

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 (20 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（80 分，开卷）

14. Sample.py 中的函数名、参数数量和顺序不可以修改。
15. 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。
16. 不要在全局环境中调用 `input()`。
17. 每个函数中不需要 `print`，而是用 `return` 返回结果。
18. 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 = []`
19. 代码中不要出现任何中文。
20. 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。
21. 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。
22. 提交前务必关闭 IDLE 或 PyCharm 编程环境。
23. 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。
24. 按要求输入两遍自己的学号。
25. 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。
26. 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 16-18。
27. 提交成功后，可点击“查看内容”按钮检查提交的内容。

答题方式示意图：

28. 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 语言体外插入测试代码
43	print(func3([]))	

图 3 错误的答题文件格式

题目说明

1、给定整数 x ，计算如下函数的结果，返回结果值。

$$y = \begin{cases} 5 * x & x \geq 0 \\ 3 * |x| + 1 & x < 0 \end{cases}$$

相关说明	
输入条件	x 是一个整数
输出要求	仅返回 y 的计算结果
其它要求	将代码写入函数 func1

2、给定整数 x ，计算如下函数的结果，返回结果值。

$$y = \begin{cases} 10 & x \geq 10 \\ 8x^3 & 8 \leq x < 10 \\ 3x^2 & 3 \leq x < 8 \\ x + 1 & 0 \leq x < 3 \\ |x| & x < 0 \end{cases}$$

相关说明	
输入条件	x 是一个整数
输出要求	仅 y 的计算结果
其它要求	将代码写入函数 func2

3、计算非负整数 m 到 n （包括 m 和 n ）之间的所有奇数的和，其中， m 不大于 n 。例如 $m=3$ ， $n=12$ ，其和则为：3+5+7+9+11=35。

相关说明	
输入条件	能保证 m 和 n 都是非负整数
输出要求	输入不合法则返回 None，否则返回计算结果
其它要求	将代码写入函数 func3

4、请统计某个给定范围 $[m, n]$ 内所有非负整数中，数字 2 出现的次数。例如给定范围 $[2, 22]$ ，数字 2 在数 2 中出现了 1 次，在数 12 中出现 1 次，在数 20 中出现 1 次，在数 21 中出现 1 次，在数 22 中出现 2 次，所以数字 2 在该范围内一共出现了 6 次。

相关说明	
输入条件	m 和 n 都是整数，且 m 一定小于等于 n
输出要求	仅返回计算结果
其它要求	将代码写入函数 func4

5、求一个正整数的位数、所有各位数字的和、各位数字的最大值，结果以列表的形式进行返回，列表各元素的顺序同上。

相关说明	
输入条件	题目输入的数据保证是整数。
输出要求	以列表形式返回。列表中各元素的顺序分别是：位数、所有各位数字的和、各位数字的最大值。如输入数据不满足要求，返回 None
其它要求	将代码写入函数 func5。

6、给定 2 个正整数 m 和 n，将 m 的最高位数字加上 n。如相加和的结果大于等于 10 则取其和结果的个位数；如 m 小于 100 时则结果仍然为 m。例如，如 m=345，则当 n=6 时结果为 945，当 n=7 时结果为 45，当 n=8 时结果为 145。编写函数实现上述功能。

相关说明	
输入条件	题目输入的 2 个整数能保证是整数，但位数不确定。
输出要求	返回整数的计算结果。如输入数据不满足要求，返回 None
其它要求	将代码写入函数 func6

7、给定一个正整数 k 和一个整数列表 lst，将列表中前 k 个元素逆序。输出最后得到的列表。如果 k 大于列表长度，则将整个列表逆序。例如 k=4，输入列表是 [3, 4, 1, 5, 2]，那么返回列表是 [5, 1, 4, 3, 2]。

相关说明	
输入条件	k 是一个正整数，lst 是一个列表
输出要求	仅返回列表
其它要求	将代码写入函数 func7

8、给定一个数 v 和一个非负整数列表 lst，去除其中“密度”小于 v 的整数，并将列表中剩余元素按从大到小的顺序排列。输出最后得到的列表。其中一个非负整数的“密度”定义为该整数的各位数字之和除以该整数的位数。例如 123 的密度为 $(1+2+3)/3=2$ ，1234 的密度为 $(1+2+3+4)/4=2.5$ 。

相关说明	
输入条件	v 是一个实数，列表 lst 中的元素都是非负整数
输出要求	仅返回列表
其它要求	将代码写入函数 func8

参考代码：

```
def func1(x):
    if x >= 0:
        y = 5*x
    else:
        y = 3*abs(x)+1
    return y
```

```
def func2(x):
    if x >= 10:
        y = 10
    elif 8 <= x < 10:
        y = 8*x**3
    elif 3 <= x < 8:
        y = 3*x**2
    elif 0 <= x < 3:
        y = x-1
    else:
        y = abs(x)
    return y
```

```
def func3(m,n):
    if m <= n:
        num = list(range(m, n+1))
        sum = 0
        for i in range(len(num)):
            if num[i]%2 != 0:
                sum += num[i]
        return sum
    else:
        return None
```

```
def func4(m,n):
    num = list(range(m, n+1))
    sum = 0
    for i in range(len(num)):
        sum += list(map(int, list(str(num[i])))).count(2)
```



```

    return sum

def func5(num):
    if num > 0:
        numlist = list(map(int, list(str(num))))
        return [len(numlist),sum(numlist),max(numlist)]
    else:
        return None

def func6(m,n):
    if m>0 and n>0:
        if m<100:
            return m
        else:
            mlist = list(map(int, list(str(m))))
            mlist[0] = (mlist[0]+n)%10
            return int(''.join(list(map(str, mlist))))
    else:
        return None

def func7(k, lst):
    if k<=len(lst):
        lst[:k]=reversed(lst[:k])
    else:
        lst.reverse()
    return lst

def func8(v, lst):
    for i in range(len(lst)-1,-1,-1):
        d = sum(list(map(int, list(str(lst[i])))))/len(str(lst[i]))
        if d < v:
            del lst[i]
    lst.sort(reverse=True)
    return lst

if __name__=="__main__":
    pass

```