# Python选择题（240）

## 1-Python基础（40）

### 1-10题

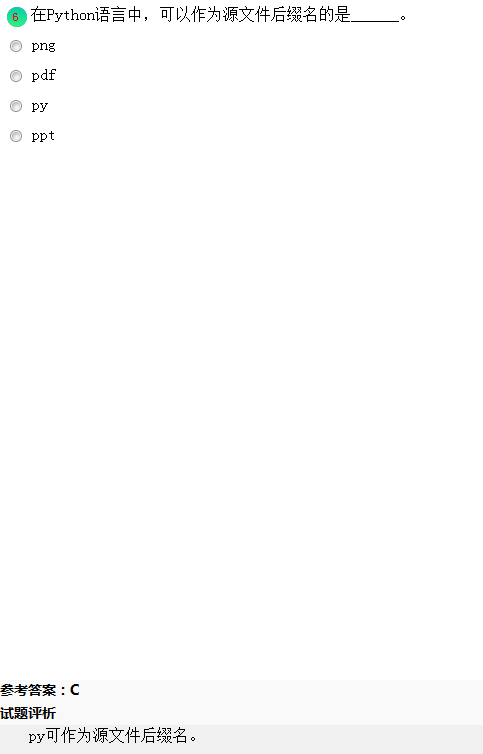


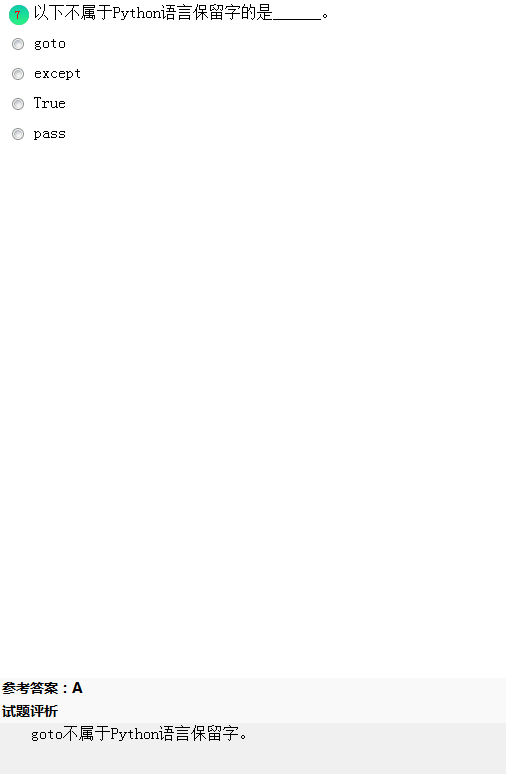






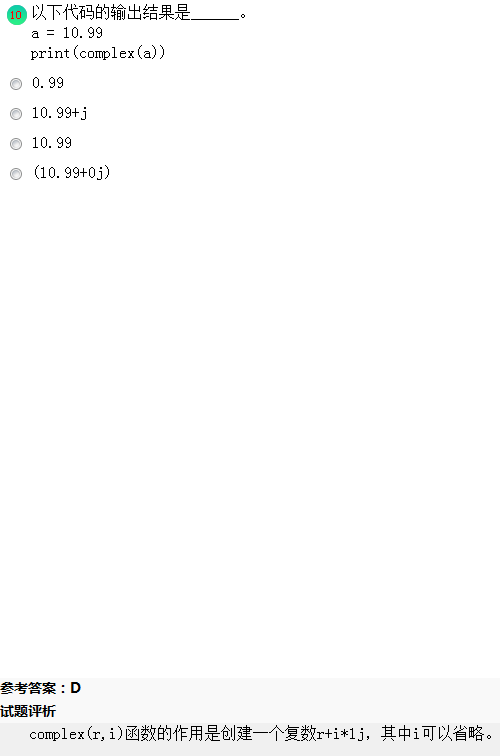






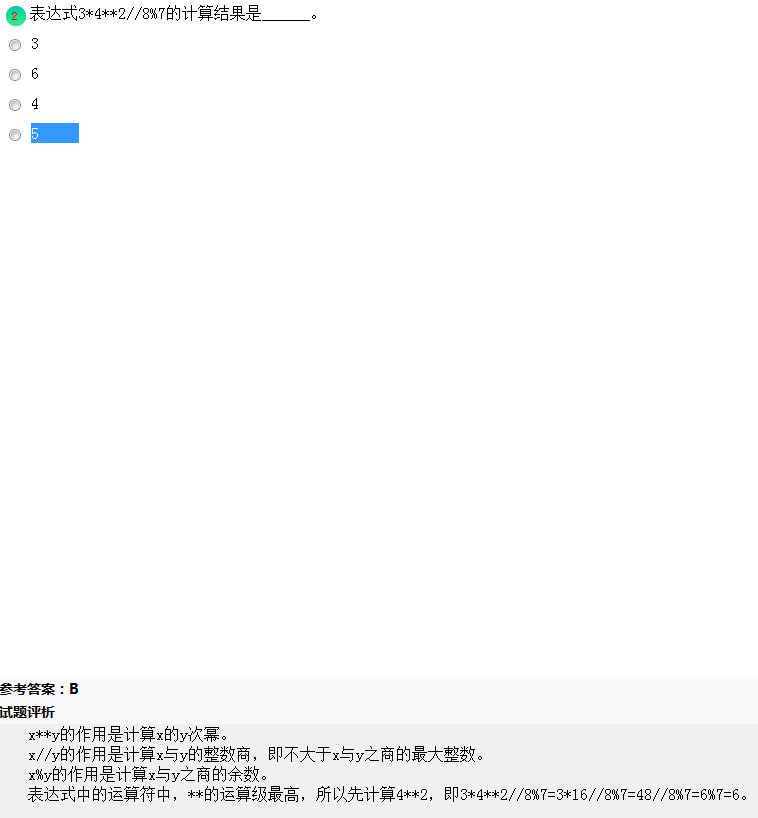




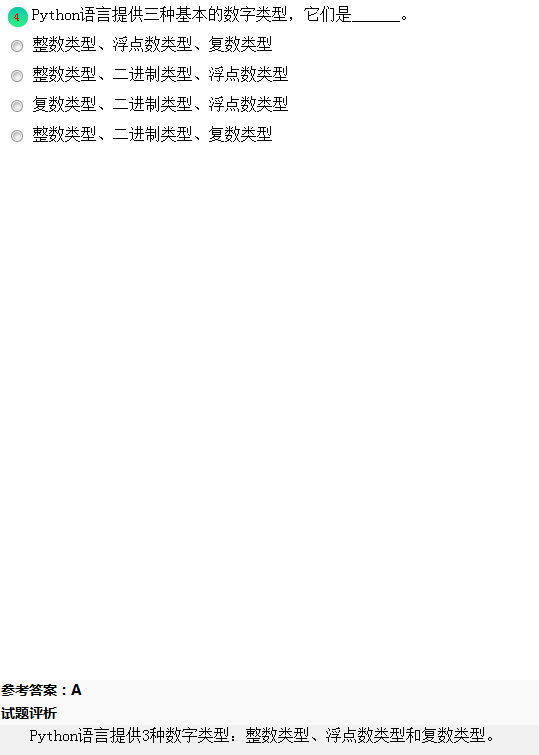


### 11-20题

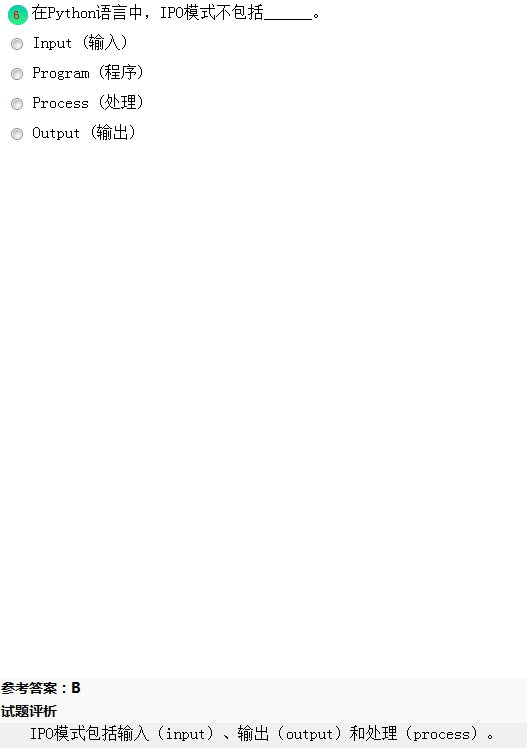
















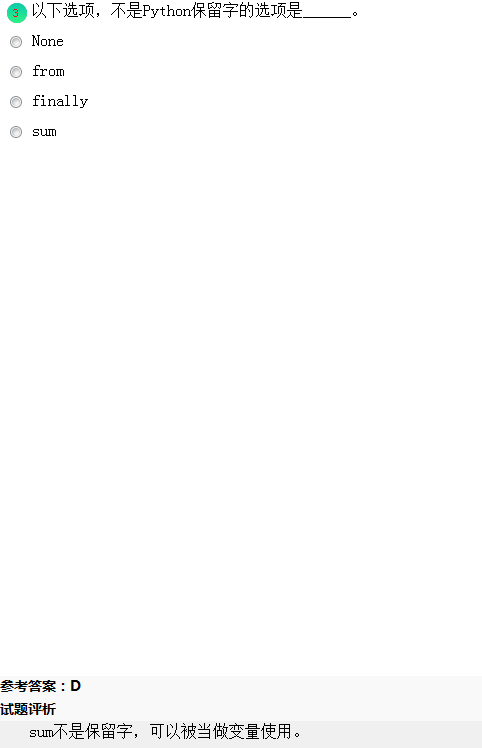




### 21-30题





















### 31-40题

xia



















## 2-Python控制（40）

### 1-10题



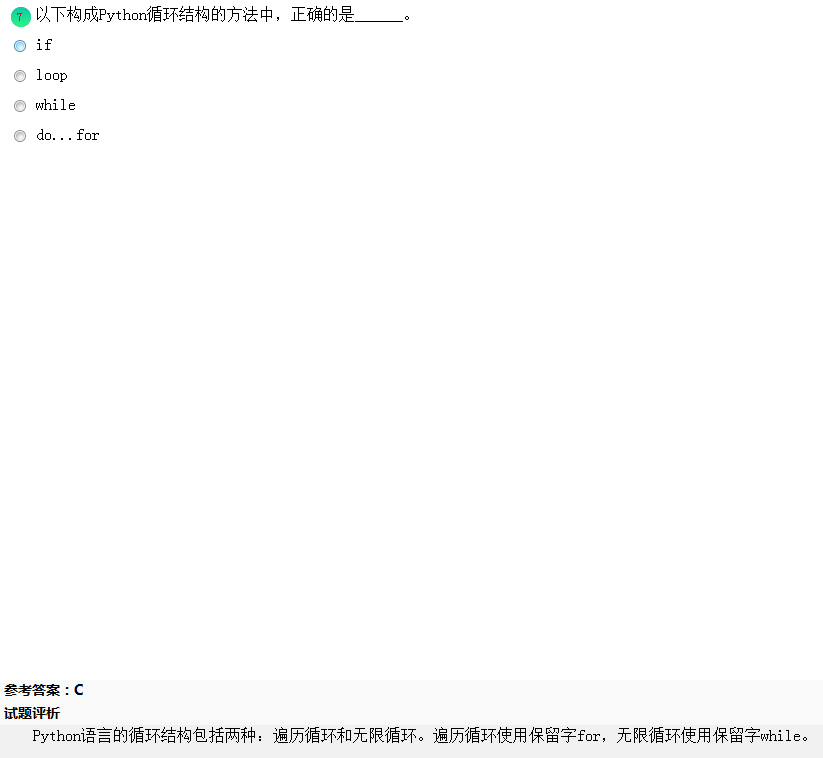




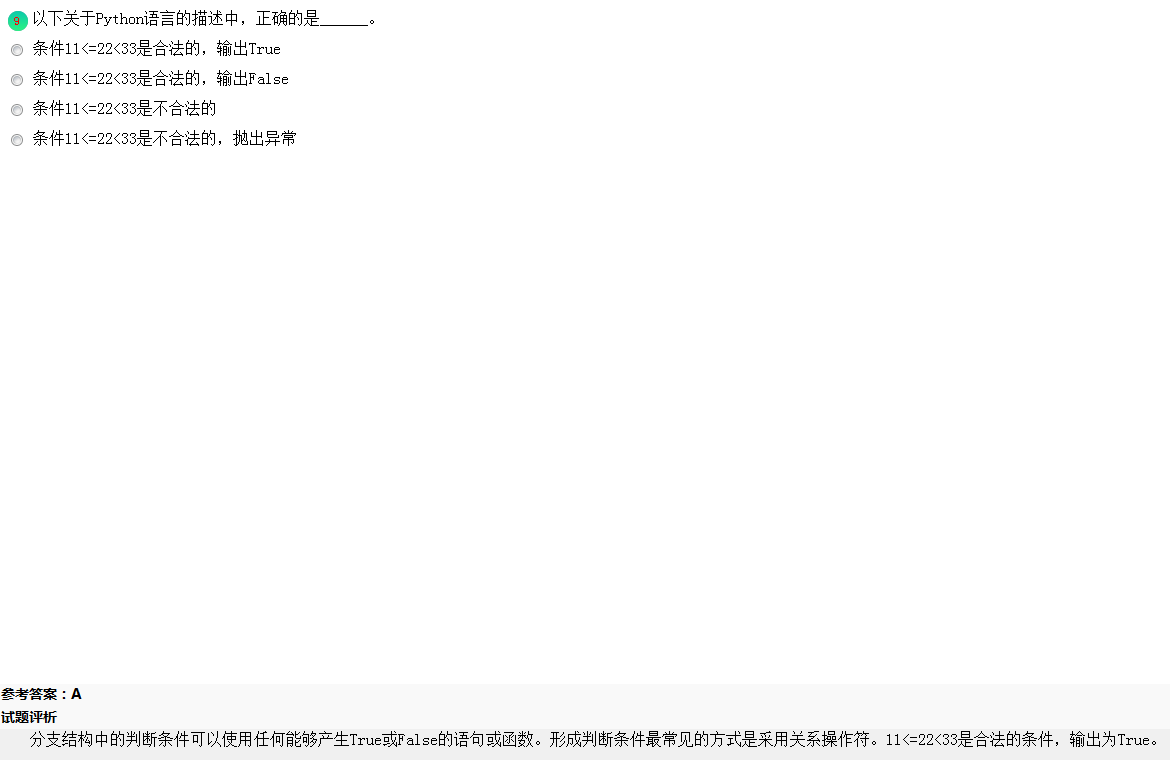














### 11-20题









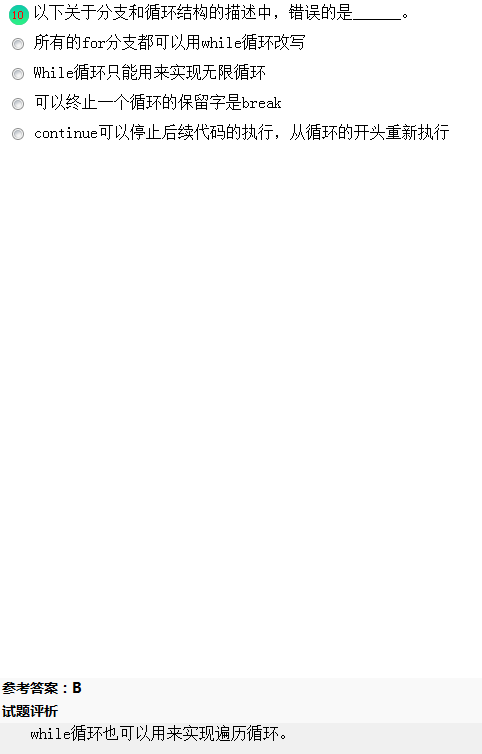




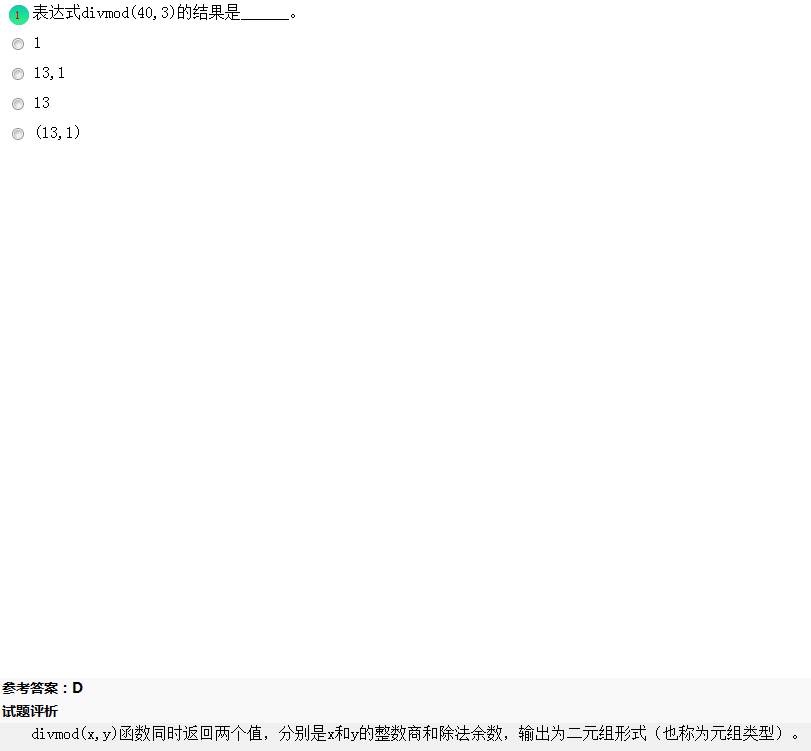






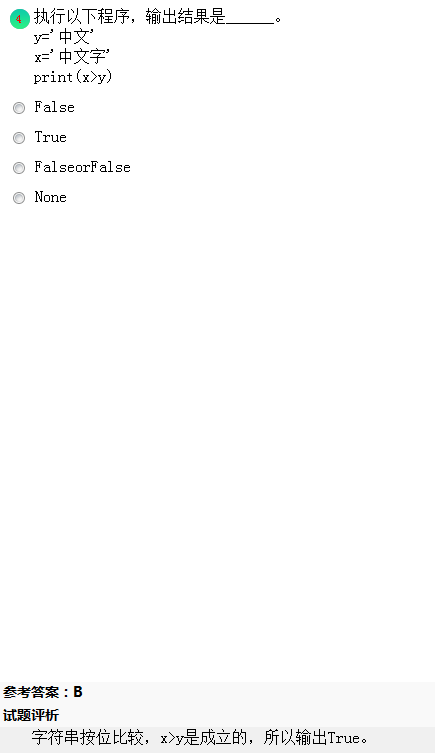


### 21-30题









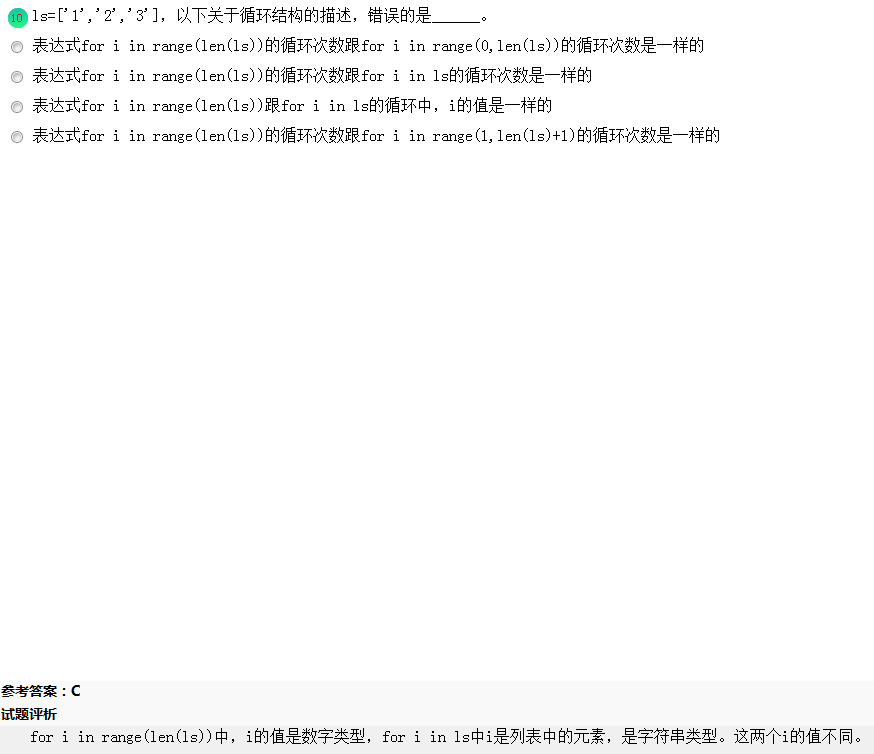










