

2023年NOC复赛–Python初中组

1. 出租车费用

时间限制：1s

内存限制：128mb

(注：input()括号中不允许添加任何提示语)

根据乘坐出租车的里程，计算应该支付的费用。

出租车根据行驶里程阶梯式计价，具体规则如下表：

	行驶里程范围	单价
第一阶梯	10公里及以下	2元/公里
第二阶梯	11–20公里	1.5元/公里
第三阶梯	21公里及以上	1元/公里

请补全下面程序，使程序实现如下功能：

- (1) 程序开始运行后，输入一个正整数，表示乘坐出租车的里程数；
- (2) 根据规则计算并输出乘坐出租车应该支付的费用，结果保留一位小数。

```
1 d = int(input())
2 if _____:
3     cost = d * 2
4     _____
5     cost = _____
6 else:
7     cost = _____
8 print('%.1f' % cost)
```

参考代码：

```
1
2
3
4
5
6
7
8
```

2. 和的结果数

时间限制：1s

内存限制：128mb

(注：input()括号中不允许添加任何提示语)

请补全下面程序，使程序实现如下功能：

- (1) 输入n ($2 \leq n \leq 100$) 个整数，整数之间以空格隔开，并保证n个数各不相同；
- (2) 从这n个数中任选两个，计算并输出任意两个数的和共有多少种不同结果。

例如：

输入一组整数1 3 4 6，任意两数和可能为4 (1+3)、5 (1+4)、7 (1+6 或 3+4)、9 (3+6)、10 (4+6)，因此，输出结果应为5。

```
1 ls = [int(i) for i in input().split()]
2 results = []
3 for i in range(_____):
4     for j in range(_____):
5         s = ls[i] + ls[j]
6         if s not in results:
7             _____
8
9 print(len(results))
```

参考代码：

```
1
2
3
4
5
6
7
8
9
```

3. 因数和个数

时间限制：1s

内存限制：128mb

(注：input()括号中不允许添加任何提示语)

把一个数的因数全加在一起，叫做这个数的因数和。例如，6的因数有1、2、3、6，则它的因数和为 $1 + 2 + 3 + 6 = 12$ 。

请编写一段程序，输入一个正整数n ($n \leq 1000$)，判断因数和为n的正整数有多少个，并依次输出这些正整数。

输入描述:

输入一个正整数n ($n \leq 1000$)

输出描述:

第一行输出一个正整数, 表示因数和为n的正整数的数量, 如不存在, 则输出0

如果存在, 则第二行由小到大输出所有符合条件的数, 数字之间以空格隔开

输入样例1:

12

输出样例1:

2

6 11

样例说明1:

因数和为12的正整数有2个, 分别是6和11

输入样例2:

10

输出样例2:

0

样例说明2:

