

高级语言程序设计 I 试题 B

学号	
姓名	

题号	一	二	三	四	五	总分
分数						

注
意
行
为
规
范

遵
守
考
场
纪
律

主管
领导
审核
签字

说明：本试卷卷面满分为 100 分。总成绩将按平时实验成绩和卷面笔试成绩各自所占比例折合后累加计算。如此卷为补考试卷，则按卷面分数记入补考成绩。

考试时间：120 分钟。

考试方式：开卷考试，可以带一切纸质资料，不能携带任何电子设备，并只可以参考自己的资料，不可以相互交流。

如无特殊说明，所有题目都针对 Python 2 版本

一、单项选择题（共 15 小题，每题 2 分，共 30 分。每题只有一个正确答案，请将正确答案对应的索引字母填入括号内。）

- (1) 下列不是 Python 语言关键字的是：()

A. print

B. in

C. open

D. exec
- (2) 关于赋值语句的作用，正确的描述是：()

A. 将变量绑定到对象

B. 每个赋值语句只能给一个变量赋值

C. 将变量改写为新的值

D. 变量和对象必须类型相同
- (3) 关于函数的下列说法正确的是：()

A. 函数内部不可改变实参指向的对象

B. 函数一次可以返回多个对象

C. 函数调用时将实参的值赋给形参

D. 函数必须有返回值
- (4) 关于数据类型，下列说法正确的是：()

A. 决定数据存储内存的大小

B. 决定以何种方式输出数据

C. 决定数据可以执行的运算

D. Python 语言不区分数据类型

- (5) 与 $a > b$ and $b > c$ 语句等价的是： ()
- A. $a > b > c$ B. $a > b$ or not $b < c$
C. not $a < b$ or $b < c$ D. not $a < b$ and $b < c$
- (6) 若 $a = \text{'abcd'}$ ，若想将 a 变为 'abce' ，则下列语句正确的是 ()
- A. $a[3] = \text{'e'}$ B. $a[4] = \text{'e'}$
C. $a.\text{replace}(\text{'d'}, \text{'e'})$ D. $a = a[:3] + \text{'e'}$
- (7) 在循环语句的循环体中执行 **break** 语句，其作用是 ()
- A. 跳出该循环体，提前结束循环
B. 继续执行 **break** 语句之后的循环体各语句
C. 结束本次循环，进行下次循环
D. 跳出全部循环体，提前结束所有循环
- (8) 关于字典数据结构，下面描述正确的是： ()
- A. 键和值可以是任意类型 B. 查找键和值的速度一样快
C. 键值对是无序的 D. 支持对键进行排序操作
- (9) 关于列表数据结构，下面描述正确的是： ()
- A. 必须按顺序插入元素 B. 可以不按顺序查找元素
C. 所有元素类型必须相同 D. 不支持 **in** 运算符
- (10) 执行以下两条语句后， lst 的结果是： ()
- ```
lst = [1, 2, 3]
lst.append(lst)
```
- A.  $[1, 2, 3, lst]$  B.  $[1, 2, 3, [1, 2, 3]]$   
C.  $[1, 2, 3, [...]]$ ，其中 “...” 表示无穷递归 D. 抛出异常
- (11) 以下程序的输出结果是： ( )
- ```
y = 0
for i in range(0, 10, 2):
    y += i
print y
```
- A. 9 B. 10
C. 20 D. 30

(12) 假设 $x = 1$, $y = -1$, and $z = 1$, 则以下程序的输出结果是： ()

```
if x > 0:
    if y > 0: print 'AAA'
elif z > 0: print 'BBB'
```

- A. AAA B. BBB
C. 无输出 D. 语法错误

(13) 以下程序的输出结果是： ()

```
x = 1
def f():
    x = x + 2
    print x
f()
print x
```

- A. 3 3 B. 3 1
C. 1 3 D. 抛出异常

(14) 假设 $x = [[1, 2], [3, 4, 5], [6, 7, 8, 9]]$, 则 $\text{len}(x)$ 和 $\text{len}(x[2])$ 分别是： ()