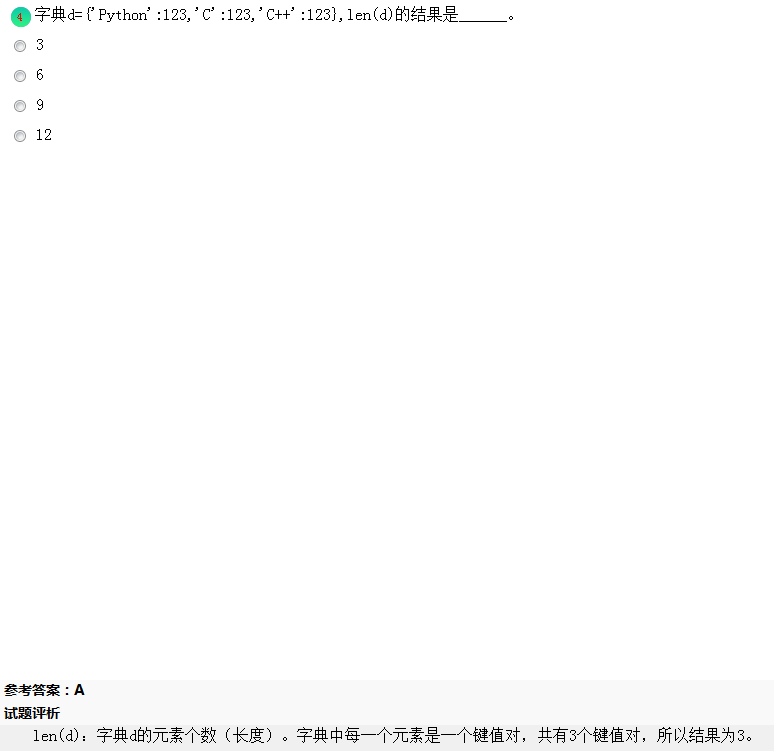


### 31-40题



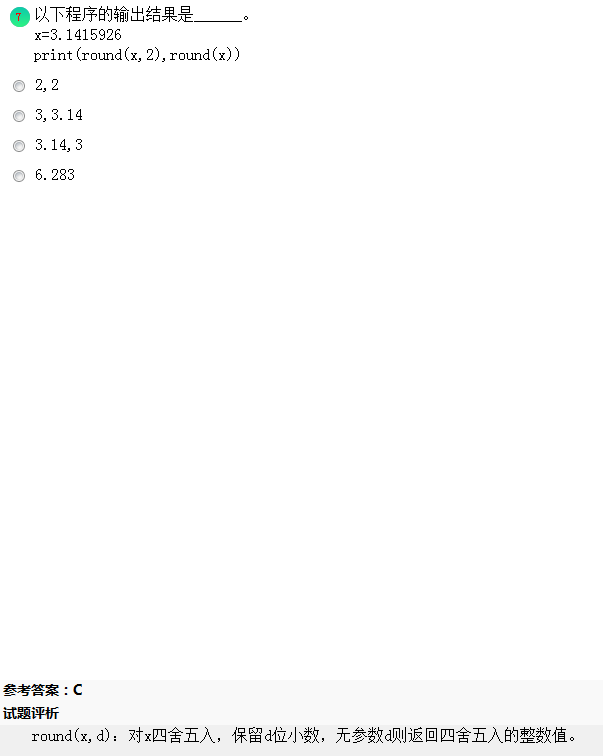


















## 3-Python函数（48）

### 1-10题



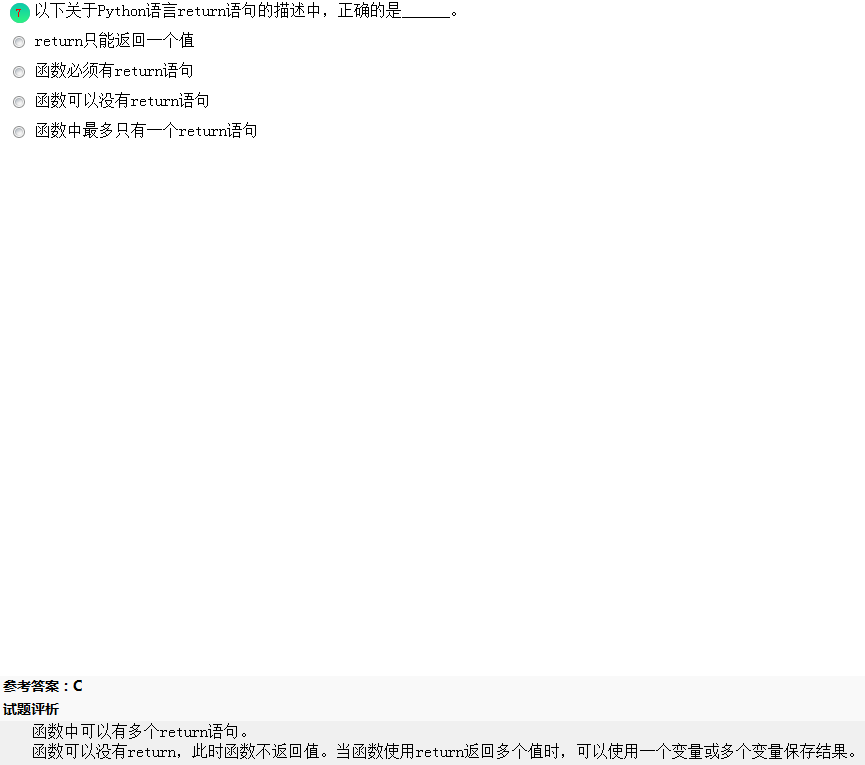










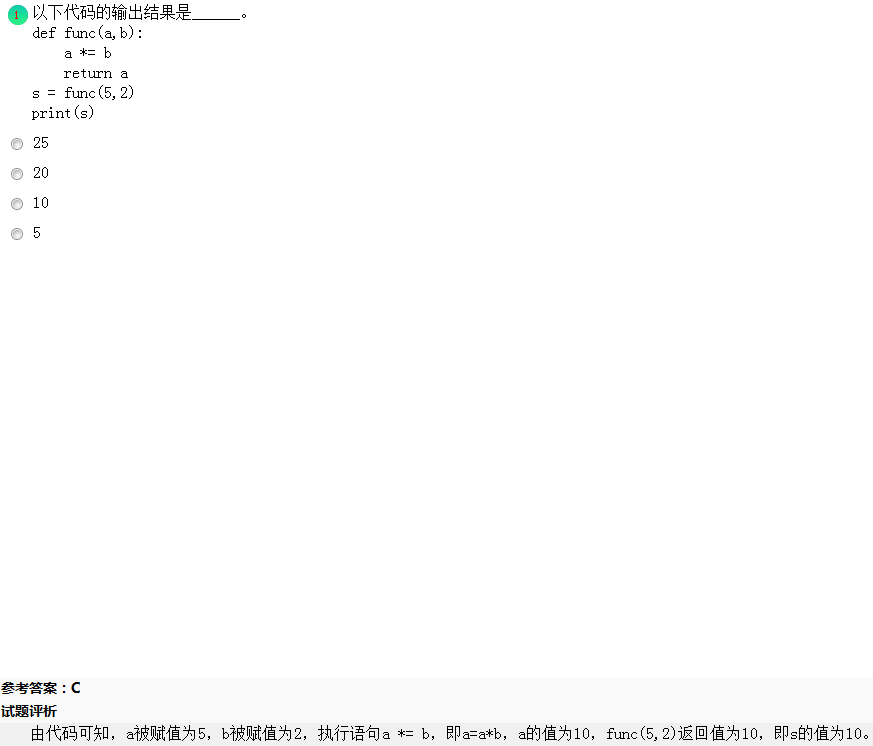


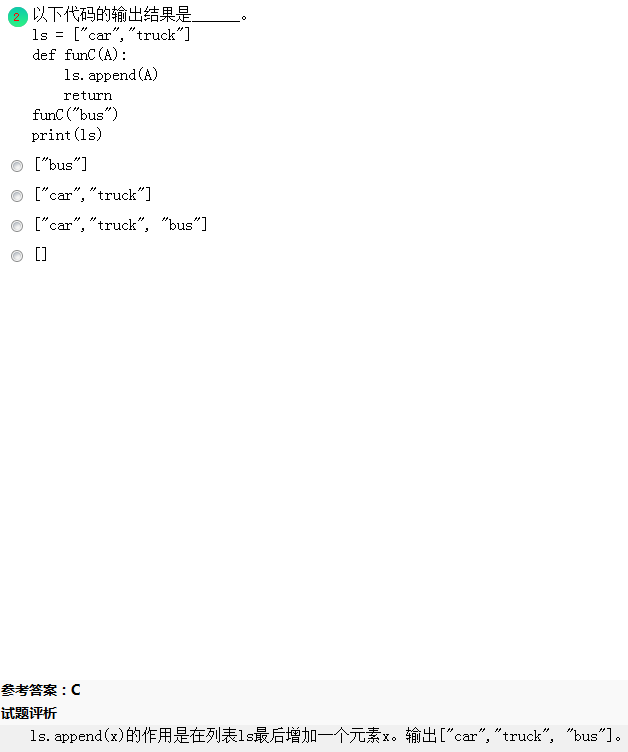






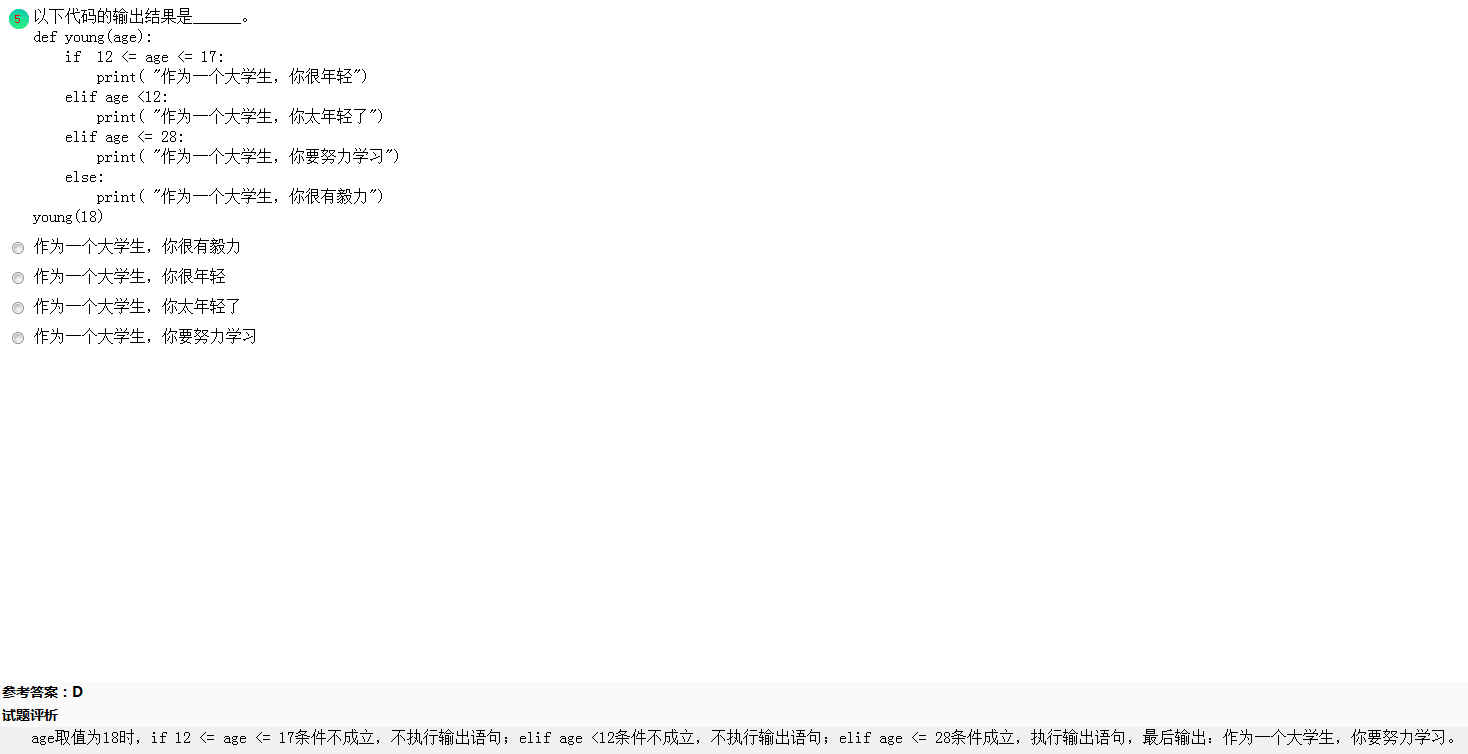
### 11-20题





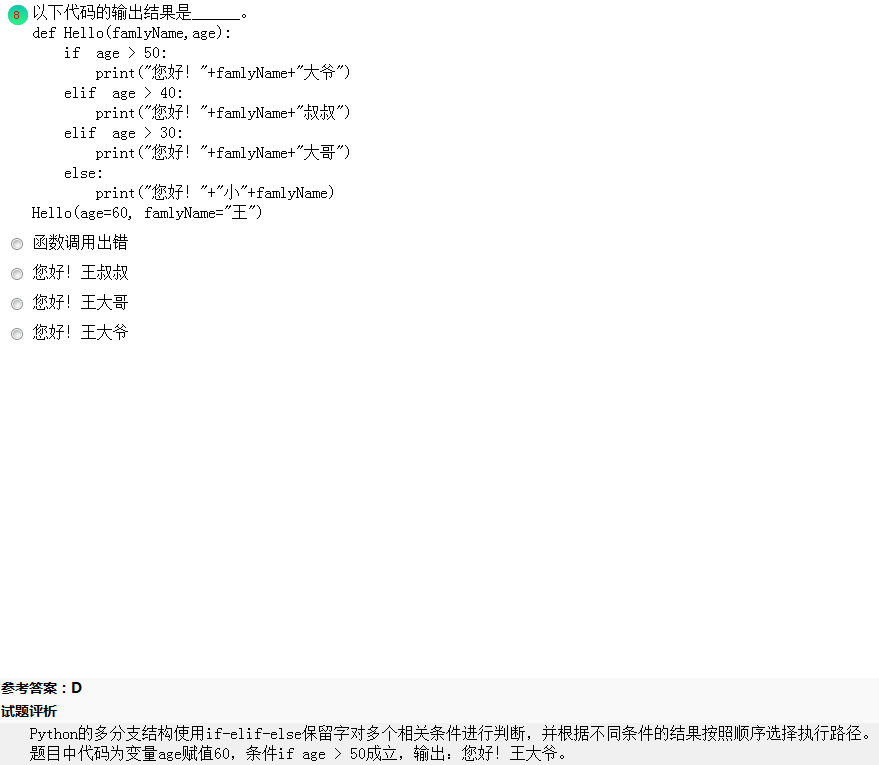










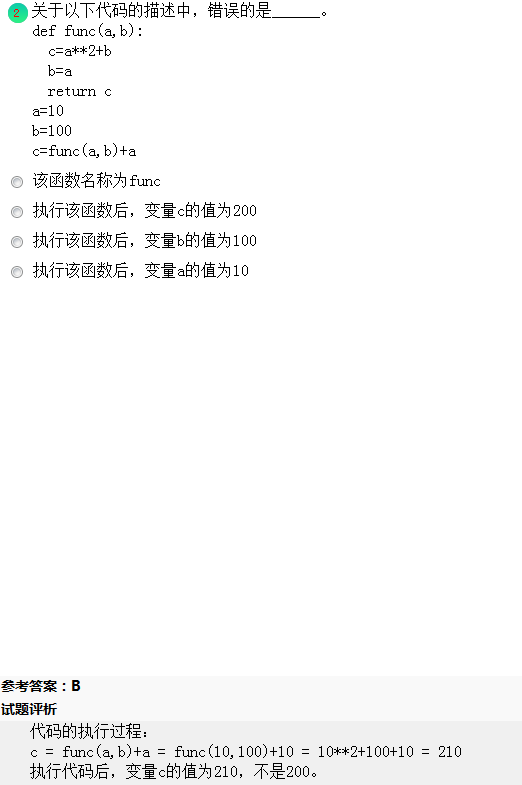






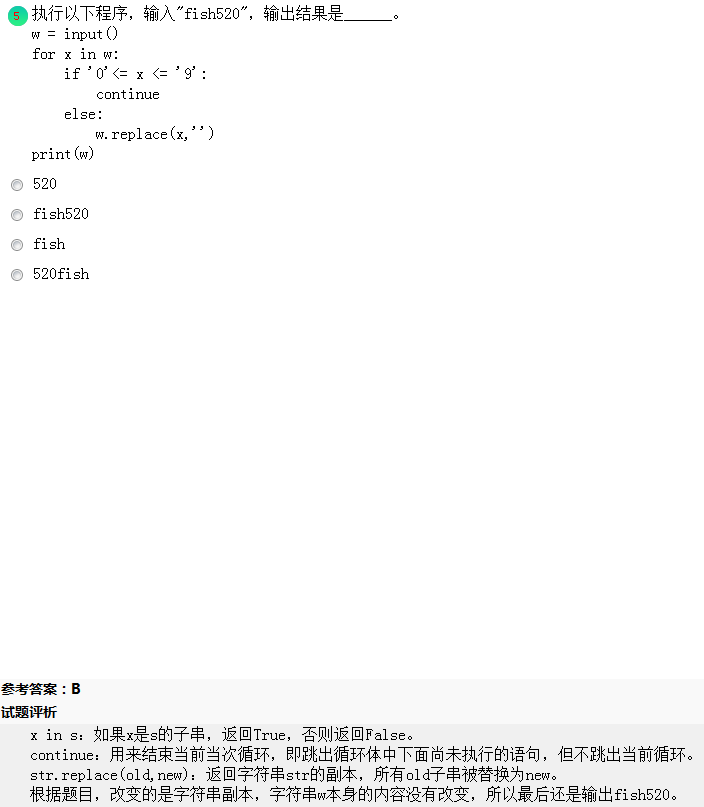
### 21-30题

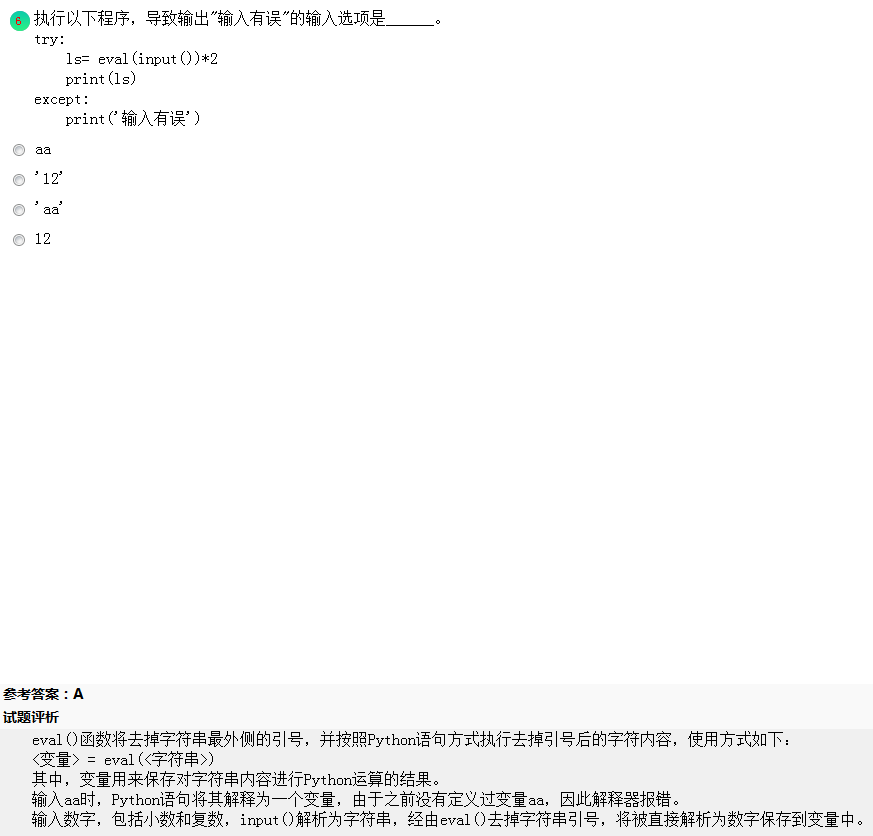




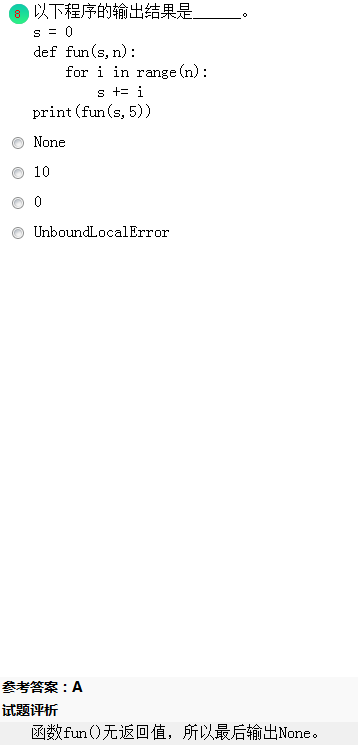




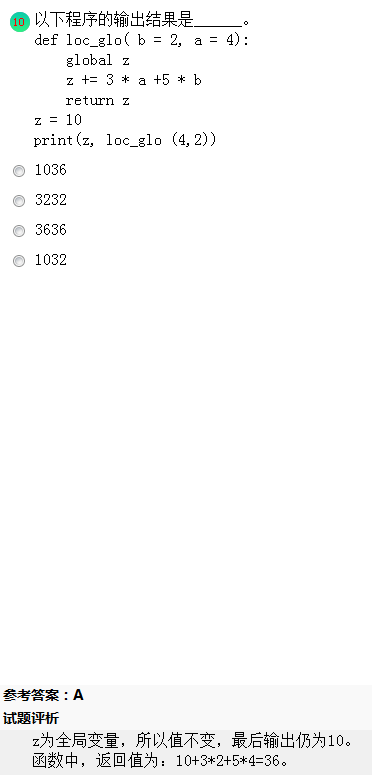










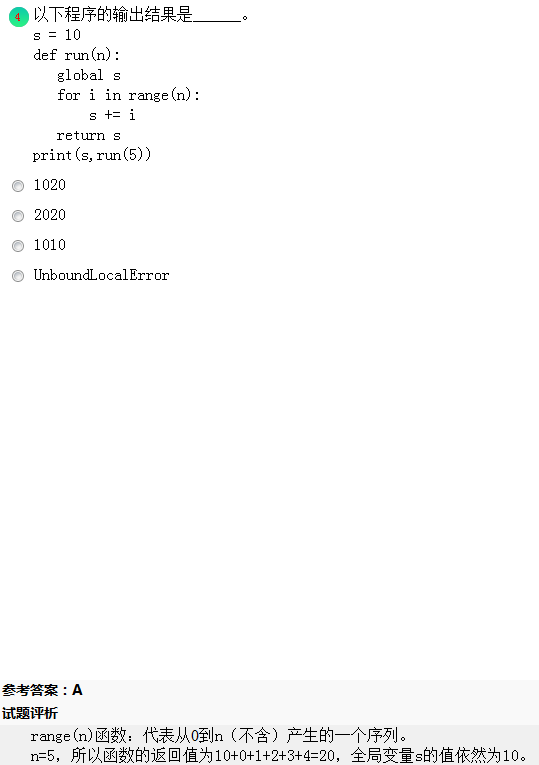


### 31-40题

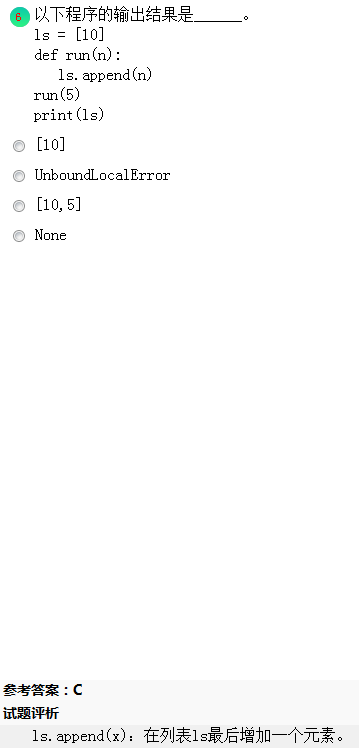


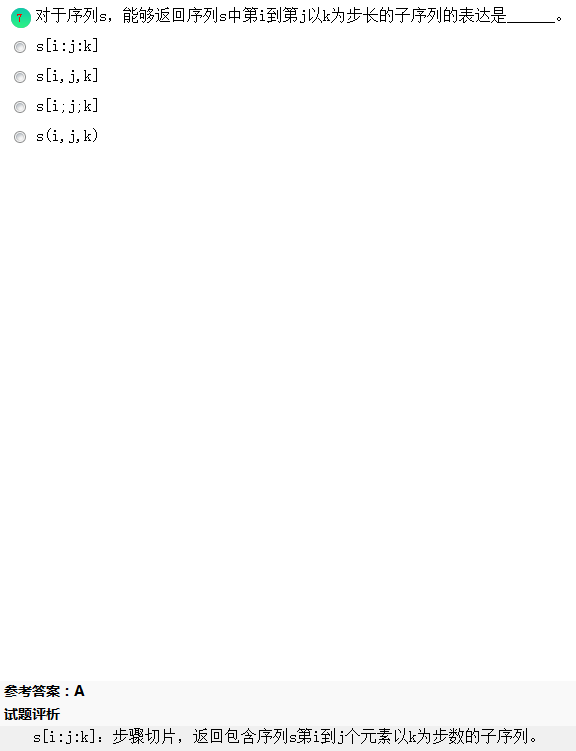


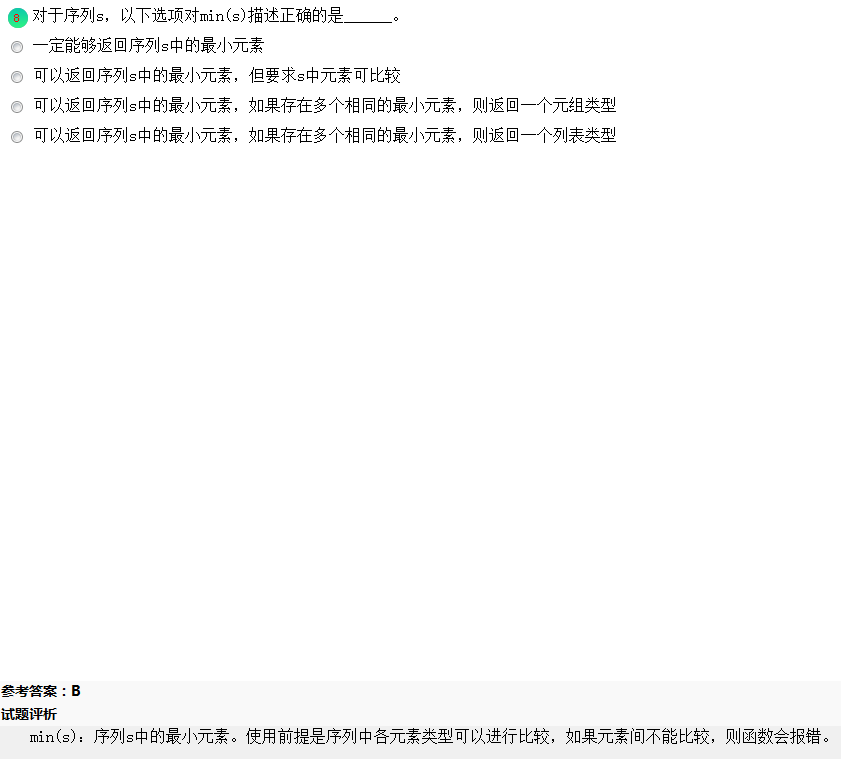


