

考试说明

1. 考试时间：合计 180 分钟，选择题部分不超过 60 分钟，允许提前交卷。
2. 考试过程中，不能连接未经指定网站或服务器。
3. 闭卷考试部分，不能查阅任何类型的参考资料。
4. 开卷考试部分，可以查阅纸质文档，不能查阅除 Python 编程环境自带帮助文件以外的任何类型的电子文档。
5. 考试过程中，不得使用任何形式的电子存储设备，不可使用手机。
6. 违反上述 2-5 条者，视为考试作弊。

选择题答题方式（25 分，闭卷，自动阅卷，严禁使用 python 编程环境进行尝试）

7. 打开浏览器，在地址栏中输入 <http://192.168.125.3>，点击相应链接进入登录页面。
8. 按要求输入两遍自己的学号。
9. 点击“登录”按钮即可进入答题页面。如考试尚未开始，系统会进入等待页面并倒计时。考试开始时间到，系统会自动进入答题页面。
10. 在页面左侧选择题号，页面右侧即会显示相应的题目。考生只需点击选择相应的选项。
11. 答题过程中如关闭浏览器或出现系统故障导致计算机重新启动，系统不会丢失之前已经完成的题目的答案。考生可以打开浏览器重新登录并继续考试。
12. 答题完成后，点击“交卷”按钮即可完成交卷。交卷后不能再次登录系统继续考试。
13. 考试结束时间到，系统会自动收卷。

编程题注意事项与提交方式（75 分，开卷，自动阅卷）

14. 下载 sample.py，保存到 D 盘。严禁将代码文件或数据文件保存在 C 盘中或桌面上。
15. 根据题目要求补全 sample.py 中的空函数。sample.py 中的已有函数的函数名、参数数量和顺序不可以修改。

16. 测试代码，请写入到 `if __name__=="__main__":` 部分，不要写到全局环境中。
17. 不要在代码中调用 `input()`，否则会导致阅卷失败。如阅卷失败，编程题部分不得分。
18. 每个函数中不需要使用 `print()` 输出结果，而是用 `return` 返回结果。如用 `print()` 输出结果将导致该函数不得分。
19. 不要使用关键字（`if`、`else`、`break`、`def` 等，会导致语法错误）作为自己的变量名，也不要使用内嵌名字（如 `list`、`int`、`input` 等，没有语法错误，但是很危险）作为自己的变量名，例如：`list = []`。
20. 代码中不要出现任何中文。
21. 提交的时候，确保程序可以正常运行，不要遗留任何语法错误。语法错误、程序异常等将导致阅卷失败。如阅卷失败，编程题部分不得分。
22. 所有脚本程序内容必须仅包含在一个脚本程序文件（`py` 文件）中。
23. 确保文件已经存盘，并知道文件存放的位置。提交前务必关闭 `IDLE` 或 `PyCharm` 编程环境。如文件丢失或无法找到，编程题部分不得分，不能重考。
24. 考试过程中，请经常进行存盘操作，以防突发异常情况而导致程序未保存。
25. 在浏览器的地址栏中输入 `http://192.168.125.3`，点击相应链接进入提交页面。
26. 按要求输入两遍自己的学号。
27. 点击“选择文件”按钮，选择自己的脚本程序文件。点击“提交”按钮提交。
28. 如提交成功，系统会显示相关信息。如果提交不成功，请重复步骤 25-27。
29. 提交成功后，可点击“查看内容”按钮检查提交的内容。
30. 编程题部分由阅卷系统自动批阅，以运行正确的测试用例数计分。不采取人工阅卷的方式。

答题方式示意图：

31. 请根据图 1、图 2 和图 3 所示的说明严格规范源文件结构。

```
1  #考试空文件
2
3  import math
4
5  def func2(a,b):
6      return
7
8  def func3(lst):
9      return
10
11 if __name__=="__main__":
12     pass
13
```

图 1 空文件样例

```
1  #考试标准答案文件
2
3  import math
4
5  def func2(a,b):
6      if a<=0 or b<=0:
7          return
8      if a>b:
9          a, b = b, a
10         s = 0
11         for i in range(a, b+1):
12             n = i
13             h = 0
14             while i>0:
15                 h = h * 10 + i % 10
16                 i = i // 10
17             if h == n:
18                 s = s + 1
19         return s
20
21
22 def func3(lst):
23     for i in range(len(lst)-1, -1, -1):
24         if lst[i]<0 or lst[i]%3==0:
25             lst.remove(lst[i])
26     lst.sort(reverse=True)
27     return lst
28
29
30 if __name__=="__main__":
31     print(func2(121,121))
32     print(func2(12221,12221))
33     print(func2(0,10))
34     print(func2(1,10))
35     print(func2(2,100))
36
37     print(func3([7,123,1,2,3,-1,66]))
38     print(func3([]))
39     print(func3([0]))
```

