

1	2	3	4	5	6	7	8	9	10	11	12
C	D	A	B	D	B	B	C	D	B	A	C

本作业

高三上信息选考限时练7（作业41）

一、选择题（本大题共 12 小题，每小题 2 分，共 24 分。每小题列出的四个备选项中只有一个是符合题目要求的，不选、多选、错选均不得分。）

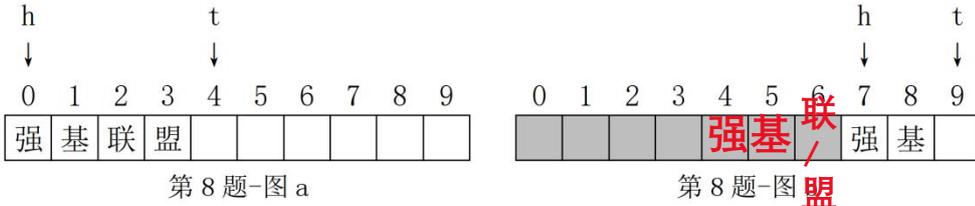
阅读下列材料，回答第 1 至 7 题：

在台风防治中，卫星、雷达、传感器等多源设备实时采集气压、风速、云图等海量数据，并通过 5G 网络传输至云数据中心。AI 技术通过深度学习模型对数据进行融合、清洗与挖掘，实现台风路径的预测和强度判定。同时，大数据技术也广泛应用于风险预警、灾害评估与救援调度，赋能防灾减灾的全流程。

- 下列关于数据的说法，不正确的是（**C**）
 - 台风路径的预测本质上是数据计算的结果
 - 对数据进行融合、清洗与挖掘，可充分发挥数据价值
 - 气象数据具备很强的实效性，过往数据没有价值
 - 卫星云图是气象数据的一种呈现形式
- 下列措施中，能有效提升数据传输效率的是（**D**）
 - 将数据转换为统一的格式
 - 对数据进行加密后传输
 - 增加服务器存储容量
 - 增加数据传输的网络带宽
- 在台风防治系统中，以下关于信息技术应用的说法，不正确的是（**A**）
 - 多源数据的融合主要依赖传感技术 **数据处理**
 - 拍摄卫星云图主要依赖遥感成像技术
 - 数据汇集到云中心依赖网络通信技术
 - 台风路径、强度的预测依赖人工智能技术 **+大数据技术**
- 以下关于该系统中数据的采集、加工和处理的说法，合理的是（**B**）
 - 该系统中的数据全部来源于传感器采集
 - 同一传感器采用不同算法能实现不同的功能
 - 传感器采集数据实现了数据的“数模转换”
 - 在气象数据加工处理过程中不会产生新的数据
- 以下关于台风防治系统功能和设计的说法，不正确的是（**D**）
 - 该系统可采用B/S 架构，以方便社会大众使用
 - 该系统在应对复杂自然环境时存在局限性
 - 该系统可预留空间以过渡到地震灾害的防治工作
 - 该系统的数据库设计需要在概要设计阶段完成 **数据库设计 属于详细设计**
- 以下关于该系统中人工智能技术应用的说法正确的是（**B**）
 - 深度学习需要事先手工构造知识库
 - 历史气象数据越丰富，学习模型越强大
 - AI 模型能实现对台风路径的精准预测
 - 台风路径预测精度不依赖实时数据的质量
- 现有一个台风预测模型，可根据中心气压 (P) 和最大风速 (V) 两个关键变量，对台风强度进行初步定性：
 - 若中心气压高于 990 hPa，则判定为“热带风暴”
 - 若中心气压低于 990 hPa，且最大风速小于 33 m/s，则判定为“强热带风暴”
 - 若中心气压低于 990 hPa，且最大风速在 33-50 m/s 之间，则判定为“台风”
 - 若中心气压低于 990 hPa，且最大风速大于或等于 50 m/s，则判定为“强台风”
 据此编写 python 程序和绘制流程图如下，其中定性台风强度有误的一项是（**B**）