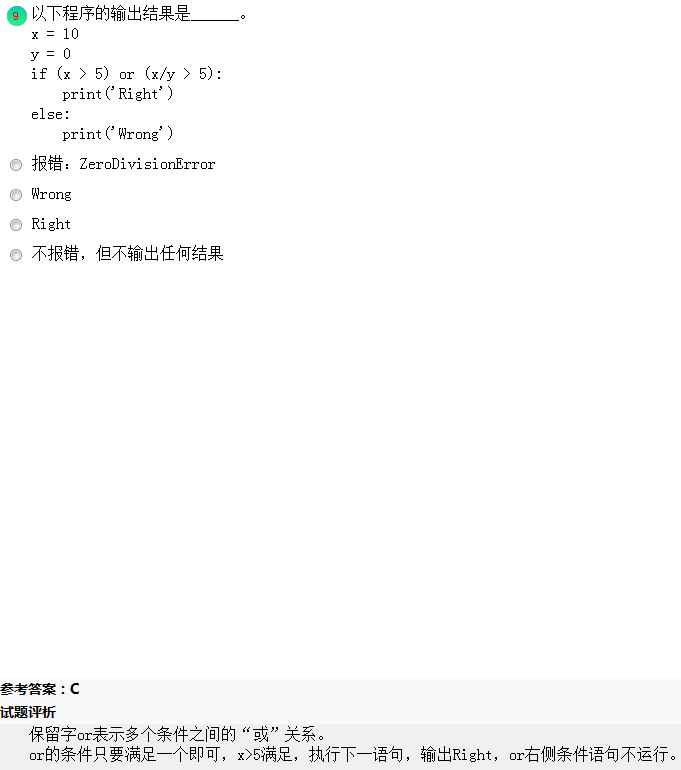


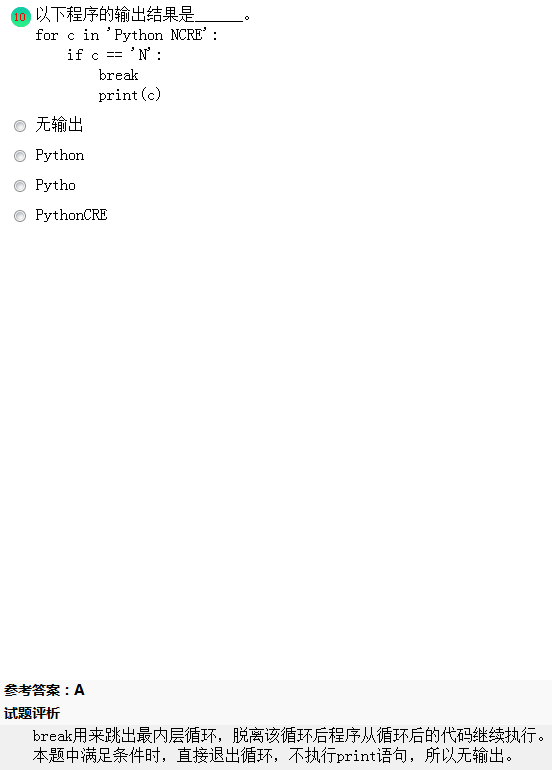




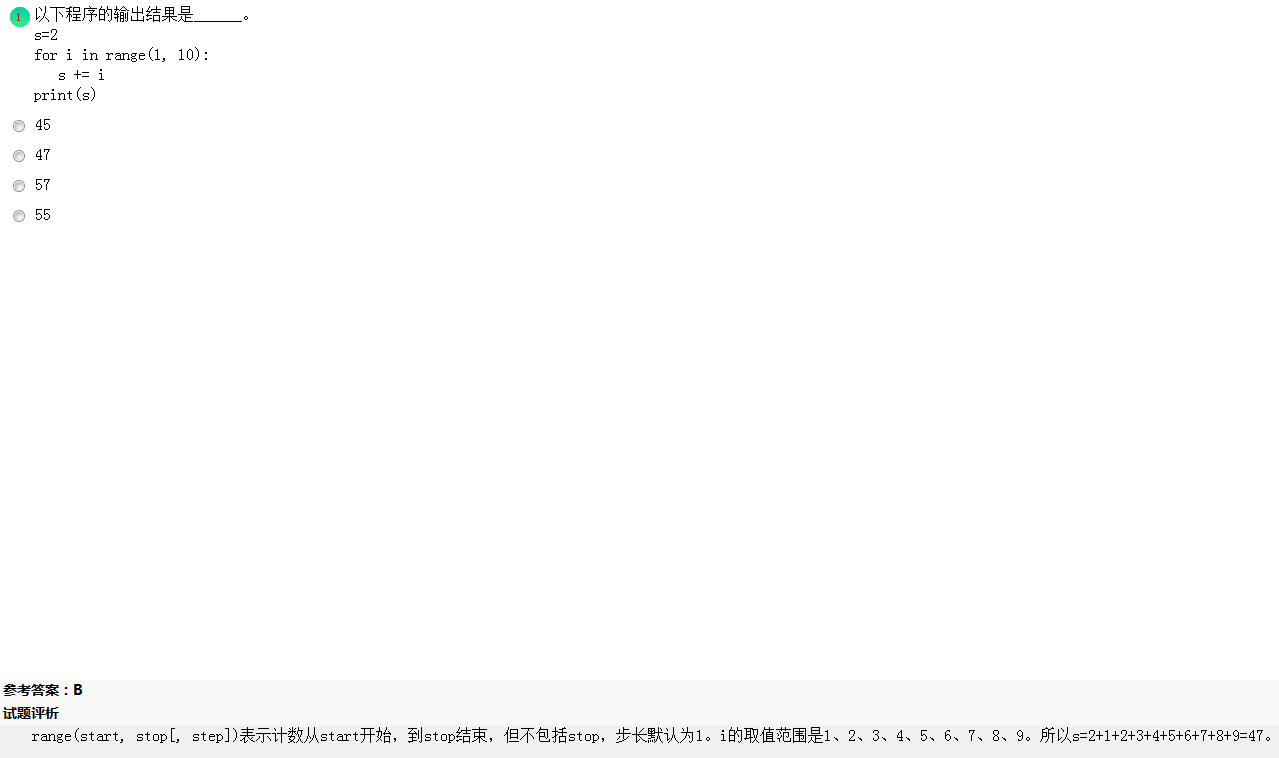






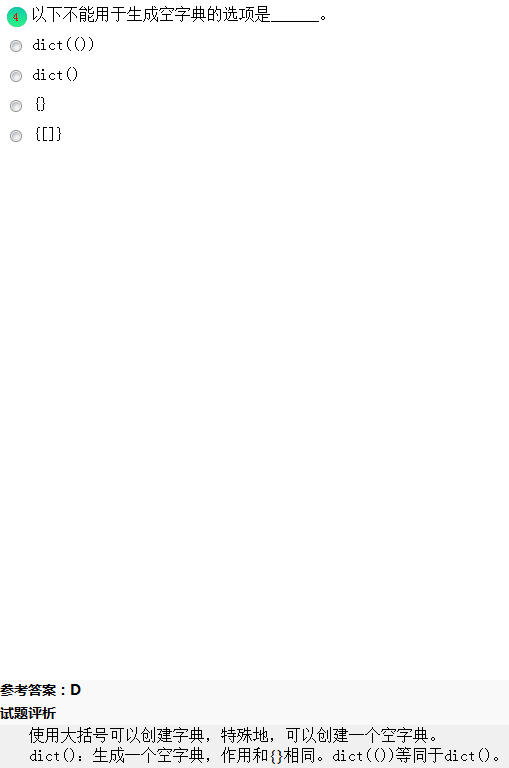


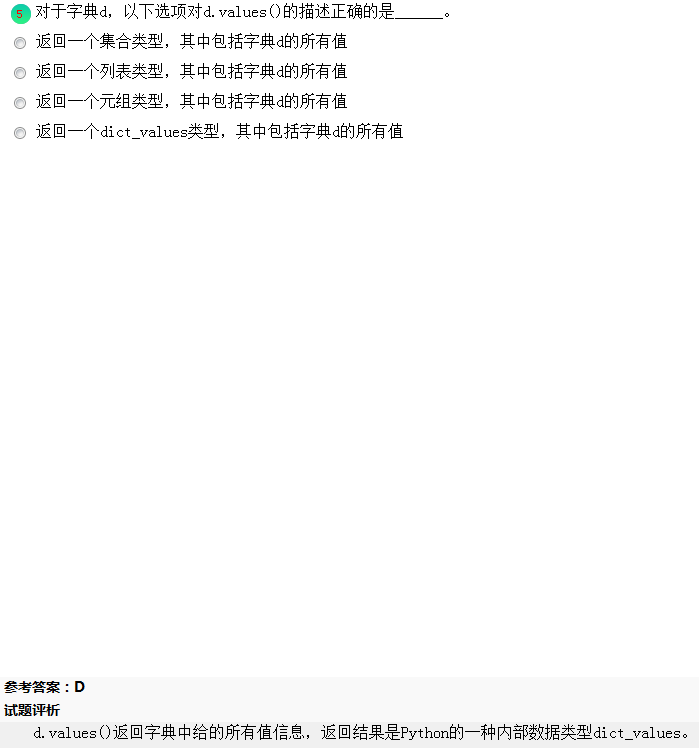
### 40-48题

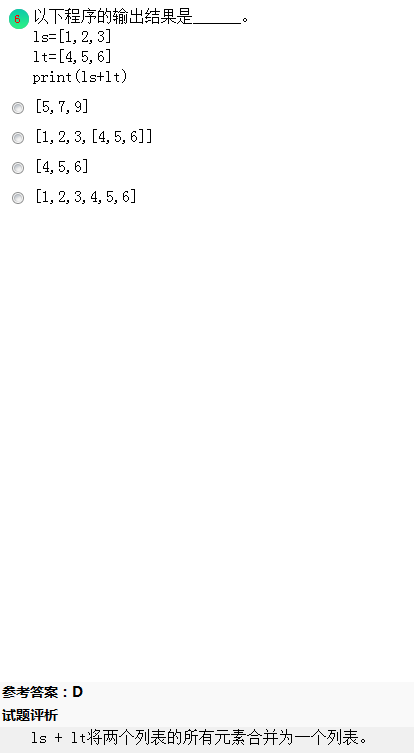


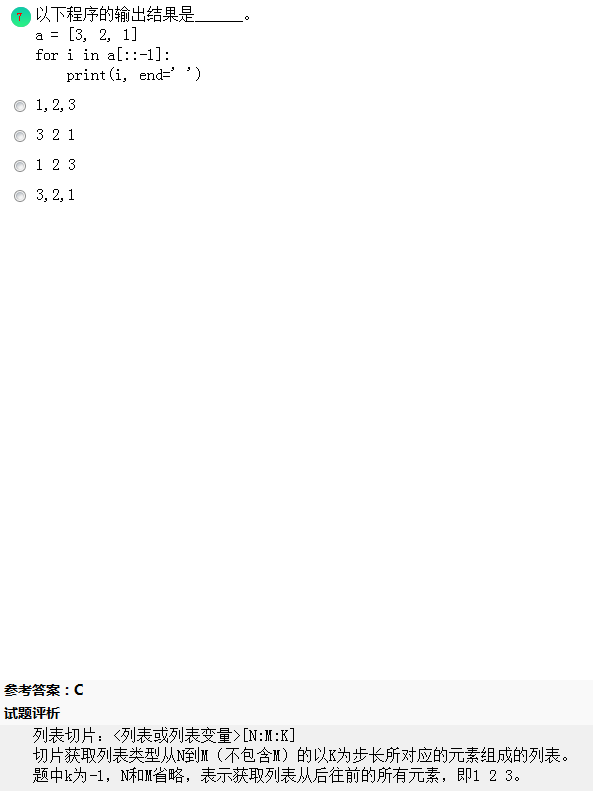


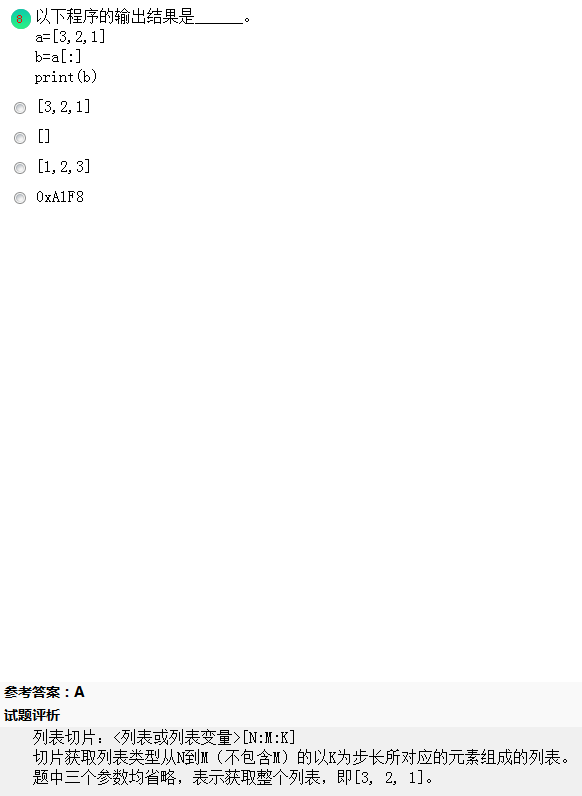








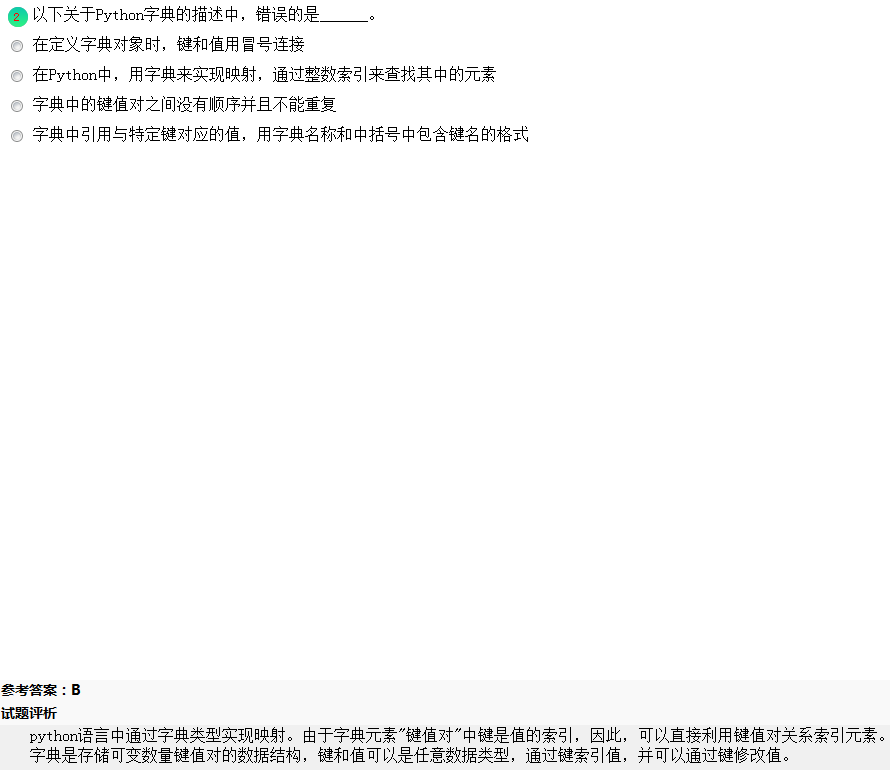




## 4-Python组合（48）

### 1-10题













Python通过保留字for实现遍历循环，使用方法如下：

for <循环变量> in <遍历结构>

<语句块>

遍历循环可以理解为从遍历结构中逐一提取元素，放在循环变量中，对于每个所提取的元素执行一次语句块。for语句的执行次数是根据遍历结构中元素个数确定的。

使用range()函数，可以指定语句块的循环次数。

题目中的代码循环过程如下：

c=[1,2,3]，j=0，c[0]=1，s=1；

c=[1,2,3]，j=1，c[1]=2，s=3；

c=[1,2,3]，j=2，c[2]=3，s=6；

c=[4,5,6]，j=0，c[0]=4，s=10；

c=[4,5,6]，j=1，c[1]=5，s=15；

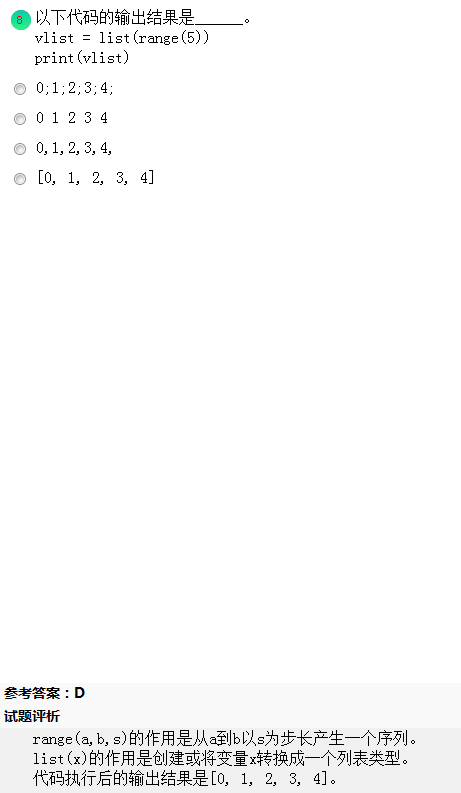
c=[4,5,6]，j=2，c[2]=6，s=21；