每道题目的代码包含在一个确定名称的函数体内。函数中不包含 input 函数和 print 函数。

测试代码写在此处，且全部包含的 if 语句体内，即保持相同的缩进位置。

图 2 正确的答题文件格式

1	#考试错误答案文件	
2		
3	import math	
4		
5	def func2(a,b):	
6	if a<=0 or b<=0:	
7	return	
8	if a>b:	
9	a, b = b, a	
10	s = 0	
11	for i in range(a, b+1):	
12	n = i	
13	h = 0	
14	while i>0:	
15	h = h * 10 + i % 10	
16	i = i // 10	
17	if h == n:	
18	s = s + 1	
19	return s	
20		
21	print(func2(121,121))	错误!!! 不要在函数之间插入测试代码
22		
23	def func3(lst):	
24	for i in range(len(lst)-1, -1, -1):	
25	if lst[i]<0 or lst[i]%3==0:	
26	lst.remove(lst[i])	
27	lst.sort(reverse=True)	
28	return lst	
29		
30	print(func3([7,123,1,2,3,-1,66]))	错误!!! 不要在函数之间插入测试代码
31		
32		
33	if __name__=="__main__":	
34		
35	print(func2(12221,12221))	
36	print(func2(0,10))	
37	print(func2(1,10))	
38	print(func2(2,100))	
39	print(func3([]))	
40	print(func3([0]))	
41		
42	print(func2(2,100))	错误!!! 不要在 if 语句体外插入测试代码
43	print(func3([]))	

图 3 错误的答题文件格式

题目说明

1. 求一个十进制正整数的二进制形式中 1 的个数。例如：6 的二进制为 110，其中 1 的个数为 2。

相关说明	
输入条件	参数 num 是一个整数。
输出要求	如 num 是负数，返回 None，否则返回 1 的个数
其它要求	将代码写入函数 func1。

测试样例：

输入	返回	解释
6	2	6 的 2 进制为 110，其中两个 1
0	0	0 的 2 进制为 0，其中没有 1
15	4	15 的二进制为 1111，其中 4 个 1

2. 给定一个元素全部是整数的列表 lst，如果有一对数字（i，j），满足 lst[i]等于 lst[j] 并且 i 小于 j，那么 i 和 j 可以称为一个好数对，请返回 lst 的好数对的个数。

相关说明	
输入条件	参数输入一个合法
输出要求	返回一个整数
其它要求	将代码写入函数 func2。

测试样例：

输入	返回	解释
[1,2,3,1,1,3]	4	分别是(0, 3)、(0,4)、 (3,4)、 (2,5)
[1,1,1,1]	6	每组数字都是好数对
[1,2,3]	0	没有好数对

3. 给定一个列表，包含整数、字符串、浮点数三种类型的元素。不同类型之间比较的规则：字符串>浮点数>整数；同类型元素之间则正常比较。请按照大小规则对列表从大到小排序，返回排序后的列表。

相关说明	
输入条件	列表，包含整数、字符串、浮点数三种类型元素。
输出要求	按照规则返回排序后的列表。
其它要求	将代码写入函数 <code>func3</code>

测试样例：

输入	返回
<code>['python', 2.3, 3.7, 9, 3, 'apple']</code>	<code>['python', 'apple', 3.7, 2.3, 9, 3]</code>
<code>[3, 'python', 1.1]</code>	<code>['python', 1.1, 3]</code>

4. 给定一个字符串，单词与单词之间以空格划分（不含标点符号,区分大小写），请用字典统计文本中各个单词出现的次数，即字典的 key 为字符串中的单词，value 为该单词出现的次数，返回该字典。

相关说明	
输入条件	字符串文本，单词与单词之间以空格间隔，不包含标点。
输出要求	统计各个单词出现的次数，返回字典。
其它要求	将代码写入函数 <code>func4</code>

测试样例：

输入	返回
<code>'Happy I am happy are you happy'</code>	<code>{'Happy': 1, 'I': 1, 'am': 1, 'happy': 2, 'are': 1, 'you': 1}</code>
<code>''</code>	<code>{}</code>

