

## 高三下信息算法综合练习4

1. 甲、乙程序段的功能相同，则乙程序段加框处的正确代码为

<pre> a = 1 b = 1 m=int (input ("请输入 m 的值: ")) while a + b &lt;=m:     c = a + b     a = b     b = c print(b)                 </pre> <p style="text-align: center;">甲程序段</p>	<pre> def f(a,b):     c = a +b     if c &gt; m:         return b     else:         return <span style="border: 1px solid black; display: inline-block; width: 80px; height: 15px; vertical-align: middle;"></span> m = int(input( "请输入 m 的值: " )) print(f(1,1))                 </pre> <p style="text-align: center;">乙程序段</p>
--	--

- A. f(b, c)
B. f(a, b)
C. f(a, c)
D. f(c, b)

2. 有如下 Python 程序段：

```

from random import randint
n = 8
a = [randint(0, 4)*2 for i in range(n)]
for i in range(2):
    j = i+1
    while j < n-1:
        if i%2 ==0:
            if a[j-1] <a[j] and a[j] > a[j+1]:
                j+=2
            else:
                a[j] += 2
        else:
            if a[j-1] > a[j] and a[j] <a[j+1]:
                j+=2
            else:
                a[j] -= 1
print(a)
                
```

运行后，数组 a 中的值可能的是

- A. [0, 4, 2, 12, 2, 10, 3, 4]
B. [2, 4, 3, 4, 2, 4, 2, 4]  
C. [7, 10, 6, 8, 6, 10, 5, 6]
D. [4, 8, 0, 10, 8, 10, 1, 2]

3. 数组元素 s[0]~s[5]为随机正整数，执行如下 Python 程序段后，s[0]~s[5]的值可能为

```

for i in range(5):
    for j in range(5, i, -1):
        if s[j] % 2 < s[j - 1] % 2:
            s[j], s[j - 1] = s[j - 1], s[j]
                
```

- A. 6, 2, 1, 3, 7, 7
B. 1, 3, 5, 7, 9, 2
C. 10, 8, 6, 3, 2, 1
D. 1, 2, 5, 5, 6, 7

4. 给定长度为 n 的数组 ts，需为每个元素 ts[i]在其后寻找首个更大元素 ts[j]。若找到，将两者位置距离存入 ans[i]，否则 ans[i]的值为 0。例如数组 ts 为[33,34,35,31,29,32,36,33]，则数组 ans 为[1,1,4,2,1,1,0,0]。实现该功能的程序段如下，方框处应填入的代码是：

```

ans = [0] * n ; s = [0] * n ; top = -1
for i in range(n):
    t = ts[i]
    while top >= 0 and t > ts[s[top]]:
        j = s[top]
        top -= 1
        
    top += 1
    s[top] = i
                
```

- A. ans[i] = j - i
B. ans[i] += 1
C. ans[j] = i - j
D. ans[i - j] += 1

5. 某款无人机具备自动返航功能，当满足以下任一条件时无人机自动返航：①信号连续丢失达3秒，②电量低于20%，③距起点超出200米。无人机启动后，每隔1秒自动采集1次自动返航的条件数据，格式记为[a,b,c]，其中a为信号是否丢失，b为电量，c为距起点距离。如[1,0.55,125]表示没有丢失信号（1为没丢失，0为丢失），电量55%，距起点125米。某次无人机飞行的数据下载至文本文件，编程分析无人机返航的时间及原因。请回答下列问题：

(1) 某次飞行数据中，无人机的初始状态正常，从第1秒开始采集的数据依次为：

[[1,0.28,5], [1,0.26,12], [0,0.23,20], [0,0.19,30], [0,0.17,40],  
[1,0.15,45],...]，无人机从第 ▲ 秒开始返航。

(2) 实现上述功能的部分Python程序如下，请在划线处填入合适的代码。

#读取文本文件，飞行数据存入列表data中，代码略

reasons=["信号连续丢失达3秒", "电量不足20%", "距起点超出200米"]

r = -1 ; j = 0 ; b = [1,1,1]

for i in range(len(data)):

①

j=(j+1)%3

if b[0] + b[1] + b[2] == 0:

r=0

break

if data[i][1] < 0.2 or ②:

if data[i][1] < 0.2:

r=1

else:

r=2

break

if r != -1:

print("无人机在第", i+1, "秒，因为", ③, "自动返航。")

else:

print("本次飞行，无人机没有自动返航。")