c=[7,8,9]，j=0，c[0]=7，s=28；

c=[7,8,9]，j=1，c[1]=8，s=36；

c=[7,8,9]，j=2，c[2]=9，s=45。

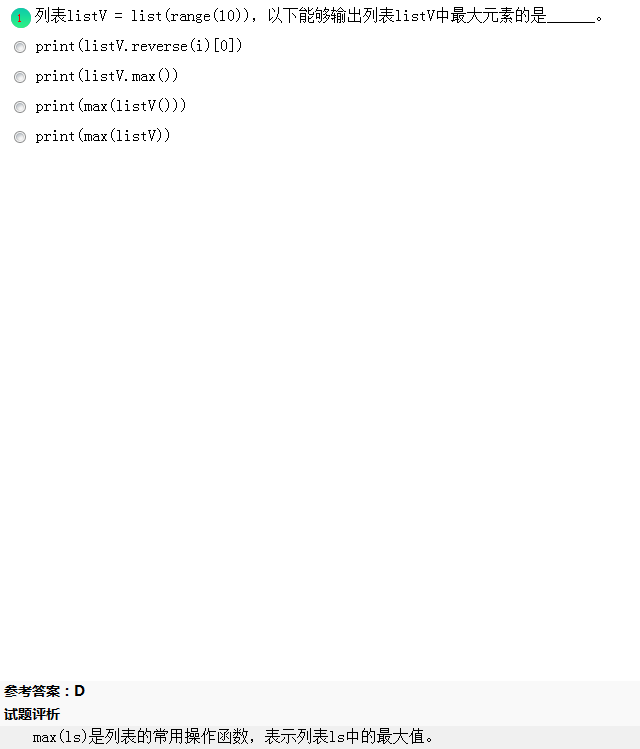


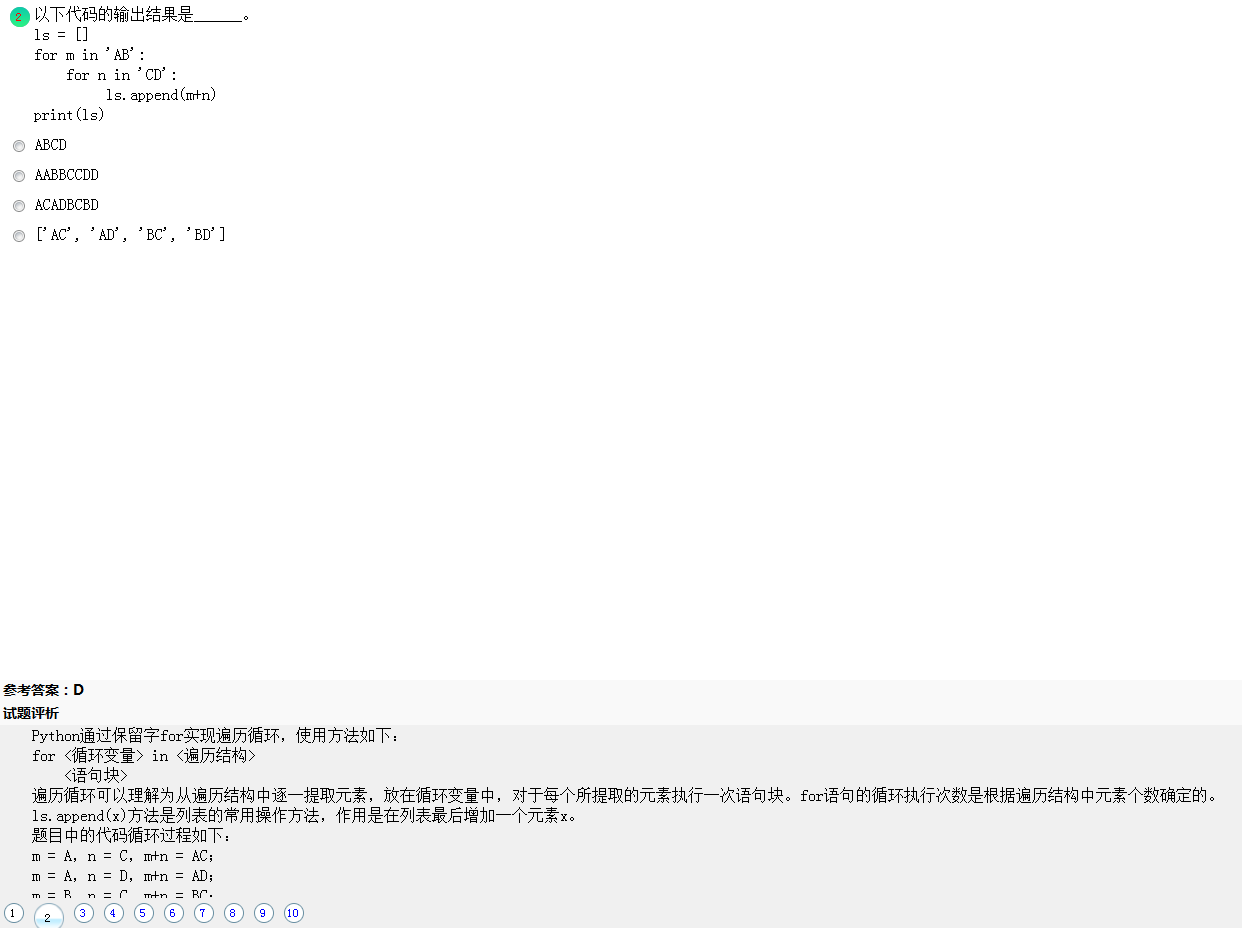






### 11-20题





Python通过保留字for实现遍历循环，使用方法如下：

for <循环变量> in <遍历结构>

<语句块>

遍历循环可以理解为从遍历结构中逐一提取元素，放在循环变量中，对于每个所提取的元素执行一次语句块。for语句的循环执行次数是根据遍历结构中元素个数确定的。

ls.append(x)方法是列表的常用操作方法，作用是在列表最后增加一个元素x。

题目中的代码循环过程如下：

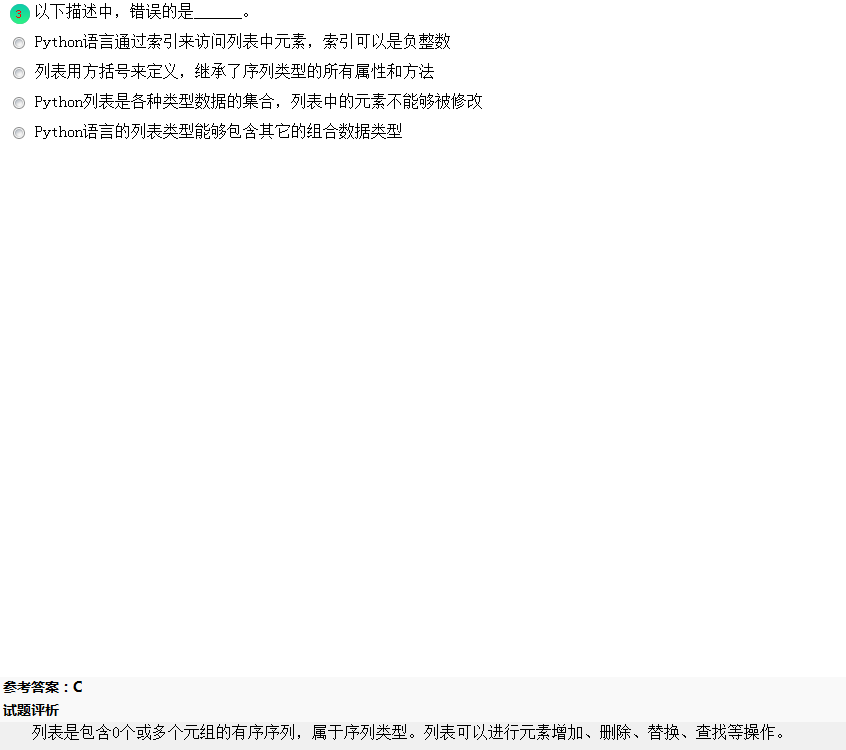
m = A，n = C，m+n = AC；

m = A，n = D，m+n = AD；

m = B，n = C，m+n = BC；

m = B，n = D，m+n = BD。

代码执行后的输出结果是['AC', 'AD', 'BC', 'BD']。







Python通过保留字for实现遍历循环，使用方法如下：

for <循环变量> in <遍历结构>

<语句块>

遍历循环可以理解为从遍历结构中逐一提取元素，放在循环变量中，对于每个所提取的元素执行一次语句块。for语句的循环执行次数是根据遍历结构中元素个数确定的。

len(S)：字符串S的长度，这里为4。i的取值范围为range(4)，即0、1、2、3。

题目中的代码循环过程如下：

i = 0，S[-i]=P；

i = 1，S[-i]=e；

i = 2，S[-i]=m；

i = 3，S[-i]=a。

代码执行后的输出结果是Pema。



