

考试说明

1. 考试时间：合计 120 分钟，选择题部分不超过 30 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. 输入平面上两个点的坐标，计算并返回两点间的欧氏距离。

相关说明	
输入条件	参数 x_1, y_1 是第一个点的坐标。 x_2, y_2 是第二个点的坐标
输出要求	返回两点间的距离
其它要求	将代码写入函数 <code>func1</code>

测试用例：

输入	返回
0,0,0,0	0.0
0,0,3,4	5.0
0,0,1,1	1.4142135623730951

2. 输入一个三位正整数，返回其十位的数字，例如 123 的十位是 2。

相关说明	
输入条件	参数 <code>num</code> 一定是一个三位正整数
输出要求	返回的十位数一定是一位整数
其它要求	将代码写入函数 <code>func2</code>

测试用例：

输入	返回
123	2
951	5
555	5

3. 某产品的生产总成本可表示为： $y = x_1 + m \cdot x_2$ ；其中 x_1 为固定成本， x_2 为单位产品的可变成本。当生产产品数量 $m < 10000$ 时， $x_1 = 20000$ 元， $x_2 = 5$ 元。当生产产品数量 $m \geq 10000$ 时， $x_1 = 50000$ 元， $x_2 = 3$ 元。当生产产品数量为 0，生产总成本也为 0。给定生产产品数量 m ，编写一个函数计算产品的单位成本价格。

相关说明	
输入条件	m 是一个整数，不保证为正整数
输出要求	如果 m 不是正整数，则返回 None，否则返回单位成本价格
其它要求	将代码写入函数 func3

测试用例：

输入	返回
-1	None
0	None
9999	7.0002000200020005
10000	8.0

4. 给定三个整数 x,y,z 表示三角形的三条边，编写一个函数判断这三条边是否可以构成三角形？如果可以，并判断是等边三角形、直角三角形或普通三角形(非等边三角形和非直角三角形)？

相关说明	
输入条件	保证 x, y, z 是三个整数，不保证为正整数
输出要求	如果 x, y, z 中有一个不是正整数，则返回 None， 如果 x, y, z 不能构造三角形，则返回 None， 如果 x, y, z 构成等边三角形，则返回 1， 如果 x, y, z 构成直角三角形，则返回 2， 如果 x, y, z 构成普通三角形，则返回 3。
其它要求	将代码写入函数 func4

测试用例：

输入	返回
-1, 2, 3	None
1,2,3	None
1, 1, 1	1
3,4,5	2
3,7,9	3

5. 给定一个三位正整数 n 和两位正整数 m ，求 $[n, n + m]$ 范围内所有偶数的和。例如，给定 $n=150, m=10$ ， $[150, 160]$ 范围内的偶数有：150，152，154，156，158，160，这些偶数的和是 930。

相关说明	
输入条件	正整数 n 的范围是 $[100, 999]$ ，正整数 m 的范围是 $[10, 99]$
输出要求	返回符合条件的所有偶数的和
其它要求	将代码写入函数 <code>func5</code>

测试用例：

输入	返回
150,10	930
200,20	2310
150,30	2640

6. 给定一个两位的正整数 n ，如果除了 1 与 n 之外，该正整数有且仅有两个因子，则将该正整数定义为“半质数”。寻找 $[10, n]$ 范围内的所有“半质数”并求和，返回求和的结果。举例：给定 $n=16$ ， $[10, n]$ 范围内的半质数有：10，14，15，求和结果为 39。

相关说明	
输入条件	一个两位的正整数 n
输出要求	返回求和结果
其它要求	将代码写入函数 <code>func6</code>

测试用例

输入	返回
16	39
20	39
30	135

7. 给定一个整数列表，请将奇数统一放在前面，偶数统一放在后面，并且奇数之间的相对顺序不变，同样偶数之间的相对位置也不变，将重新排序的结果以列表形式返回。输入一定是满足要求的整数列表，无需做边界条件判断。

相关说明	
输入条件	整数列表
输出要求	返回列表，奇数在前，偶数在后，且奇数之间的相对顺序不变，偶数之间的相对顺序也不变
其它要求	将代码写入函数 <code>func7</code>

测试用例：

输入	返回
[3,0,7,8]	[3,7,0,8]
[1,9,7,4,3]	[1,9,7,3,4]
[]	[]

8. 给定一个整数列表，如果一个整数重复出现，则只保留第一次出现的整数。请将结果以列表形式返回。(注：输入都是满足要求的，不需要做边界条件判断)

相关说明	
输入条件	列表，且列表中元素都是整数。
输出要求	删除重复出现的整数，只保留第一个，不改变数字之间的相对顺序，结果以列表返回
其它要求	将代码写入函数 <code>func8</code>

测试用例：

输入	返回
[3,2,1,2,3,5]	[3,2,1,5]
[-3,0,8,8,3,0]	[-3,0,8,3]
[3,3,3,3,3]	[3]
[1,2,3]	[1,2,3]