

## 考试说明

1. 考试时间：合计 150 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（20 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（80 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(1st):
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. 求一个正整数（大于零的整数）除了 1 和它本身以外的升序排列的因数列表。

相关说明	
输入条件	参数 n 是一个整数。
输出要求	除了 1 和它本身以外的升序排列的因数列表。如 n 不是正整数，返回空列表。
其它要求	将代码写入函数 func1。

测试用例：

输入	返回
61	[]
24	[2, 3, 4, 6, 8, 12]
0	[]

2. 求 a 和 b 两个正整数（大于零的整数）除了 1 以外的升序排列的公因数列表。

相关说明	
输入条件	参数 a 和 b 是整数。a 和 b 的大小关系未知。
输出要求	除了 1 以外的升序排列的公因数列表。如 a 或 b 不是正整数，返回空列表。
其它要求	将代码写入函数 func2。

测试用例：

输入	返回
3, 61	[]
40, 120	[2, 4, 5, 8, 10, 20, 40]
12, 18	[2, 3, 6]

3. 求列表的中位数。中位数是按升序或降序排列的一组数据中居于中间位置的数。如有偶数个数据，则取中间两个数的平均值的取整值为中位数。

相关说明	
输入条件	参数 <code>lst</code> 是一个整数列表，元素个数未知，无序。
输出要求	如果 <code>lst</code> 是空列表，则返回 <code>None</code> ，否则返回题意要求的中位数。
其它要求	将代码写入函数 <code>func3</code> 。

测试用例：

输入	返回
[1, 3, 1]	1
[40, -1, 13, 79, 120]	40
[8, 6, 7, 10, -2, 101]	7

4. 求列表中有几个值有重复值。

相关说明	
输入条件	参数 <code>lst</code> 是一个列表，元素个数未知。
输出要求	列表中有重复值的值的个数。
其它要求	将代码写入函数 <code>func4</code> 。

测试用例：

输入	返回（括号内是注释，不是返回的结果）
[1, 1, 1]	1（1 有重复值）
[40, -1, 40, -1, -1, 120]	2（40 和-1 有重复值）
[8, 6, 7, 10, -2, 101]	0

5. 给定正整数 n，求[2,n)内的所有回文质数（既是质数又是回文数）构成的降序列表。

相关说明	
输入条件	参数 n 是一个整数。
输出要求	[2, n)内的回文质数降序列表。如 n<2，则返回空列表。
其它要求	将代码写入函数 func5。

测试用例：

输入	返回
1	[]
11	[7, 5, 3, 2]
200	[191, 181, 151, 131, 101, 11, 7, 5, 3, 2]

6. 求一个集合中有多少个整数可以表达为该集合中另外两个整数的和。

相关说明	
输入条件	参数 s 是一个整数集合。
输出要求	可以表达为该集合中另外两个整数的和的整数个数。
其它要求	将代码写入函数 func6。

测试用例：

输入	返回（括号内是注释，不是返回的结果）
{1, 1, 1, 2}	0（输入集合中实际只有两个值）
{40, -1, 50, -1, 90, 120}	1（90=40+50）
{8, 6, 7, 10, -2, 13, 6, 14, 101}	4（8=10+(-2), 6=8+(-2), 13=6+7, 14=8+6）



7. 给定一个字符串，将字符串正中间的 4 个字符改成“\*\*\*\*”。如字符串长度为奇数，则扣除（不是删除）字符串尾部的 1 个字符后确定字符串正中间的位置。

相关说明	
输入条件	参数 s 是一个字符串。
输出要求	返回正中间的 4 个字符被改成“****”的字符串。如字符串长度小于 6 或大于 10，则返回“****”。
其它要求	将代码写入函数 func7。

测试用例：

输入	返回（括号内是注释，不是返回的结果）
"8671021314101"	"****"
"867102131"	"86****131"
"86710"	"****"

8. 已知一个等差数列缺失了一个值，求出这个缺失值。

相关说明	
输入条件	参数 lst 是一个未排序整数列表。缺失值一定不在数列的两端。
输出要求	返回一个缺失的值，可使得 lst 构成等差数列。如 lst 长度小于 4，则返回 None。
其它要求	将代码写入函数 func8。

测试用例：

输入	返回（括号内是注释，不是返回的结果）
[1, 2, 3, 4, 5, 8, 7, 9, 10]	6
[3, 4, 1, 2, 5, 6, 7, 9, 10]	8
[2, 4, 6, 8, 12]	10