<p>A.</p> <pre> if p > 990: print("热带风暴") elif v < 33: print("强热带风暴") elif v < 50: print("台风") else: print("强台风") </pre>	<p>B.</p> <pre> if p > 990: print("热带风暴") else: if v < 33: print("强热带风暴") if v < 50: print("台风") if v >= 50: print("强台风") </pre>
<p>C.</p>	<p>D.</p>

8. 非循环队列 q 初始状态如图 a 所示, h 表示队首指针, t 表示队尾指针。元素可直接出队, 也可出队后再重新入队, 经过若干次操作后, 队列 q 的状态如图 b 所示, 则整个过程中元素“强”不可能出现的索引位置是 (C)

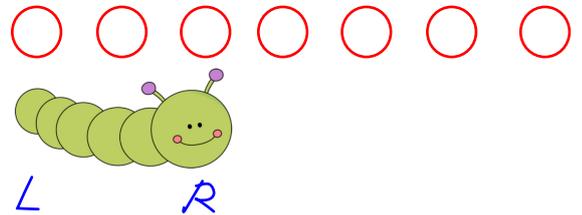


"强"、"基"出队后必入队

- A. 4, 5 B. 4, 6 C. 5, 6 D. 4, 5, 6
9. 某完全二叉树, 其前序遍历序列为“浙江强基联盟”, 以下说法不正确的是 (D)
- A. 其后序遍历序列为“强基江盟联浙” B. 该二叉树有 3 个叶子节点
C. 该二叉树根节点为“浙” D. 节点“联”的父节点为“盟”
10. 有如下 python 程序段

```
def f(a, k):
    n = len(a); left, ssum, mlen = 0, 0, 0
    for right in range(n):
        ssum += a[right]
        while left <= right and ssum > k:
            ssum -= a[left]
            left += 1
        if ssum <= k:
            mlen = max(mlen, right - left + 1)
    return mlen
a = [-5, 8, 7, -4, 5, 2]; k = int(input("k="))
print(f(a, k))
```

滑窗法求连续元素和小于等于的k的最长子序列



- 运行程序, 输入k 的值为 5, 输出结果为 (B)
- A. 2 B. 3 C. 4 D. 5
11. 使用二维数组 link 模拟链表结构, 每个节点数据依次为数据域和指针域, head 和 tail 分别指向链表的头节点和尾节点, 以下 python 程序实现对各节点数据的升序排序,

```
def sortL(head, tail):
    p = head; st = []
    while p != -1:
        while len(st) != 0 and link[p][0] < st[-1]:
            val = st.pop()
            link.append([val, -1])
            link[tail][1] = len(link) - 1
            tail = link[tail][1]
        st.append(link[p][0])
        p = link[p][1]
    return st
```

队列和栈的配合下, 实现升序排序最终结果在栈中

出栈, 重新入链 (入队)

入栈

```
link = [[2, 2], [4, 4], [8, 3], [3, -1], [5, 0], [1, 1]]; head, tail = 5, 3
print(sortL(head, tail))
```

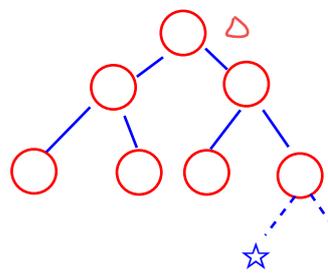
- 划线 (1) (2) (3) (4) 处可供选择的语句有 ①p!=-1 ②p!=tail ③p=link[p][1]
④tail=link[tail][1] ⑤val=st.pop() ⑥val=link[p][0]; 其填入正确的顺序是 (A)
- A. ①⑤④③ B. ①⑥③④ C. ②⑥④③ D. ②⑤③④

```
12. 有如下 python 程序, 实现在数组 a 中查找关键字 key,
a = [10, 15, 18, 25, 25, 26, 30, 33]
key = int(input("key=")) ; L, R = 0, len(a) - 1
m = (L + R) // 2
while L <= R and a[m] != key:
    if a[m] > key:
        R = m - 1
    else:
        L = m + 1
    m = (L + R) // 2 # 注释①
if L > R:
    print(-1)
else:
    print(m)
```

```

if L > R:
    print(-1)
else:
    print(m)