Python通过保留字for实现遍历循环，使用方法如下：

for <循环变量> in <遍历结构>

<语句块>

遍历循环可以理解为从遍历结构中逐一提取元素，放在循环变量中，对于每个所提取的元素执行一次语句块。for语句的循环执行次数是根据遍历结构中元素个数确定的。

Python的单分支结构使用if保留字对条件进行判断，使用方法如下：

if <条件>：

<语句块>

其中，if、：和<语句块>前的缩进都是语法的一部分。<条件>是一个产生True或False结果的语句，当结果为True时，执行<语句块>，否则跳过<语句块>。

continue用来结束当前当次循环，即跳出循环体中下面尚未执行的语句，但不跳出当前循环。

题目中的代码循环过程如下：

s = H，输出结果H；

s = e，输出结果e；

s = l，输出结果l；

s = l，输出结果l；

s = o，输出结果o；

s = W，跳出当前当次循环，继续下一次循环；

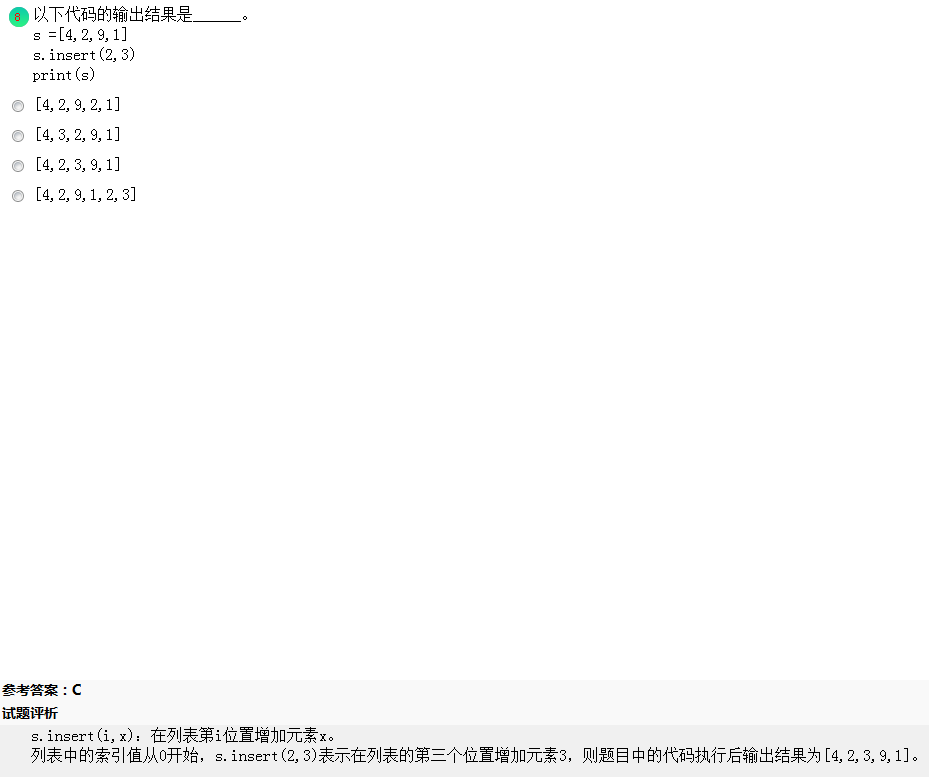
s = o，输出结果o；

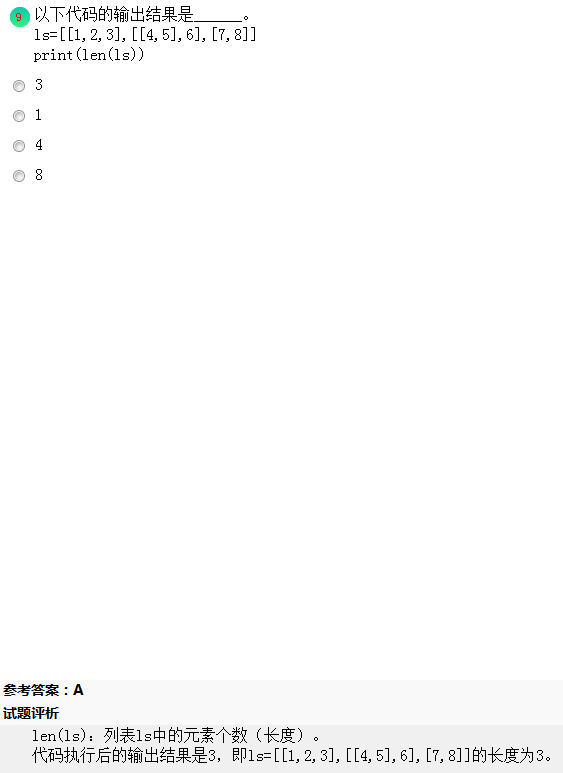
s = r，输出结果r；

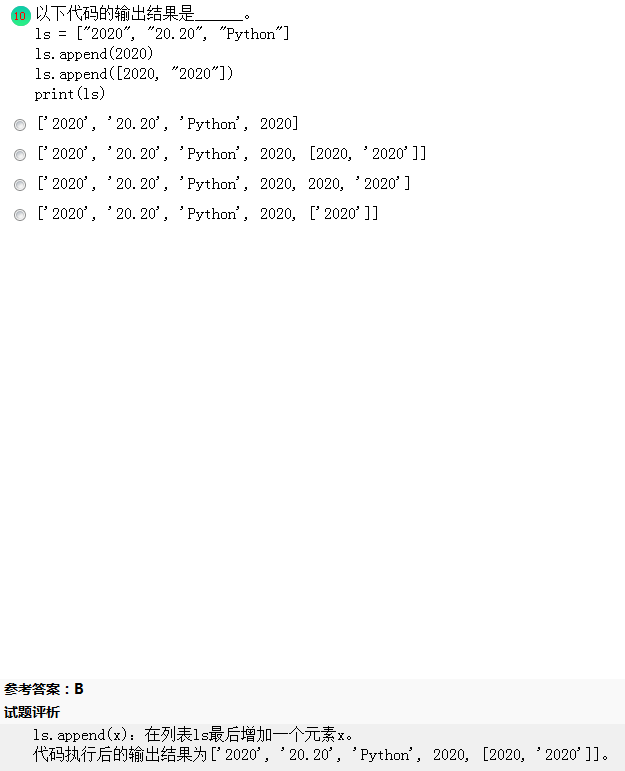
s = d，输出结果d。

代码执行后的输出结果是Helloorld。









### 21-30题







Python通过保留字for实现遍历循环，使用方法如下：

for <循环变量> in <遍历结构>

<语句块>

遍历循环可以理解为从遍历结构中逐一提取元素，放在循环变量中，对于每个所提取的元素执行一次语句块。for语句的循环执行次数是根据遍历结构中元素个数确定的。

chr(x)返回Unicode编码x对应的单字符，ord(x)返回单字符x表示的Unicode编码。

