

考试说明

1. 考试时间：合计 180 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（25 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（75 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. 给定两个包含任意字符的字符串 s1 和 s2，编写程序判断 s2 在 s1 中不重叠出现的次数。

相关说明	
输入条件	字符串 s1 和 s2
输出要求	返回 s2 在 s1 中不重叠出现的次数
其它要求	将代码写入函数 func1。

测试用例：

输入	返回
"abded", "ded"	1
"abded", "d"	2
"abdeded1#2%ded", "ded"	2
"d", "abded"	0

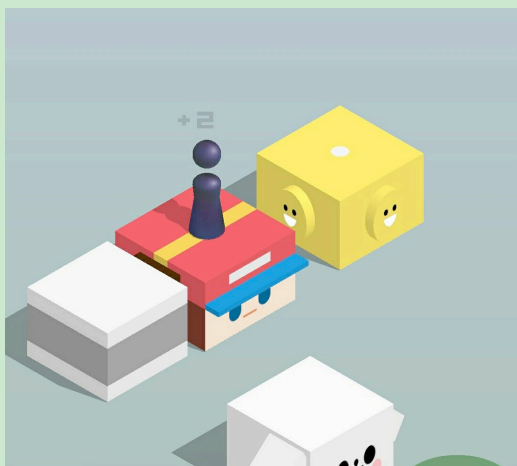
2. 给定两个正整数 n, m，计算在[n,m]区间内的双素数个数。所谓双素数是一个正整数是素数，其逆序整数也是素数，并且该正整数与其逆序整数不相等。

相关说明	
输入条件	● 两个正整数 n, m
输出要求	● 如果 n>m，则返回 0 ● 如果 n<m，则返回[n,m]区间内双素数的个数
其它要求	将代码写入函数 func2。

测试用例：

输入	返回
2,8	0
2, 13	1
11, 17	2
7,23	2

3. 有一个称为跳一跳的游戏，曾经非常风靡。



现在有一个简化后的跳一跳，玩家顺序向前跳跃，得分规则如下：

- 如果没有跳到下一个方块上则游戏结束；
- 如果跳到了方块上，但没有跳到方块的中心则获得 1 分；
- 如果跳到方块且在中心时，若上一次的得分为 1 分或这是本局游戏的第一次跳跃则此次得分为 2 分，否则此次得分比上一次得分多 2 分（即连续跳到方块中心时，总得分将+2，+4，+6，+8...）。

现在有一个列表 `lst` 给出一个玩家跳一跳的过程，请你求出他游戏的得分。`lst` 的元素的取值一定是整数 0、1、2 之一，具体意义如下：

- 0 表示此次跳跃没有跳到方块上（此时游戏结束）；
- 1 表示此次跳跃跳到了方块上但是没有跳到中心；
- 2 表示此次跳跃跳到了方块上并且跳到了方块中心；

相关说明	
输入条件	<code>lst</code> 的所有元素都是整型，取值一定是 0、1、2 之一
输出要求	返回一个整数，代表按照规则计算出的总分
其它要求	将代码写入函数 <code>func3</code> 。

测试用例：

输入	返回
<code>[1,1,2,2,2,1,1,2,2,0]</code>	22
<code>[0]</code>	0
<code>[2,2,2,2,2,2,2]</code>	56

4. 篮球三分球团体赛的规则为：每个参赛队由若干队员组成；所有队员独立比赛；参赛队的成绩为所有队员的成绩和；成绩最高的队获胜。现给定所有队员的比赛成绩，请你找出冠军队。

输入是一个元组 `tup`，它的每个元素是一个包含 3 个元素的元组，3 个元素都是整型，分别代表队伍编号、队员编号和成绩。

- 其中队伍编号为 1 到 1000 的正整数；
- 队员编号为 1 到 10 的正整数；
- 成绩为 0 到 100 的整数；
- 题目保证冠军队是唯一的。

相关说明	
输入条件	<code>tup</code> 是一个二维元组，其中每个数据的取值都是合法的
输出要求	以元组的形式返回结果，元组的第 1 个元素是队伍编号，第 2 个元素是成绩
其它要求	将代码写入函数 <code>func4</code>

