print("%.2f" % e)
```

OJ23654: 特殊年份

年号中的每个数之和为20的年号是特殊年号。例如：2099、1991、1892是特殊的年号，而2021则不是。给定一个年号，找出严格大于给定年号的最小特殊年号。

输入

年号：整数 y ($1000 \leq y \leq 9000$)。

输出

特殊年号：严格意义上大于 y 的最小年号，并且它的每个数之和为20。

样例输入

```
1 # 样例输入1
2 1892
3
4 # 样例输入2
5 2021
```

样例输出

```
1 # 样例输出1
2 1919
3
4 # 样例输出2
5 2099
```

提示

因为 $2+0+9+9=20$ ，所以2099是特殊年号。

因为 $2+0+2+1=5$ ，所以2021不是特殊年号。

特殊年号可能大于9000。

来源

改编自<https://codeforces.com/problemset/problem/271/A>

【刘思瑞，元培学院物理方向，2023年秋】

思路：

思路就是分类讨论但这个题真的是非常恶心，我实在找不出来bug了，我就把所有的特殊年份都print出来测试了所有的数据，后来果然发现一个小bug（（

Python3 代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 sum = 0
5 inn = input()
6 year = []
7 for i in inn:
8     sum += int(i)
9     year.append(int(i))
10 if sum < 20:
11     if 20 - sum <= 9 - year[3]:
12         year[3] = year[3] + 20 - sum
13     elif (20 - sum) - (9 - year[3]) <= 9 - year[2]:
14         year[2] = year[2] + (20 - sum) - (9 - year[3])
15         year[3] = 9
16     else:
17         year = [1, 1, 9, 9]
18 elif sum > 20:
19     if year[2] != 9:
20         if sum - 20 <= year[3] - 1:
21             year[2] = year[2] + 1
```

```

22         year[3] = year[3] -1 - sum +20
23     elif year[1] != 9:
24         year[1] = year[1]+1
25         if 20 - year[0] -year[1] <= 9:
26             year[2] = 0
27             year[3] = 20 - year[0] - year[1]
28         else:
29             year[3] = 9
30             year[2] =20 - year[0] - year[1]-9
31     elif year[1] == 9:
32         year[0] = year[0]+1
33         year[3] = 9
34         year[2] =20 - year[0] - 9
35         year[1] =0
36     elif year[2] == 9:
37         if year[1] != 9:
38             year[1] = year[1]+1
39             if 20 - year[0] -year[1] <= 9:
40                 year[2] = 0
41                 year[3] = 20 - year[0] -year[1]
42             else:
43                 year[3] = 9
44                 year[2] =20 - year[0] -year[1]-9
45         elif year[1] == 9:
46             year[0] = year[0]+1
47             year[3] = 9
48             year[2] = 20 - year[0] - 9
49             year[1] = 0
50     else:
51         if year[2] != 9 and year[3] != 0:
52             year[2] = year[2]+1
53             year[3] = year[3]-1
54         else:
55             if year[1] != 9:
56                 year[1] = year[1]+1
57                 if 20 - year[0]- year[1] <= 9:
58                     year[3] = 20 - year[0] - year[1]
59                     year[2] = 0
60                 else:
61                     year[3] = 9
62                     year[2] = 20 -year[0] - year[1]-9
63             else:
64                 year[0] += 1
65                 year[3] = 9
66                 year[2] = 20 -year[0] -9
67                 year[1] = 0
68     print(str(year[0])+str(year[1])+str(year[2])+str(year[3]))

```

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
sum = 0
inn = input()
year = []
for i in inn:
    sum += int(i)
    year.append(int(i))
if sum < 20:
    if 20 - sum <= 9 - year[3]:
        year[3] = year[3] + 20 - sum
    elif (20 - sum) - (9 - year[3]) <= 9 - year[2]:
        year[2] = year[2] + (20 - sum) - (9 - year[3])
        year[3] = 9
    else:
        year = [1, 1, 9, 9]
elif sum > 20:
    if year[2] != 9:
        if sum - 20 <= year[3] - 1:
            year[2] = year[2] + 1
            year[3] = year[3] - 1 - sum + 20
        elif year[1] != 9:
            year[1] = year[1] + 1
            if 20 - year[0] - year[1] <= 9:
                year[2] = 0
                year[3] = 20 - year[0] - year[1]
            else:
                year[3] = 9
                year[2] = 20 - year[0] - year[1] - 9
        elif year[1] == 9:
            year[0] = year[0] + 1
            year[3] = 9
            year[2] = 20 - year[0] - 9
            year[1] = 0
    elif year[2] == 9:
```

OJ04015:邮箱验证

POJ 注册的时候需要用户输入邮箱，验证邮箱的规则包括：

- 1)有且仅有一个'@'符号
 - 2) '@'和'.'不能出现在字符串的首和尾
 - 3) '@'之后至少要有个'.'，并且 '@'不能和'.'直接相连
- 满足以上3条的字符串为合法邮箱，否则不合法，
编写程序验证输入是否合法

输入

输入包含若干行，每一行为一个待验证的邮箱地址，长度小于100

输出

每一行输入对应一行输出

如果验证合法，输出 YES

如果验证非法：输出 NO

样例输入

```
1 .a@b.com
2 pku@edu.cn
3 cs101@gmail.com
4 cs101@gmail
```

样例输出

```
1 NO
2 YES
3 YES
4 NO
```

【刘思瑞，元培学院物理方向，2023年秋】

思路：

利用正则表达式，先把所有违规的都排除，再验证@.+\.可以被search到即可

Python3 代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 import re
5 while True:
6     try:
7         address = input()
8         if not address:
9             break
10
11         if re.search('@.*@',address) or re.match('\s*[@.]',address) or
re.search('[@.]$',address) or re.search('@\.',address) or
re.search('\. @',address):
12             print('NO')
13         elif re.search('@.+\. ',address):
14             print('YES')
15         else:
16             print('NO')
17     except EOFError:
18         break
```

Python代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
import re
while True:
    try:
        address = input()
        if not address:
            break

        if re.search('@.*@', address) or re.match('\s*[@.]', address) or re.
            print('NO')
        elif re.search('@.+\. ', address):
            print('YES')
        else:
            print('NO')
    except EOFError:
        break
```

附录

如果设好了 typora 的图床，md 文件中图片在其他地方也能看见。因为图片存在云端。如果不好设置，注意导出的 pdf 文件包含图片。

Typora+PicGo+Github 解决个人博客图片上传问题 <https://zhuanlan.zhihu.com/p/367529569>