代码一开始生成一个空字典，第一个for循环用于向其中添加字典元素。range(26)表示i的取值范围是0至25之间的整数（包括0与25）。"a"对应的Unicode编码为97。

i=0时，chr(ord("a"))='a'，chr((13) % 26 + ord("a"))=chr(13+97)= 'n'，则第三行赋值语句简化为d['a']='n',此时字典d为{'a': 'n'}。

i=1时，chr(1+ord("a"))= 'b'，chr((1+13) % 26 + ord("a"))=chr(14+97)= 'o'，则第三行赋值语句简化为d['b']='o'，此时字典d为{'a': 'n', 'b': 'o'}。

依次类推，最后得出字典d为{'a': 'n', 'b': 'o', 'c': 'p', 'd': 'q', 'e': 'r', 'f': 's', 'g': 't', 'h': 'u', 'i': 'v', 'j': 'w', 'k': 'x', 'l': 'y', 'm': 'z', 'n': 'a', 'o': 'b', 'p': 'c', 'q': 'd', 'r': 'e', 's': 'f', 't': 'g', 'u': 'h', 'v': 'i', 'w': 'j', 'x': 'k', 'y': 'l', 'z': 'm'}，简而言之，每个字母键对应的是Unicode编码比自己大13的字母值。

d.get(key, default)表示若键存在则返回相应值，否则返回默认值default。所以第二个for循环最后的输出结果为Plguba。







range(start, stop[, step])表示计数从start开始，到stop结束，但不包括stop，步长默认为1。start默认从0开始，range(2)等价于range(0,2)，则k的取值范围为0、1。