测试用例：

输入	返回
<code>((2,1,5),(3,1,3),(4,1,3),(2,2,15),(8,1,18))</code>	<code>(2,20)</code>
<code>((8,1,5),(3,2,4))</code>	<code>(8,5)</code>
<code>((2,1,5),)</code>	<code>(2,5)</code>

5. 已知一个正整数集合，如果将其中的数按从小到大的顺序排列，是缺失一个数的等比数列。请求出这个缺失的值。

相关说明	
输入条件	<ul style="list-style-type: none"> ● 参数 <code>s</code> 是一个正整数集合。 ● 缺失值一定不在数列的两端。
输出要求	<ul style="list-style-type: none"> ● 返回一个缺失的值，使得 <code>s</code> 中的值能构成等比数列。

	<ul style="list-style-type: none"> 如集合少于 4 个元素，则返回-1。
其它要求	将代码写入函数 func5。

测试用例：

输入	返回
{1, 2, 4, 8, 32}	16
{2, 4, 8, 32, 64}	16
{3, 9, 81, 243}	27

6. 已知一个列表 lst 中存放了若干英文单词集合。请找出结尾字母频度最大的字母集合和每个字母所对应的频度。

相关说明	
输入条件	<ul style="list-style-type: none"> 参数 lst 是一个英文单词列表。 英文单词中可能包含小写字母、单个单引号、连字符（即减号）等符号。
输出要求	<ul style="list-style-type: none"> 返回一个元组，结构为 “({“字母 1”, “字母 2”, ..., “字母 k”}, 整数 m) ”。以“字母 1”、“字母 2”、...、“字母 k”结尾的单词都有 m 个。 如输入为空列表，则返回空列表。
其它要求	将代码写入函数 func6。

测试用例：

输入	返回
[]	[]
['hate', 'first', 'get', 'local', 'tool', 'consider', 'balanced', 'wireless']	({'l', 't'}, 2)
['products', 'mountains', 'little', 'supply', 'president's', 'effective', 'acquisition']	({'s'}, 3)

7. 指定范围的字符串访问。给定一个字符串 `strn` 和一个单词 `word` 以及一个由正整数组成的元组 `t=(n,m)`，判断下标 `n` 到 `m`(包含 `m`)范围内的字符串是否包含单词 `word`，若包含则返回 `True`，否则返回 `False`。举例：给定字符串 “I have a dream”，元组 `(1,7)`，单词 “have”，指定范围内的字符串包含该单词，因此返回 `True`。

相关说明	
输入条件	给定字符串 <code>strn</code> 的长度大于等于 2，字符串 <code>word</code> ，元组 <code>(n,m)</code> ，其中 <code>n</code> 小于等于 <code>m</code> ， <code>n</code> 与 <code>m</code> 都大于 0 且小于字符串长度
输出要求	若包含单词 <code>word</code> ，返回 <code>True</code> ；否则，返回 <code>False</code>
其它要求	将代码写入函数 <code>func7</code> 。

测试用例：

输入	返回
"panda", "an", (1, 2)	True
"a cat", "cat", (1, 2)	False

8. 给定形如 `<strn1>strn2</strn1>` 的 XML/HTML 标签，需要将该标签自动格式化为 `strn1:strn2`。在格式化的过程中需要将 `strn1` 中每个单词的首字母大写，其他字母小写，`strn2` 做同样的处理。举例，给定 `<su Zhou>beautiful city</su Zhou>`，输出为 `Su Zhou:Beautiful City`，给定标签 `<wu Han>grEat citY</wu Han>`，输出为 `Wu Han:Great City`。

相关说明	
输入条件	字符串 <code>strn</code> ，格式如题中说明，其中 <code>strn1</code> 与 <code>strn2</code> 中的单词由空格隔开，单词只包含字母， <code>strn1</code> 中的单词数量为 2， <code>strn2</code> 单词数量大于 2
输出要求	返回对应的字符串
其它要求	将代码写入函数 <code>func8</code> 。

测试用例：

输入	返回
<code><su Zhou>beautiful city</su Zhou></code>	<code>Su Zhou:Beautiful City</code>

<wu Han>grEat citY<wu Han>	Wu Han:Great City
----------------------------	-------------------