因数和为10的正整数不存在, 所以仅输出0

参考代码:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22
```

4. 单词变复数

时间限制：1s

内存限制：128mb

(注：input()括号中不允许添加任何提示语)

英语单词在变成复数形式时，有以下几种常见情况（不完全）：

- (1) 常规情况下结尾直接加s；
- (2) 以s、sh、ch、x结尾的单词，加es；
- (3) 以辅音字母加上y结尾的单词，去掉y加ies；

(注：英文字母中，除了a、e、i、o、u这5个元音字母外，其他都是辅音字母)

请编写一段程序，输入n ($1 \leq n \leq 1000$) 个英文单词（单数，仅包含小写字母），并按以上规则变成复数形式。

输入描述：

输入n ($1 \leq n \leq 1000$) 个英文单词（单数，仅包含小写字母），单词之间以空格隔开

输出描述：

输出这组单词按以上规则变成的复数形式，单词之间以空格隔开

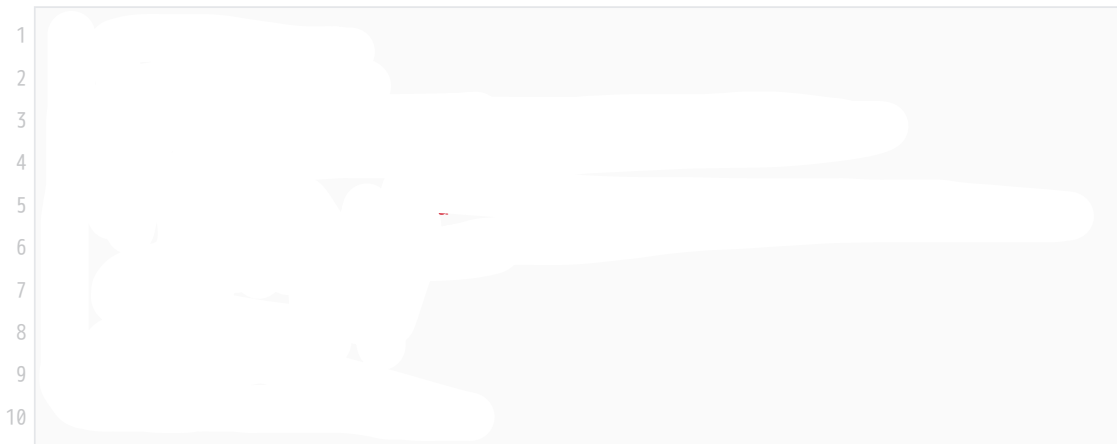
输入样例：

teacher box butterfly

输出样例：

teachers boxes butterflies

参考代码：



5. 啤酒兑换

时间限制：1s

内存限制：128mb

(注：input()括号中不允许添加任何提示语)

某啤酒品牌正在举办一次促销优惠活动。凭3个啤酒瓶可以再换一瓶啤酒，凭5个瓶盖也可以再换一瓶啤酒，换来的啤酒可以继续换，但不允许赊账。

请编写一段程序，计算一个人初始买入 n ($1 \leq n \leq 100$) 瓶啤酒后，最终他最多能得到多少瓶啤酒，并将这个结果输出。

输入描述：

输入一个正整数 n ，表示初始买入的啤酒瓶数 ($1 \leq n \leq 100$)

输出描述：

输出一个正整数，表示最终能得到的啤酒数

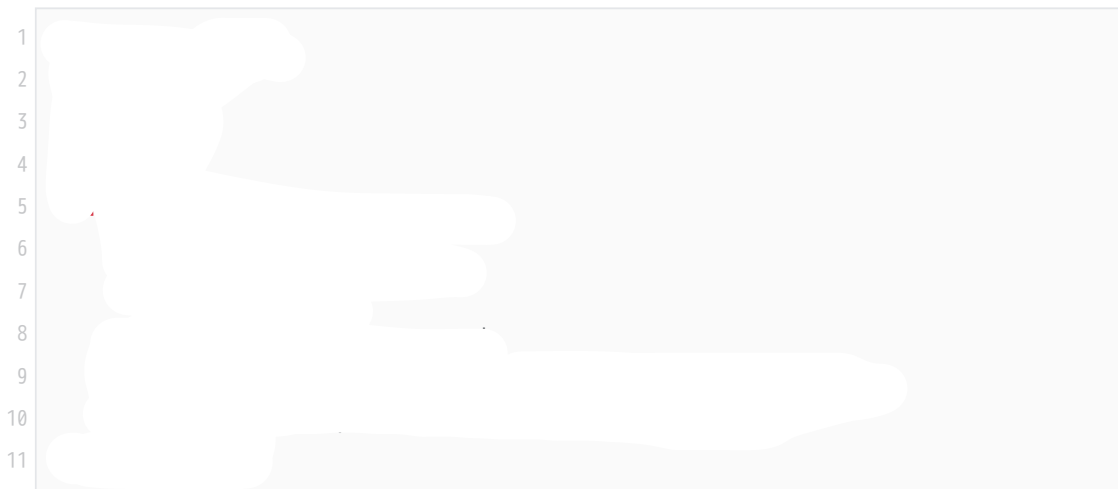
输入样例：

10

输出样例：

19

参考代码：



6. 反转递增串

时间限制： 1s

内存限制： 128mb

(注：input()括号中不允许添加任何提示语)

编写一段程序，输入一个 n ($2 \leq n \leq 1000$) 位正整数，将其中所有递增数字子串进行反转。

例如，正整数12387645包含两个递增数字子串1238和45，将这两部分进行反转，得到最终的83217654。

输入描述：

输入一个 n ($2 \leq n \leq 1000$) 位正整数

输出描述：

输出一行，为将输入中所有递增数字子串进行反转的结果

输入样例1：

12387645

输出样例1：

83217654

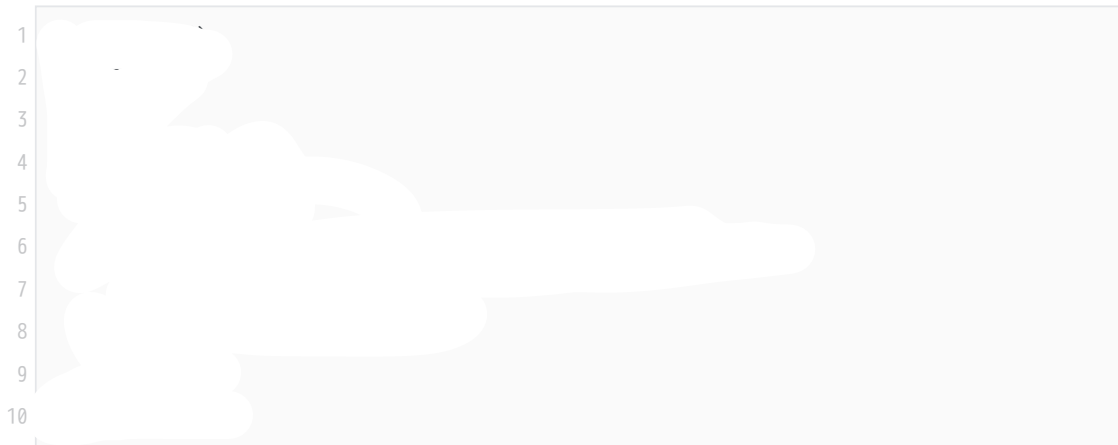
输入样例2:

87654321

输出样例2:

87654321

参考代码:



7. 包围黑子块

时间限制: 1s

内存限制: 128mb

(注: input()括号中不允许添加任何提示语)

小张发明了一种新式的棋类游戏——战斗棋。在“战斗棋”中，棋子分为黑、白两色。有一条重要的规则叫做“包围败地”。即，横竖相连的同色棋子算成一块棋，一块棋被对方棋子横竖包围起来就算一块“败地”，棋盘边角也算是包围。

现有一片战斗棋区域，包含 $N * M$ ($1 \leq N \leq 100$, $1 \leq M \leq 100$) 个落子点，所有落子点均摆满了棋子，其中1代表黑子、0代表白子。

请编写一段程序，计算并输出这片区域中被包围的黑色败地数量。

输入描述:

第一行输入2个正整数N和M ($1 \leq N \leq 100$, $1 \leq M \leq 100$)，N表示区域的行数，M表示区域的列数，正整数之间以一个空格隔开

接下来的N行每行包括M个数字 (数字只能为1或0)，1表示黑子，0表示白子，数字之间以一个空格隔开

输出描述:

输出一个正整数，表示 $N * M$ 的区域中被包围的黑色败地数量

输入样例:

5 3

1 0 1

0 1 0

0 1 0

1 0 1

0 0 0

输出样例:

5

参考代码:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17
```