break用来跳出最内层循环，脱离该循环后程序从循环后的代码继续执行。

第1次循环，i=N，k=0，if判断条件不成立，输出i的值为N。

第2次循环，i=N，k=1，if判断条件不成立，输出i的值为N。

第3次循环，i=a，k=0，if判断条件不成立，输出i的值为a。

第4次循环，i=a，k=1，if判断条件不成立，输出i的值为a。

……

第11次循环，i=n，k=0，if判断条件成立，跳出内层循环。

第12次循环，i=n，k=1，if判断条件成立，跳出内层循环。

最后输出为NNaattiioo。

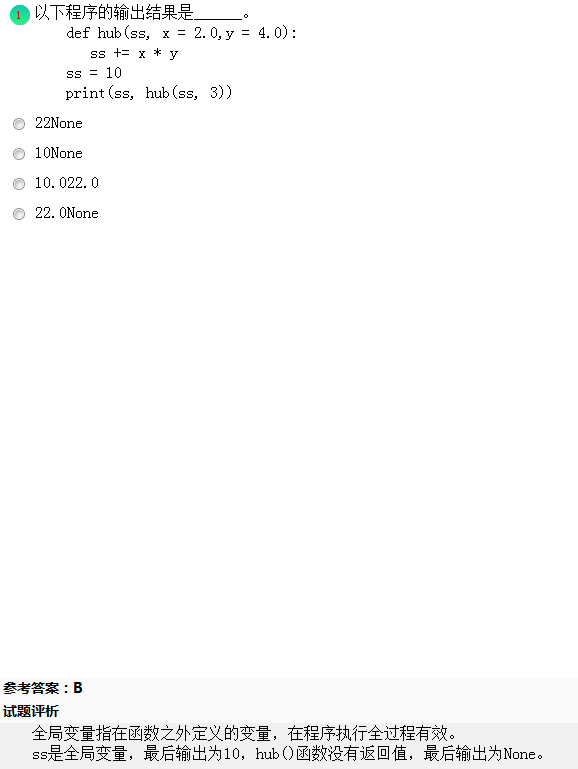






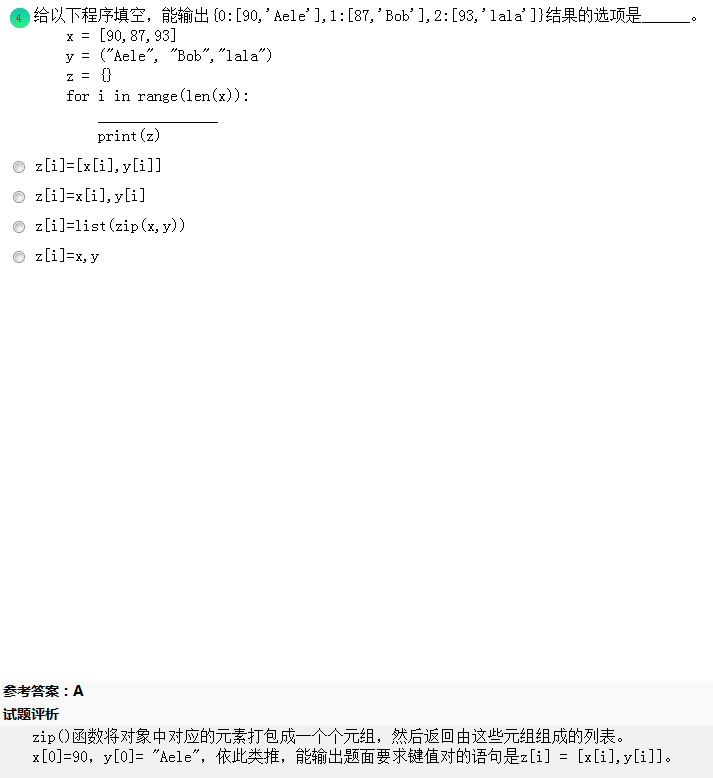


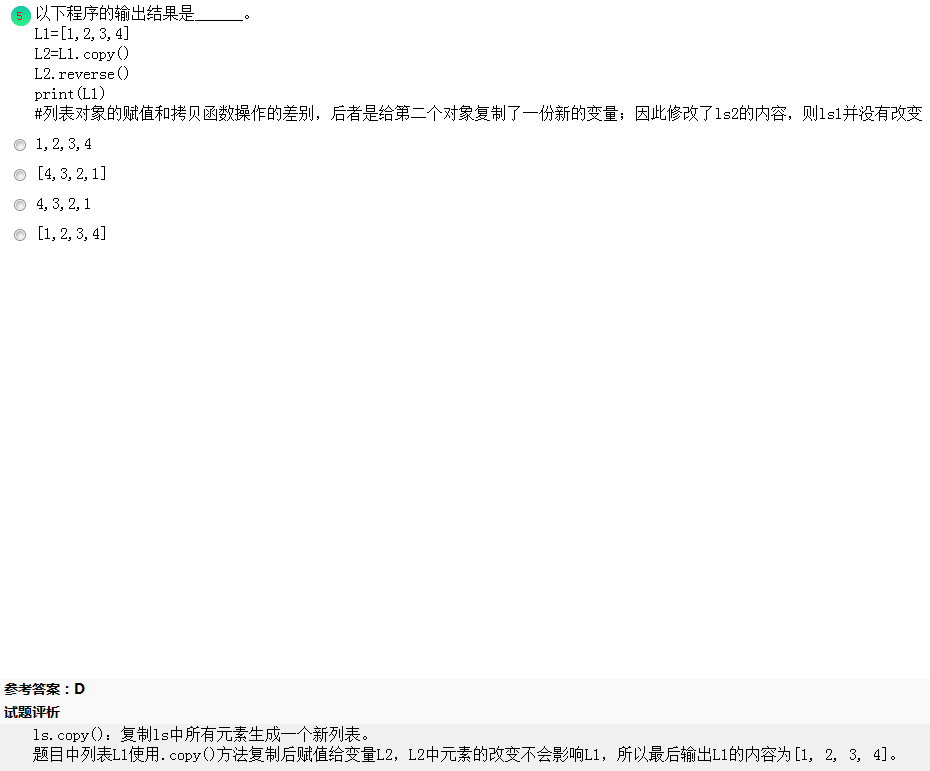
### 31-40题



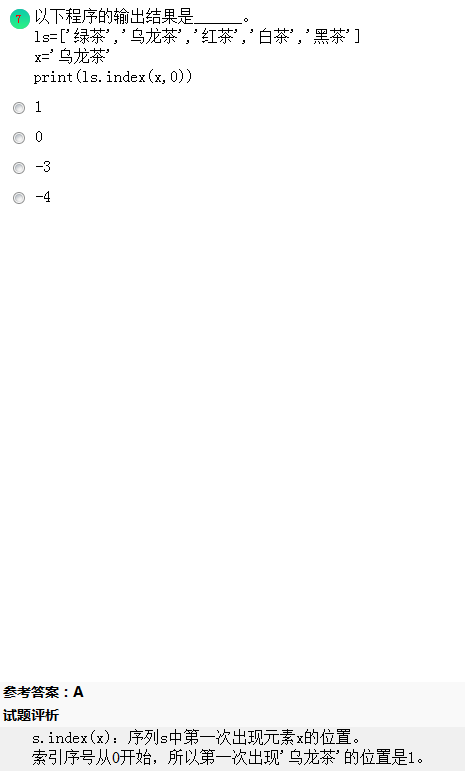


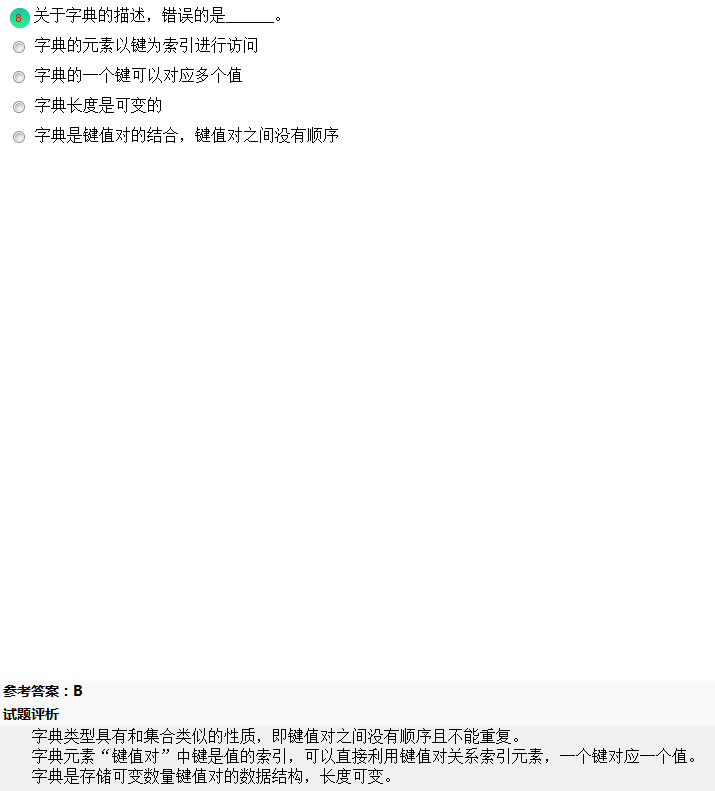










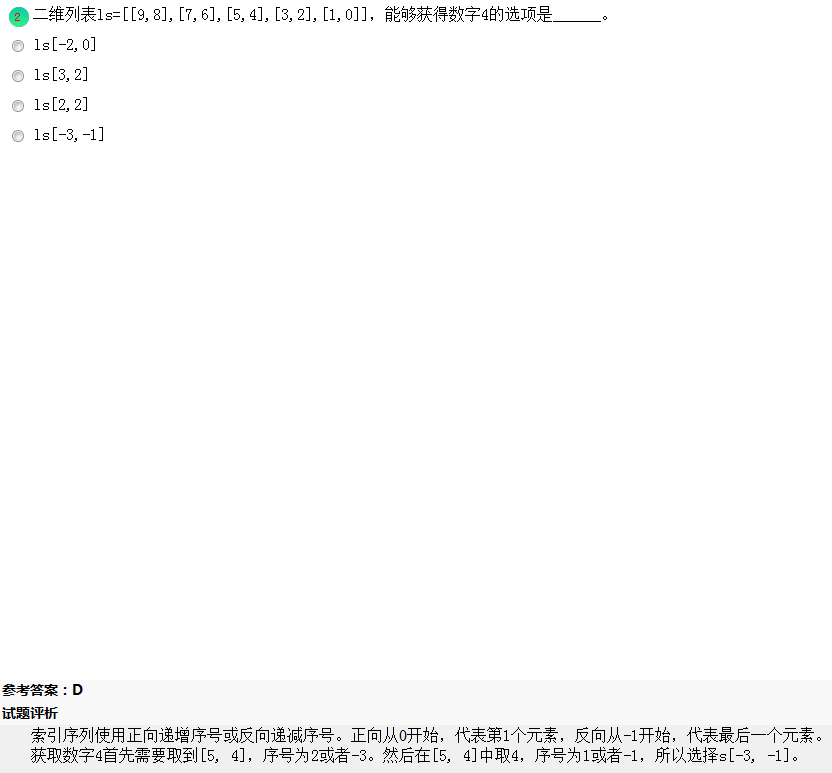


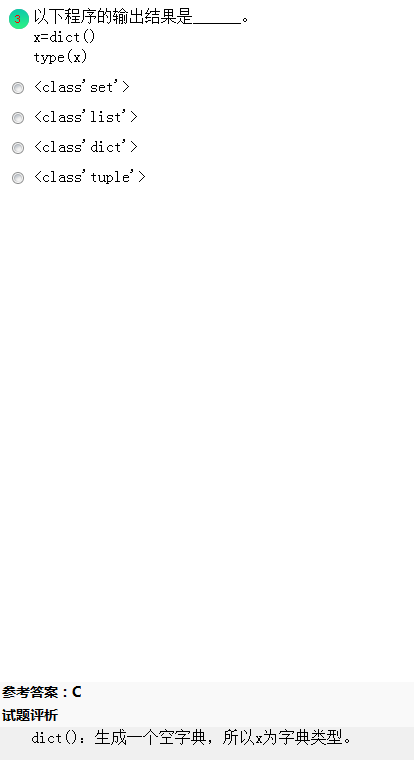




### 40-48题



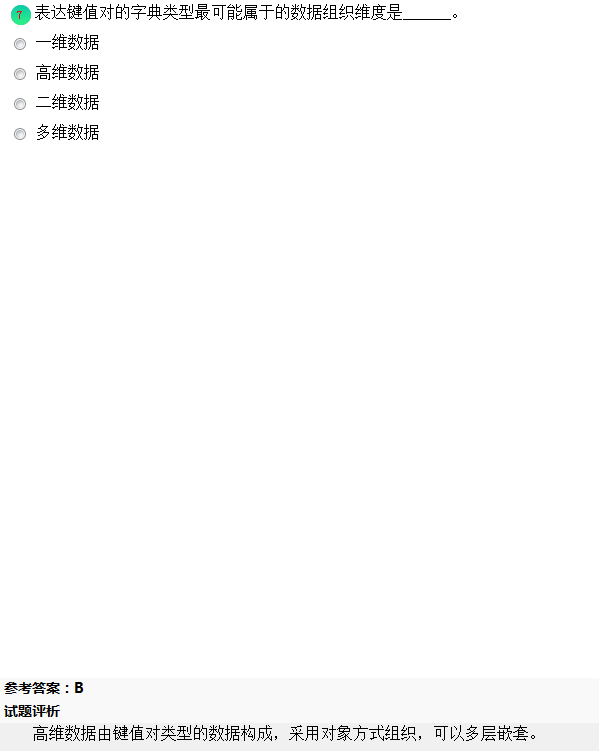


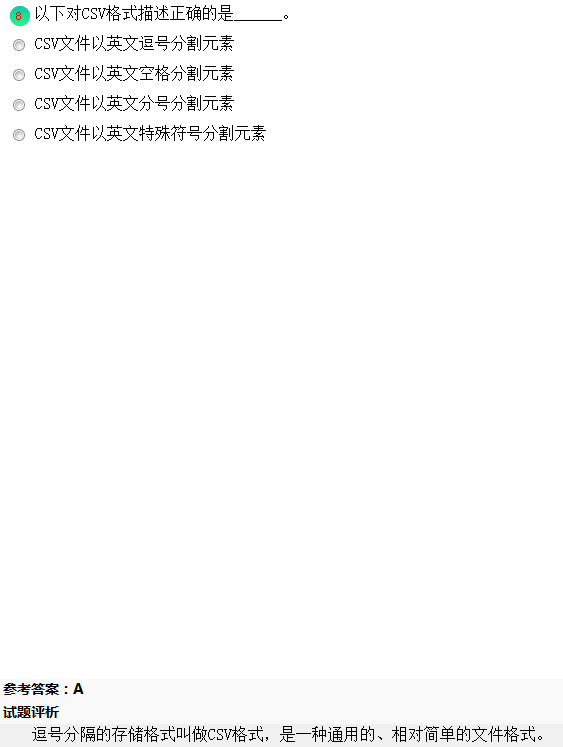






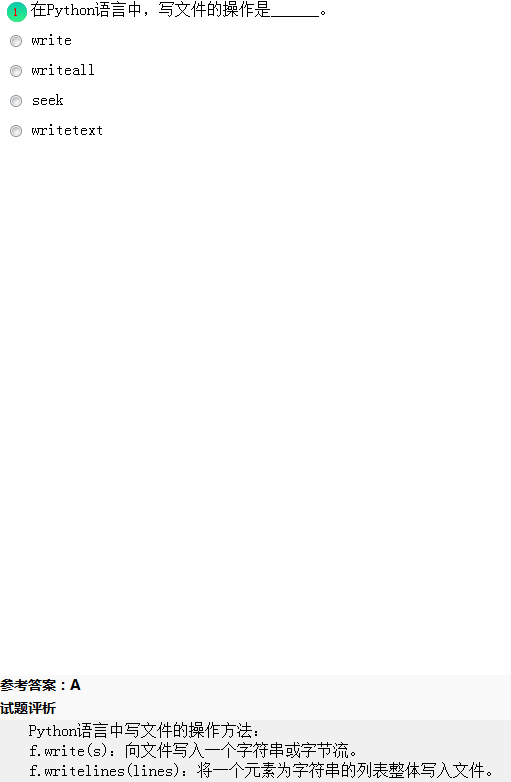






## 5-Python文件（40）

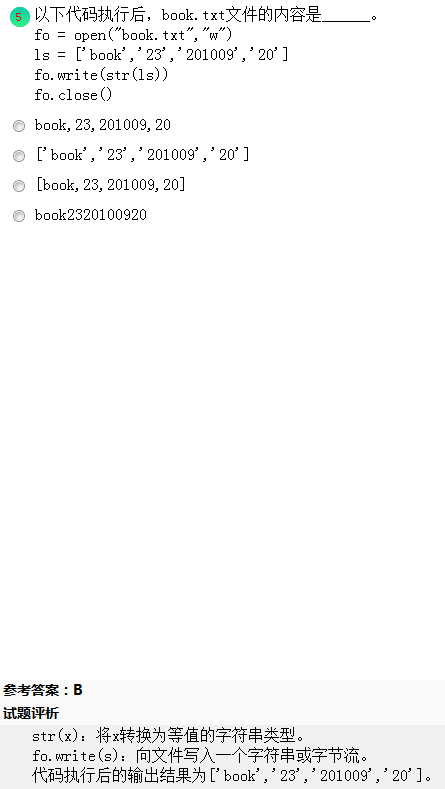
### 1-10题





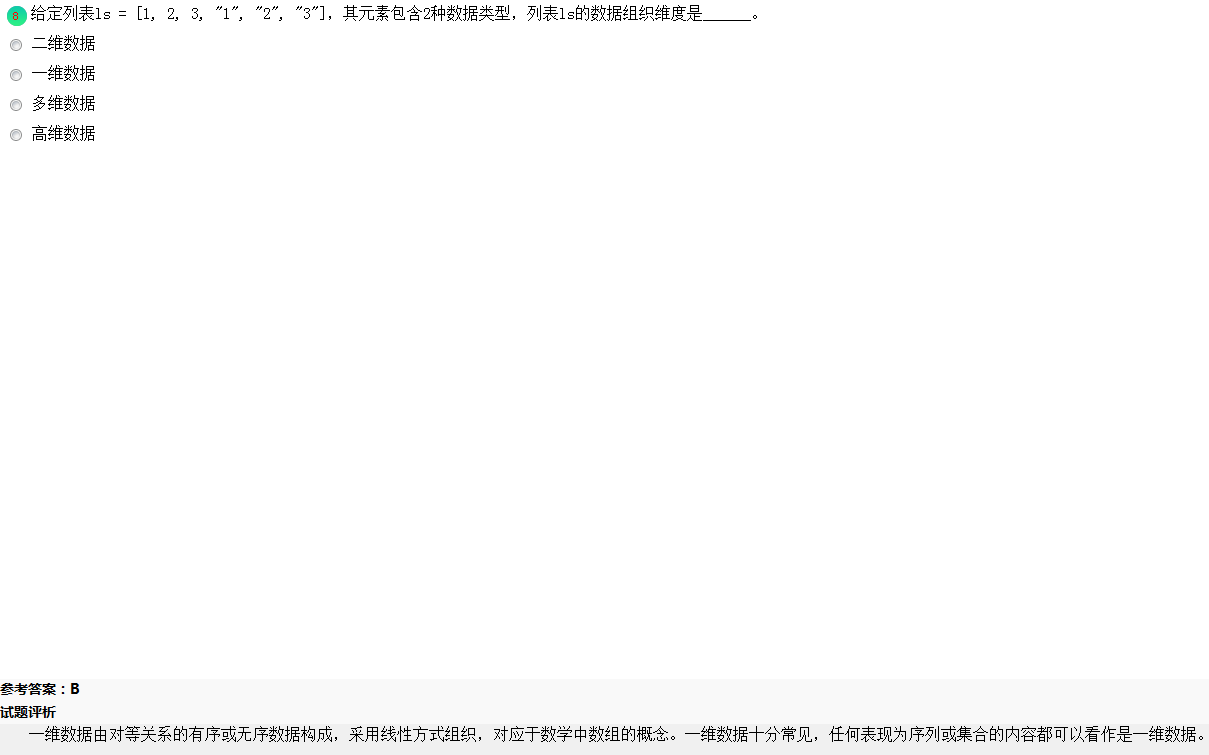
















### 11-20题





Python通过open()函数打开一个文件，并返回一个操作这个文件的变量，语法形式如下：

<变量名> = open(<文件路径及文件名>,<打开模式>)