- A. 2, 2 B. 3, 3
C. 3, 4 D. 抛出异常

(15) 以下哪条语句定义了一个 Python 字典： ()

- A. $[1, 2, 3]$ B. $(1, 2, 3)$
C. $\{1, 2, 3\}$ D. $\{\}$

二、改错题（10 分）

编写程序，输入一个大于等于 0 的数 x ，使用二分查找方法获得 x 的平方根。请找到并改正其中的任意五行错误或不当的代码，不改正不得分（可以在题目上直接改）。

```
x = raw_input('Input: ')

low = 0

high = x

ans = (low + high) / 2

while ans**2 != x:

    if ans**2 < x:

        low = ans + 1

    else:

        high = ans - 1

    ans = (low + high) / 2

print ans
```

三、写出程序的执行结果（共 20 分，每小题 4 分）

1. 写出下面程序的输出结果

```
list1 = [1, 2, 3]
list2 = list1
list3 = list2
list1.remove(1)
print list3
```

答：

2. 写出下面程序的输出结果

```
list1 = ['I', 'want', 'you', '.']  
list2 = [[i.upper(), i.lower(), len(i)] for i in list1 if len(i) < 4]  
print list2
```

答：

3. 写出下面程序的输出结果

```
def fun(lst):  
    lst = [4, 5, 6]  
  
lst = [1, 2, 3]  
fun(lst)  
print lst
```

答：

4. 写出下面程序的输出结果

```
d1 = {}  
d1[2] = 10  
d1['2'] = 20  
  
d2 = {}  
d2[2] = d1  
d2['2'] = d2  
  
print d2['2']['2']['2']['2'][2][2]
```

答：

5. 写出下面程序的输出结果

```
from turtle import forward, backward, left, right, up, down

def branch(length, level):
    if level <= 0:
        return
    forward(length)
    left(45)
    branch(0.6 * length, level - 1)
    right(90)
    branch(0.6 * length, level - 1)
    left(45)
    up()
    backward(length)
    down()
left(90)
branch(100, 4)
```

答：

四、程序设计题（共 2 小题，每题 20 分，共 40 分）

1. 帕斯卡三角形，又称杨辉三角形是二项式系数在三角形中的一种几何排列。帕斯卡三角形通常从第 0 行开始枚举，并且每一行的数字是上一行相邻两个数字的和。在第 0 行只写一个数字 1，然后构造下一行的元素。将上一行中数字左侧上方和右侧上方的数值相加。如果左侧上方或者右侧上方的数字不存在，用 0 替代。下面给出 6 行的帕斯卡三角形：

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5
```

编写程序，输入帕斯卡三角形的高度，然后生成和上面例子一样风格的三角形。

2. 假设丢失了单词的最后一个字母，例如单词 “fantasti”，需要从单词表中找到原单词 “fantastic”。请分别使用两种方法（不同的数据结构和算法）查找原单词。注意，在某些情况下，可能会找到多个单词。例如，“bas”，原单词可能为 “bass” 或 “base”，需要找到给定单词表中的全部原单词。其中，单词表存放在 “words.dic” 文件中。最后，请对这两种实现方法的效率加以比较。

五、附加题 (10 分)

编写程序实现一个猜字游戏，从单词文件 “words.dic” 中随机的选取一个单词，提示用户一次猜一个字母，下面是一个例子，首先单词中的字母都由*代替，如果用户猜的字母恰好在单词中，则显示该字母的正确位置。当用户猜出整个单词后，显示用户猜错的次数。

```
Enter a letter in word *****: p
p is not in the word
Enter a letter in word *****: l
Enter a letter in word l*****: a
Enter a letter in word la***a*: n
Enter a letter in word lan**a*: g
Enter a letter in word lang*ag*: u
Enter a letter in word languag*: a
a is already in the word
Enter a letter in word languag*: g
g is already in the word
Enter a letter in word languag*: e
The word is language. You missed 1 time(s).
```

