

一、 选择

1. 下列选项中正确的有 ():
 - A. Python3 中 input()默认返回字符串类型的数据。
 - B. 集合中不能包含相同元素, 也不能修改。
 - C. 使用 index()方法可以判断列表中是否包含某一元素, 若不包含则返回-1。
 - D. 生成器与迭代器都可以用于 for 循环。

参考答案: AD

二、 阅读代码给出运行结果

1.

```
l1=list(tuple('12ab'))
l1.append([3])
l2=l1.copy()
print(l2)
```

```
l2[-1].append(4)
del l2[1::2]
print(l1)
print(l2)
```

参考答案:

```
['1', '2', 'a', 'b', [3]]
['1', '2', 'a', 'b', [3, 4]]
['1', 'a', [3, 4]]
```

2. `async def working(time):`

```
    await asyncio.sleep(time)
```

`async def worker_1():`

```
    task_list=[5,1,3]
```

```
    for task in task_list:
```

```
        await working(task)
```

```
        print('t1',task)
```

`async def worker_2():`

```
    task_list=[3,1,6]
```

```
    for task in task_list:
```

```
        await working(task)
```

```
        print('t2',task)
```

`async def main():`

```

t1=asyncio.create_task(worker_1())
t2=asyncio.create_task(worker_2())
await t1
await t2
asyncio.run(main())

```

参考答案:

t2 3

t2 1

t1 5

t1 1

t1 3

t2 6

```

3. graph=np.array(
    [[0,1,1,1,1],
     [1,0,0,0,0],
     [1,1,0,0,0],
     [0,1,1,0,1],
     [1,1,1,1,0]])
D=nx.from_numpy_array(graph,create_using=nx.DiGraph)
for node in D.neighbors(0):
    if D.in_degree(node)>2:
        print(node,D.in_degree(node))

```

参考答案:

1 4

2 3

三、 填空

1. l=['Alice-0001','Bob-0022','Carol-0333','Dave-4444']

l 中以“姓名-序号”的形式记录信息，请用列表 l1 记录 l 中包含的姓名，用列表 l2 记录 l 中去除前置 0 的序号。

l1=_____

l2=_____

参考答案: l1=list(map(lambda x:x.split('-')[0],l))

l2=list(map(lambda x:x.split('-')[1].strip('0'),l))或

```
l2=list(map(lambda x:int(x.split('-')[1]),l))
```

四、 代码填空：

一、使用递归神经网络 RNN 模型预测单词的词性

1. 代码填空：对于每个单词，我们使用 one-hot 向量表示单词中的每一个字母，然后每次输入一个字母向量到 RNN 中。请参照注释以及训练函数，补充合适的代码，构建 RNN 模型。(3 分)

```
class RNN(nn.Module):
    def __init__(self, embedding_dim, hidden_size, output_size):
        super(RNN, self).__init__()
        self.hidden_size = hidden_size
        # TODO: 请补充两个线性层，分别实现“从输入到隐状态”、“从隐状态到输出”的变换
        _____
        _____

        self.softmax = nn.LogSoftmax(dim=1)

    def forward(self, input, hidden):
        combined = torch.cat((input, hidden), 1) # 将输入的字母向量和隐状态拼接
        # TODO: 请运用__init__()中定义的线性层和 softmax 层，实现 hidden 和 output 的计算
        _____
        _____
        _____

        return output, hidden

    def init_hidden(self):
        return torch.zeros(1, self.hidden_size) # 初始化全 0 的隐状态向量

# 以下为训练函数中的一部分代码
for name_ohe, label in data_: # name_ohe 为每个字母的 one-hot 表示，label 为预测标签
    hidden = RNN.init_hidden()
    for i in range(name_ohe.size()[0]): # 遍历每个字母，每次输入一个字母向量
        output, hidden = net(name_ohe[i], hidden)
    loss = criterion(output, label) # 用最后一次的 output 与真实 label 计算 loss
    loss.backward(retain_graph=True)
    total_loss += loss
```

答案：

```
self.i2h = nn.Linear(input_size + hidden_size, hidden_size)
```

```
self.h2o = nn.Linear(hidden_size, output_size)
```

```
hidden = self.i2h(combined)
```

```
output = self.h2o(hidden)
```

```
output = self.softmax(output)
```

五、 简答

1. 简述协程概念以及 python 中有哪两种实现协程的机制？