open()函数有两个参数：文件名和打开模式。打开模式用于控制使用何种方式打开文件，open()函数提供7中基本的打开模式，如下：

r：只读模式，如果文件不存在，返回异常FileNotFoundError，默认值

w：覆盖写模式，文件不存在则创建，存在则完全覆盖原文件

x：创建写模式，文件不存在则创建，存在则返回异常FileExistsError

a：追加写模式，文件不存在则创建，存在则在原文件最后追加内容

b：二进制文件模式

t：文本文件模式，默认值

+：与r/w/x/a一同使用，在原功能基础上增加读写功能

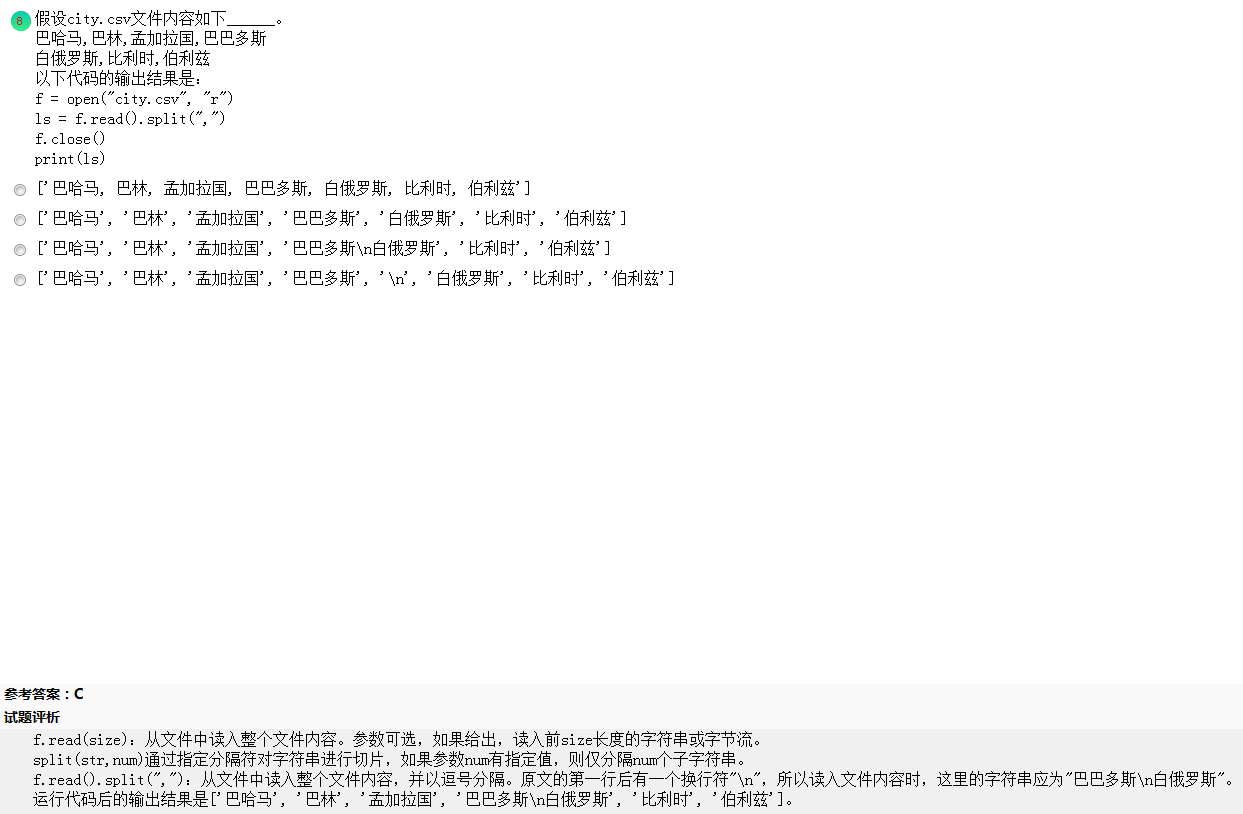
13b

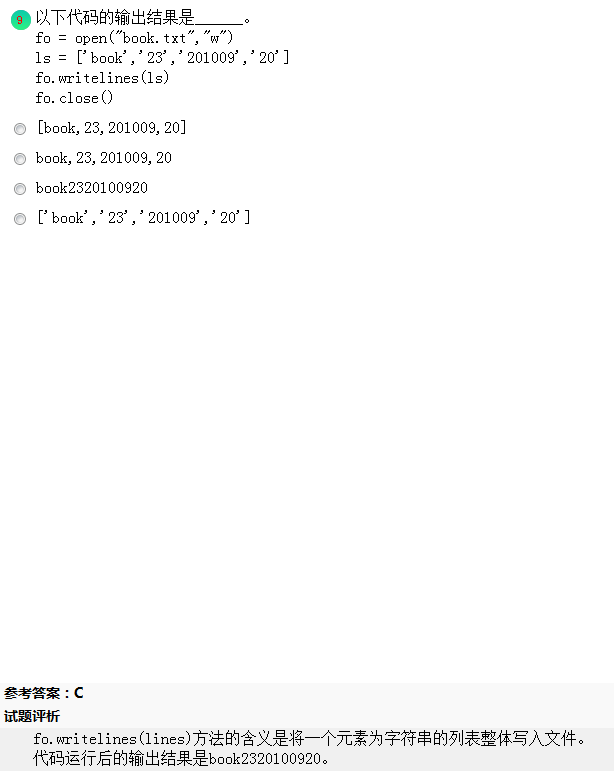














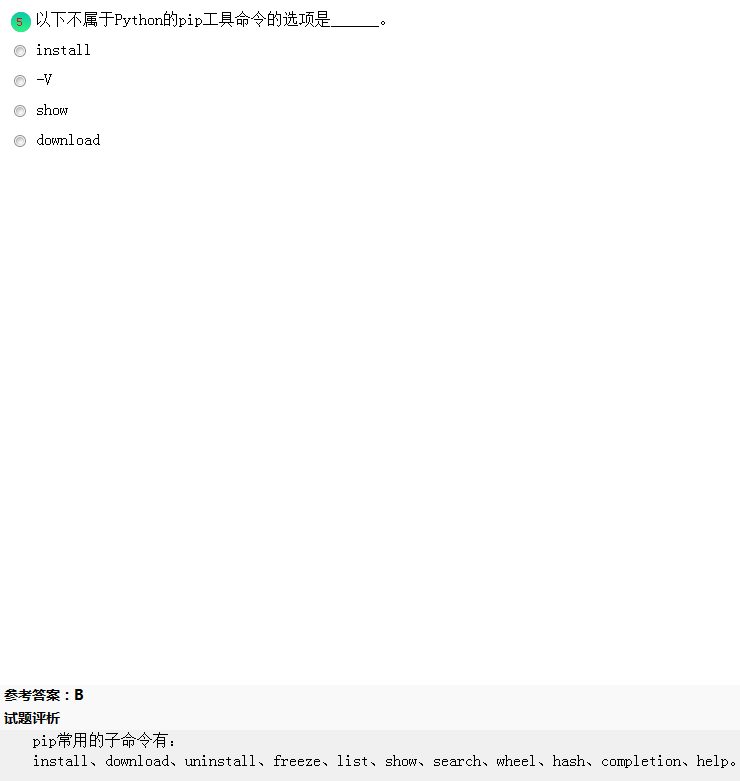
### 21-30题



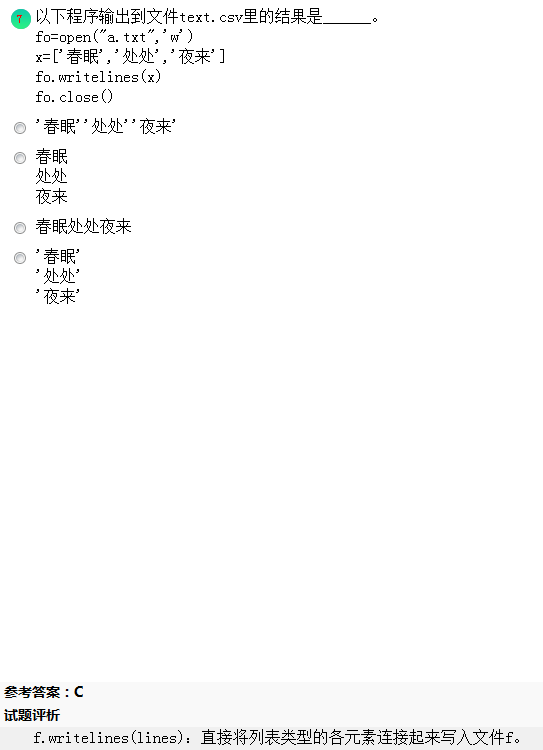


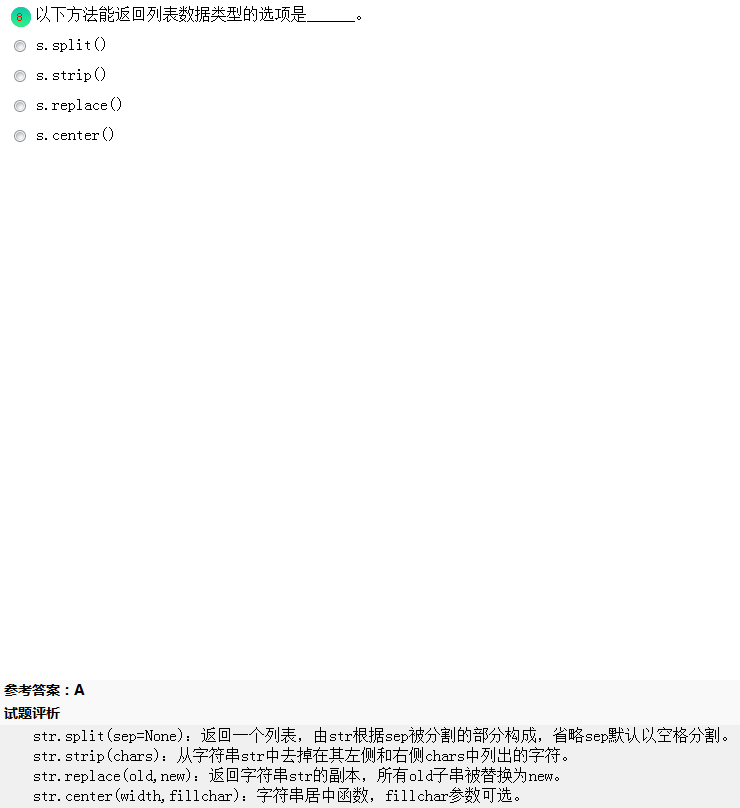














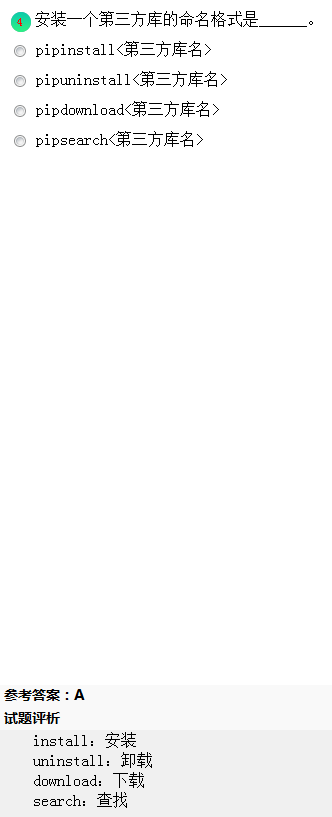


### 31-40题









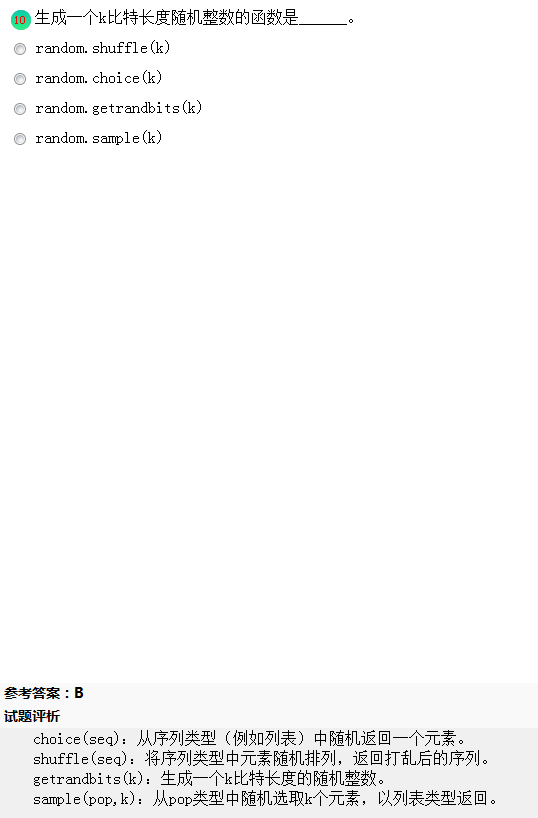










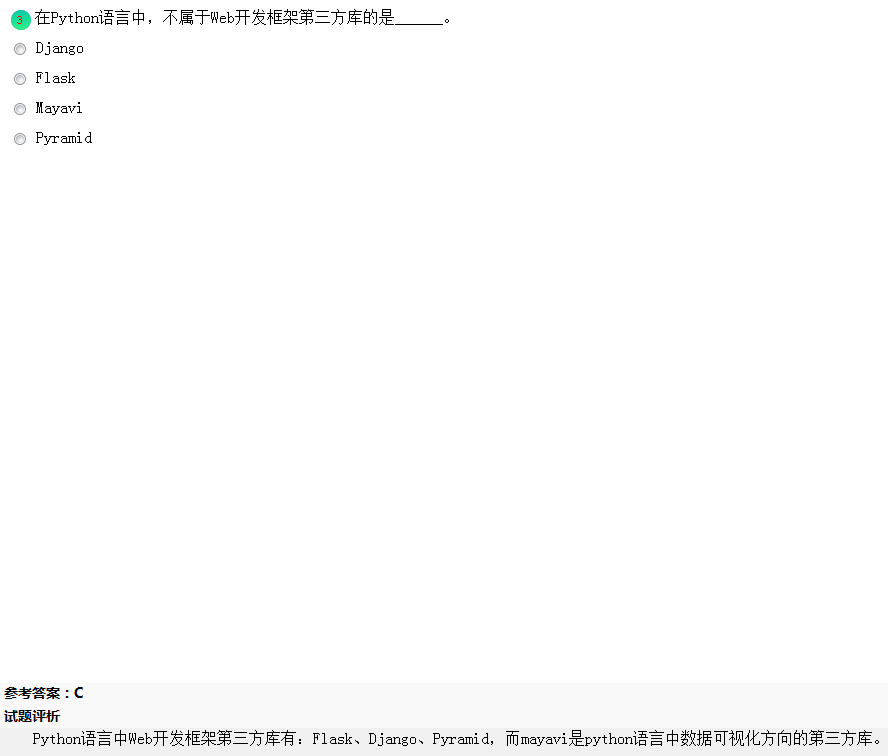
AAA

## 6-Python生态（24）

### 1-10题













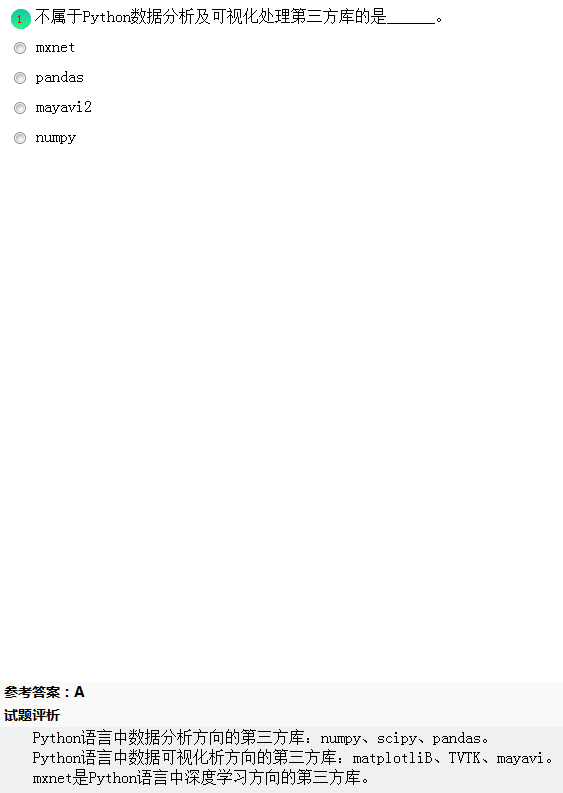








### 11-20题





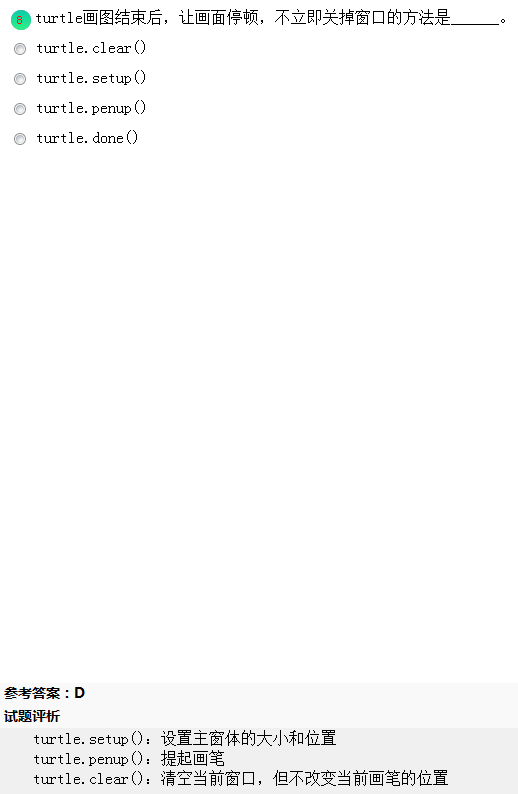
















### 21-24题







