

考试说明

1. 考试时间：合计 150 分钟，选择题部分 30 分钟交卷，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（30 分，闭卷，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（70 分，开卷）

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. 调用自己写的函数、自己测试的代码等，请写入到 `if __name__=="__main__":`，不要写到全局环境中。
17. 不要在全局环境中调用 `input()`，否则会导致阅卷失败
18. 每个函数中不需要使用 `print` 输出结果，而是用 `return` 返回结果。
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. 确保提交的时候，自己的程序可以正常运行，不要遗留任何语法错误。
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. 提交前务必关闭 `IDLE` 或 `PyCharm` 编程环境。
24. 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。
25. 按要求输入两遍自己的学号。
26. 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。
27. 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 24-26。
28. 提交成功后，可点击“查看内容”按钮检查提交的内容。

	A	B	C	D	E	F	G	H	I	J	K	L	M
ASCII 码值	65	66	67	68	69	70	71	72	73	74	75	76	77
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
ASCII 码值	78	79	80	81	82	83	84	85	86	87	88	89	90
	a	b	c	d	e	f	g	h	i	j	k	l	m
ASCII 码值	97	98	99	100	101	102	103	104	105	106	107	108	109
	n	o	p	q	r	s	t	u	v	w	x	y	z
ASCII 码值	110	111	112	113	114	115	116	117	118	119	120	121	122

英文字母 ASCII 码表(供参考)

答题方式示意图：

29. 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，且全部包含在 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))
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]))
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))
43 print(func3([]))
```

错误!!! 不要在函数之间插入测试代码

错误!!! 不要在函数之间插入测试代码

错误!!! 不要在 if 语言体外插入测试代码

图 3 错误的答题文件格式

题目说明

1. 质数和：给定一个小于 1000 的正整数 n ，找到 $(1,n]$ 范围内的所有质数，并求和。例如，给定正整数 11，在 $(1,11]$ 范围内的质数有 2，3，5，7，11，求和结果为 28。

相关说明	
输入条件	参数 n 是一个小于 1000 的正整数。
输出要求	返回求和结果
其它要求	将代码写入函数 <code>func1</code>

建议测试用例：

输入	返回
11	28
13	41

2. 对称字符串：给定一个由数字和英文字母组成的字符串，如果首尾字符依次相同，则该字符串就是一个对称字符串。例如，字符串'abccba'，'3cbc3'。其中，在判断字符是否相同的过程中，需要区分字母的大小写。例如，'ABCcba'为非对称字符串。

相关说明	
输入条件	给定的字符串由数字和英文字母组成且长度大于 1
输出要求	如果给定字符串是对称的，返回 <code>True</code> 如果给定字符串不对称，返回 <code>False</code>
其它要求	将代码写入函数 <code>func2</code>

建议测试用例：

输入	返回
'efNNfe'	<code>True</code>
'345base'	<code>False</code>

3. 四叶玫瑰数：给定一个四位正整数 n ，如果它的每个位上的数字的 4 次幂之和等于它本身，则 n 是一个四叶玫瑰数。

相关说明	
输入条件	参数 n 是一个 4 位的正整数
输出要求	如果 n 是四叶玫瑰数，则返回 True，否则返回 False。
其它要求	将代码写入函数 func3

建议测试用例：

输入	返回
1000	False
1634	True

4. 给定一个包含若干个正整数的列表 lst 和代表下标的正整数 k，要求判断：列表中处于 k 前面的元素有多少个小于 lst[k]；列表中处于 k 后面的元素有多少个大于 lst[k]。最后，以元组的形式返回这两个数。

例如，给定列表 lst=[15,2,6,1,18,20]和下标 k=2，下表 k 前面小于 lst[k]=6 的元素有 1 个，下标 k 后面大于 lst[k]的元素有 2 个。所以，返回(1,2)。

相关说明	
输入条件	列表长度大于 3，k 大于 0 小于列表长度
输出要求	返回一个元组，元组中的两个数顺序不能反。
其它要求	将代码写入函数 func4

建议测试用例：

输入	返回
lst = [15,6,3,7,8], k= 2	(0,2)
lst = [1,2,3,4,5], k=4	(4,0)

5. 空间点距离计算：给定半径 r 和一个列表 lst，列表中的每个元素是一个形如(x,y)的实数元组，判断列表中有多少个元素到原点(0,0)的距离小于半径 r。每个元素到原点的距离计算公式如下：距离= $\sqrt{x^2 + y^2}$ 。

相关说明	
输入条件	半径 r 是一个大于 0 的实数，列表长度大于等于 1
输出要求	返回到原点距离小于 r 的元素个数
其它要求	将代码写入函数 func5

建议测试用例：

输入	返回
r = 5, lst = [(1,2)]	1
r = 5, lst = [(8,-8), (9,10.5)]	0
r = 5, lst = [(1,1.5), (6,7.5),(-1.5,-1.5)]	2

6. 单词处理：给定由不同单词组成的列表 lst，每个单词都由英文字母组成，单词的 ASCII 码值定义为：单词中所有字符 ASCII 码值的算术平均。例如：单词‘hello’的 ASCII 码值为： $(104+101+108+108+111)/5 = 106.4$ 。根据单词的 ASCII 码值，对列表 lst 中的单词按从大到小的顺序进行排序。

相关说明	
输入条件	每个单词都至少包含一个字符，lst 的长度大于等于 1，不会出现单词 ASCII 码值相同的情况。
输出要求	排序后的单词列表
其它要求	将代码写入函数 func6

建议测试用例：

输入	返回
['I', 'love', 'China']	['love', 'China', 'I']
['is', 'OK']	['is', 'OK']

7. 统计数值个数：在一个整数列表中可能存在重复的数据。对该列表统计各整数出现的次数，并返回出现次数最少的 3 个整数的和，不足 3 个则返回全部整数的和。例如：[123,0,89,0,0,0,123,89,1,123]，其中出现次数最少的 3 个整数分别为 1、89、123， $1+89+123=213$ ，则返回 213

相关说明	
输入条件	由若干个整数组成的列表，其中各整数的出现次数肯定不相同
输出要求	返回出现次数最少的 3 个整数的和，不足 3 个则返回全部整数的和
其它要求	将代码写入函数 func7

建议测试用例：

输入	返回
[123,0,89,0,0,0,123,89,1,123]	213
[]	0
[1,1,1,1,1]	1

8. 统计相同字母：给定两个非空的、且全部由英文字母组成的单词字符串。统计出这两个单词中同时出现的字母，在统计过程中，大写字母和小写字母被认为是不同的字母；如某字母在单词中重复出现多次，则仅记为 1 个字母。将得到的满足条件的全部字母以列表形式进行保存，并按照其字典序（所谓字典序即为每个字母的 ASCII 码值的顺序）进行增序排序，然后以列表的形式进行返回。

相关说明	
输入条件	给定两个非空的、且全部由英文字母组成的单词字符串
输出要求	返回满足条件的字母列表。如不存在同时出现的字母，返回 None
其它要求	将代码写入函数 func8

建议测试用例：

输入	返回
‘China’, ‘Cap’	[‘C’,‘a’]