5. 给定一个字符串和一个正整数 n ，字符串只包含英文字母或数字，如果 n 大于等于 10，则先让 n 对 10 取余赋值给 n ，然后将字符串中的所有英文字母转换成其后面第 n 个字符。如果大写字母后面的第 n 个字符超过了大写字母 'Z'，则从大写 'A' 接着计数，如果小写字母后面的第 n 个字符超过了小写字母 'z'，则从小写 'a' 接着计数，保证转换后的字符仍然是英文字母。例如：当 n 为 5 时，则 'V' -> 'A'，'W' -> 'B'，'X' -> 'C'，'Y' -> 'D'，'Z' -> 'E'，'v' -> 'a'，'w' -> 'b'，'x' -> 'c'，'y' -> 'd'，'z' -> 'e'。

相关说明	
输入条件	字符串 s 和正整数 n
输出要求	返回转换后的字符串。
其它要求	将代码写入函数 func5

测试样例：

输入	返回
'vw1122xyz', 4	'za1122bcd'
'uvW1122xyz', 5	'zaB1122cde'
'abc1122def', 15	'fgh1122ijk'

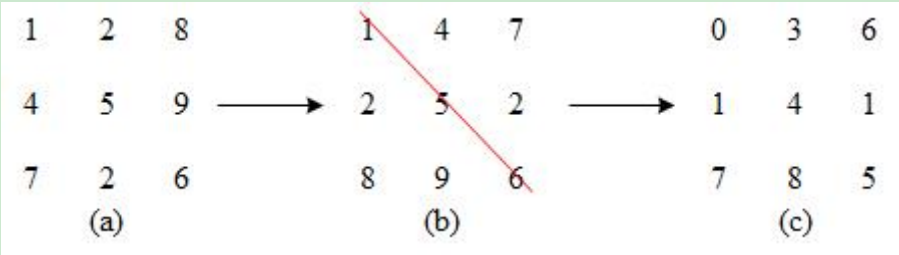
6. 给定一个包含若干电话号码的字符串，其中电话号码由 3 或 4 位区号和 7 或 8 位电话组成，提取所有电话号码组成列表并返回该列表。

相关说明	
输入条件	字符串 s ， s 中不会出现两个电话号码相邻出现
输出要求	返回电话号码组成的列表,如果字符串中没有符合要求的电话号码，则返回空列表
其它要求	将代码写入函数 func6

测试样例：

输入	返回
'vw1122xyz 0512-1234567 abc 010-12345678'	['0512-1234567', '010-12345678']
'vw1122 0512-123456 abc 010-12345678 a'	['010-12345678']
$s = \text{'vw1122 0512-123 abc 01-12345678 a'}$	[]
'vw1122 abc a'	[]

7. 给定一个二维列表 `lst`，该列表存储了如下图(a)所示的矩阵，以矩阵的对角线(下图红色线条)为轴进行翻转得到下图(b)，将图(b)中的所有元素减去对角线上的最小值即可得到图(c)所示的矩阵。最后，以二维列表的形式返回最终的矩阵。本例中图(a)与(b)对应的二维列表分别是`[[1,2,8],[4,5,9],[7,2,6]]`和`[[1,4,7],[2,5,2],[8,9,6]]`，对角线上的最小值是 1，返回结果对应的二维列表是`[[0,3,6],[1,4,1],[7,8,5]]`。



相关说明	
输入条件	矩阵中的元素都是整数，矩阵的行数与列数相等并且大于等于 2
输出要求	以二维列表的形式返回最终矩阵
其它要求	将代码写入函数 <code>func7</code>

测试样例：

输入	返回
<code>[[1,2,8],[4,5,9],[7,2,6]]</code>	<code>[[0,3,6],[1,4,1],[7,8,5]]</code>
<code>[[4, 5], [6, 7]]</code>	<code>[[0, 2], [1, 3]]</code>

8. 给定一个由元组构成的列表`L`，其中每个元组由 2 个 0 至 99 之间的整数元素构成。设有两个元组`t1`和`t2`，如果`t1`中的元素都不比`t2`中对应（同一位置）的元素大，且`t1`中至少存在一个元素比`t2`中对应元素小，则称`t1`支配`t2`。保留`L`中所有不被任何元组支配的元组，并将`L`按元组第 1 个元素从小到大的顺序排列。返回排序后的列表`L`。

相关说明	
输入条件	列表 <code>L</code> 一定是满足题意的合法输入
输出要求	返回列表 <code>L</code> 。
其它要求	将代码写入函数 <code>func8</code> 。

测试样例：

输入	返回
[(4, 2), (88, 21), (25, 27), (40, 72), (17, 33), (79, 14), (67, 66), (7, 18)]	[(4, 2)]
[(5, 94), (68, 73), (77, 73), (21, 14), (30, 81), (43, 60), (73, 68), (84, 61)]	[(5, 94), (21, 14)]
[(42, 11), (49, 46), (89, 33), (89, 96), (13, 1), (89, 11), (45, 39), (23, 5)]	[(13, 1)]