6. 某仓库记录每天出入库情况，正数表示入库数量，负数表示出库数量。编写程序，查询出入库总量等于k的连续时间段。某星期出入库情况为" 10, -10, 5, 6, 7, 8, -5"，k值为18的连续时间段有第1天~第5天，第3天~第5天。请回答下列问题：

(1) 若将题干例子中的k值改为21，则有 ▲ 个时间段满足要求。

(2) 实现上述功能的部分Python程序如下，请在划线处填入合适的代码。

# 读取出入库情况，存入列表a中，读取k值。代码略。

pos = {0 : [-1]}; count = 0

①

for i in range(len(a)):

cur += a[i]

②

if pre in pos:

for j in pos[pre]:

print("时间段:", j + 2, "-", i + 1)

③

if cur in pos:

pos[cur].append(i)

else:

pos[cur] = [i]

print("一共有", count, "个时间段")

7. 某软件对用户先后两次的运动状况进行评估，两次评估总分之差为其成长分数。针对某次评估，用户的总分由基础分和等级分相加得到。基础分根据用户成绩（0~100 之间的整数）所处的成绩区间对应取得，如第 15 题图所示；等级分由公式  $(f1 + 0.5 \times f2) / n \times 100$  计算后向下取整得到，其中，f1 为该区间内低于该用户成绩的人数，f2 为该区间内等于该用户成绩的人数，n 为该区间内的总人数（不包含仅参加一次评估的用户）。

成绩区间	基础分
81-100	80
61-80	60
41-60	40
21-40	20
0-20	0

第 15 题图

请回答下列问题：

- (1) 某次评估中，若用户成绩为 50，其中，f1 为 10，f2 为 4，n 为 20，则该用户总分为     ▲    。

- (2) 定义如下 intersection(lst1, lst2) 函数，列表 lst1、lst2 为先后两次评估的用户成绩数据表，lst1、lst2 中的每个元素依次由用户的编号和成绩 2 个数据项组成，列表均已按用户编号升序排列。函数的功能是筛选出参与了两次评估的用户的索引，结果存 lst 并返回。

```
def intersection (lst1, lst2):
    lst = [ ]
    i, j = 0, 0
    m1, m2 = len(lst1), len(lst2)
    while i < m1 and j < m2:
        n1, n2 = lst1[i][0], lst2[j][0]
        if n1 == n2:
            lst.append([i,j])
            i += 1
            j += 1
        elif n1 < n2:
            i += 1
        else:
            j += 1
    return lst
```

若 lst1 为[[1,95],[2,95],[3,78],[4,100],[5,99],[6,100]]，lst2 为[[2,86],[3,88],[4,99],[5,87],[6,77],[7,65]]，执行语句 lst = intersection (lst1, lst2) 后，请回答问题：

- ① 加框处语句执行的次数是     ▲    。  
 ② lst[0] 的值为     ▲    。

(3) 实现计算用户成长分数功能的部分程序如下，请在程序中划线处填入合适的代码。

```
def proc(lst, ilst, v):
    #创建 score 列表，共 101 个元素，每个元素的值均为[0,0,-1]，代码略
    #如果 ilst 值为 lst1，则 v = 0；如果 ilst 值为 lst2，则 v = 1
    for c in lst:
        m = c[v]
        _____ ① _____
        score[r][0] += 1
    heads = [-1,-1,-1,-1,-1]
    w, i = 20, 0
    for j in range(101):
        if j > w:
            i += 1
            w += 20
        if score[j][0] != 0:
            if heads[i] != -1:
                score[j][0] += score[heads[i]][0]
                _____ ② _____
            heads[i] = j
    for i in range(5):
        p = heads[i]
        if p != -1:
            n = score[p][0]
            while p != -1:
                q = score[p][2]
                f1 = 0
                if q != -1:
                    _____ ③ _____
                f2 = score[p][0] - f1
                score[p][1] = i * 20 + int((f1 + 0.5 * f2) / n * 100)
                p = q
    return score
'''
```

读取先后两次评估的用户成绩数据表数据分别存入 lst1、lst2。lst1 和 lst2 的每个元素依次由用户编号和成绩 2 个数据项组成，lst1、lst2 均已按用户编号升序排列，且同一用户在 lst1、lst2 中的编号保持一致。代码略

```
'''
lst = intersection(lst1, lst2)
score1 = proc(lst, lst1,0)
score2 = proc(lst, lst2,1)
for c in lst:
    s = lst1[c[0]][0]          #取出用户编号
    a = lst1[c[0]][1]          #取出先一次评估的用户成绩
    b = lst2[c[1]][1]          #取出后一次评估的用户成绩
    print("用户编号:",s,"用户成长分数:",score2[b][1] - score1[a][1])
```