

## 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(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))
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 语句体外插入测试代码

图 3 错误的答题文件格式

## 题目说明

1. 给定两个整数，判定这两个整数是否互质。

| 相关说明 |  |
|------|--|
| 输入条件 | <ul style="list-style-type: none"><li>● 参数 a、b 是两个整数。</li><li>● 输入的两数确定是整数，无需考虑不是整数的情况。</li><li>● 两个整数的大小关系未知。</li></ul> |
| 输出要求 | <ul style="list-style-type: none"><li>● 如果两数互质，返回 True，否则返回 False。</li><li>● 如果 a、b 中任何一个数小于等于 0，则返回 False。</li></ul>    |
| 其它要求 | 将代码写入函数 func1。   |

测试用例：

| 输入     | 返回    |
|--------|-------|
| -4, -8 | False |
| 48, 14 | False |
| 55, 23 | True  |
| 1, 1   | True  |

2. 输入一个正整数 num，返回 num 的中间那位数字。

| 相关说明 |   |
|------|---|
| 输入条件 | <ul style="list-style-type: none"><li>● 参数 num 是一个整数。</li><li>● 输入参数确定是正整数，不用考虑非正整数的情况。</li></ul>                                     |
| 输出要求 | <ul style="list-style-type: none"><li>● 输出一位整数。不是输出字符串。</li><li>● 如果输入的整数有奇数位数字，则返回中间那位数。</li><li>● 如果输入的整数有偶数位数字，返回 False。</li></ul> |
| 其它要求 | 将代码写入函数 func2。  |

测试用例：

| 输入    | 返回    |
|-------|-------|
| 123   | 2     |
| 95134 | 1     |
| 5555  | False |

3. 奶茶店的奶茶 10 元一杯,现在促销买 5 杯送 3 杯,买 3 杯送 1 杯,现有  $n$  元, 计算最多可以买多少杯。

| 相关说明 |                  |
|------|------------------|
| 输入条件 | ● $n$ 一定是一个正整数。  |
| 输出要求 | ● 返回一个整数表示奶茶的杯数。 |
| 其它要求 | 将代码写入函数 func3    |

测试用例:

| 输入 | 返回 |
|----|----|
| 10 | 1  |
| 30 | 4  |
| 50 | 8  |
| 80 | 12 |

4. 给定三个整数  $r$ 、 $x$ 、 $y$  表示圆的半径和一个点的  $x$ 、 $y$  坐标, 判断这个点和以  $(0,0)$  为圆心且半径为  $r$  的圆的关系。

| 相关说明 |   |
|------|---|
| 输入条件 | ● 参数 $r$ 、 $x$ 、 $y$ 是三个整数。<br>● 不保证 $r$ 、 $x$ 、 $y$ 为正整数。  |
| 输出要求 | ● 如果 $r$ 、 $x$ 、 $y$ 中任何一个小于等于 0, 则返回 False。<br>● 如果点在圆内部, 则返回 1。<br>● 如果点在圆周上, 则返回 2。<br>● 如果点在圆外部, 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 | False |
| 5,3,4    | 2     |
| 6,6,6    | 3     |
| 10,6,6   | 1     |

5. 给定两个正整数  $n$  和  $m$ ，求  $n$  到  $m$  范围内（含  $n$  和  $m$ ）所有回文数的个数。所谓回文数就是将各位数字反向排列所得自然数与自己相等的数。

| 相关说明 |  |
|------|--|
| 输入条件 | <ul style="list-style-type: none"><li>● 参数 <math>n</math> 和 <math>m</math> 都是正整数。</li><li>● <math>n</math> 和 <math>m</math> 的大小关系未知。</li></ul> |
| 输出要求 | <ul style="list-style-type: none"><li>● 返回回文数的个数。</li></ul>  |
| 其它要求 | 将代码写入函数 <code>func5</code>   |

测试用例：

| 输入         | 返回  |
|------------|-----|
| 6718, 40   | 153 |
| 2852, 7056 | 43  |
| 100, 101   | 1   |

6. 给定一个正整数  $n$ ，求  $n!$  末尾有多少个连续的 0。

| 相关说明 |  |
|------|--|
| 输入条件 | <ul style="list-style-type: none"><li>● 参数 <math>n</math> 为正整数</li></ul>         |
| 输出要求 | <ul style="list-style-type: none"><li>● 返回 <math>n!</math> 末尾连续 0 的个数。</li></ul> |
| 其它要求 | 将代码写入函数 <code>func6</code>   |

测试用例：

| 输入  | 返回 |
|-----|----|
| 36  | 8  |
| 83  | 19 |
| 100 | 24 |



7. 给定一个正整数列表 `lst`，请将列表中元素重新排序。奇数集中存放在列表首部，偶数集中存放在列表尾部，奇数增序排列，偶数降序排列。返回一个重新排序列表。

| 相关说明 |   |
|------|---|
| 输入条件 | <ul style="list-style-type: none"> <li>● 参数 <code>lst</code> 是正整数列表。</li> <li>● 所有元素均为正整数，无需额外的检查。</li> </ul> |
| 输出要求 | <ul style="list-style-type: none"> <li>● 返回重新排序的列表。</li> <li>● 如果列表为空，返回空列表。</li> </ul>                       |
| 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]   | [3,7,8,0]   |
| [1,9,7,4,3] | [1,3,7,9,4] |
| []          | []          |

8. 给定一个整数列表，求其中出现次数最多的元素。

| 相关说明 |  |
|------|--|
| 输入条件 | <ul style="list-style-type: none"> <li>● 参数 <code>lst</code> 是整数列表。</li> <li>● 所有元素均为整数，无需额外的检查。</li> </ul>                          |
| 输出要求 | <ul style="list-style-type: none"> <li>● 如果只有一个元素出现次数最多，则返回一个整数，即为该元素。</li> <li>● 如果出现次数最多的元素有多个，则返回这些元素构成的列表。返回的列表按降序排列。</li> </ul> |
| 其它要求 | 将代码写入函数 <code>func8</code>   |

测试用例：

| 输入                                 | 返回                  |
|------------------------------------|---------------------|
| [60, 36, 1, 49, 35, 35]            | 35                  |
| [69, 69, 70, 2, 42, 69, 88, 91, 6] | 69                  |
| [15, 29, 69, 93, 3]                | [93, 69, 29, 15, 3] |
| [95, 95, 39, 12, 12, 23, 15, 15]   | [95, 15, 12]        |