```



此时还有一次求m

运行程序，以下说法正确的是 (C)

- A. 输入 key 的值为 25，程序输出 4 **3**
- B. 输入 key 的值为 2025，程序报错：下标越界 **不报错，返回-1**
- C. 注释①语句改为 $m=(L+R+1)//2$ ，输出结果不变 **不影响对分查找结果**
- D. 输入 key 值为任意整数，注释①语句至多执行 3 次 **4**

二、综合题（本大题共 3 小题，第 13 题 9 分，第 14 题 8 分。）

13. H 市在水稻种植区部署智能设备，构建病虫害监测与防治系统，实现农田数字化管理。该系统利用虫情测报灯诱集害虫，并利用高清摄像头拍照识别采集害虫数据；同时，借助土壤传感器采集土壤温湿度数据；智能终端连接各类传感器，并通过 5G 模块将数据上传至服务器。当服务器分析数据发现异常时，向农技人员发送警示信息，并通过智能终端控制测报灯闪烁。农技人员可通过浏览器随时查看系统数据。请回答下列问题：

- (1) 以下不属于该系统服务器功能的是 B （单选，填字母）
- A. 存储虫害数据
 - B. 控制系统数据上传速率
 - C. 向智能终端发送警示信息
- (2) 关于该系统的搭建与应用，以下说法正确的是 CD （多选，填字母）

- A. 在系统概要设计阶段，需要进行数据的输入输出设计 **详细设计**
- B. 服务器负责数据存储，智能终端负责数据处理
- C. 若智能终端故障，仍能通过浏览器查看历史数据
- D. 服务器和客户端之间可以进行双向通信

(3) 在稻田中的 n 块种植区域，分别安装虫情测报灯（编号为“s1”，“s2”……，“sn”），进行虫情监测与示警。监测与示警规则是：系统每隔 1 小时依次采集各区域害虫数据，统计害虫数量最多的区域（若同一时段出现多个相同的最大值，则同时记录）。若该区域害虫数量连续 3 小时最多，则该区域的虫情测报灯闪烁示警。已发出警报区域不再进行害虫数量监测，直至农技人员操控无人机完成该区域除虫工作。部分 Python 程序代码如下，输出界面如 13 题图所示，请在划线处填入合适的代码。

```

n = 20      # 种植区块数量
insects = [0] * n
alarm = [False] * n
cx = [0] * n
while True:

```

区域s9连续3小时害虫数量最多，虫情测报灯示警闪烁！
区域s7连续3小时害虫数量最多，虫情测报灯示警闪烁！

每隔 1 小时采集各区域害虫数据，并存储在 insects 列表中，代码略
maxc = -1

for i in range(n): **insects [50,1,1,.....,1] n个数据**
 areas = [] **没警报过，状态就是False**

if not alarm[i] and insects[i] >= maxc:

if insects[i] > maxc:
 maxc = insects[i]
 areas = [i] **有新的最大值出现时，之前的全部推翻不要**
 areas.append(i)

for i in range(n): **例如本小时最高的区域为：areas [A,C]**
 if i in areas: **则其他区域比如 B区 计数器就清零**
 cx[i] += 1

else:
 cx[i] = 0 **一旦不是连续在最大害虫数量的区域中出现，就归零**

for idx in areas:
 if cx[idx] >= 3:
 alarm[idx]=True

print(f"区域 s{idx + 1} 连续 3 小时害虫数量最多，虫情测报灯示警闪烁！")

检测示警闪烁区域是否完成除虫，若检测完成，对应区域 alarm 值为 False，代码略

(4) 系统运行一段时间后，浏览器上观察到某区域害虫数量连续 3 小时最多，但相应虫情测报灯并未示警闪烁，**经排查系统中硬件连接均正常**，出现该现象的原因可能是 **该区域虫情测报灯故障；服务器（智能终端）启动该区域的虫情测报灯**。（注：回答 2 项，一项正确得 1 分）

14. 小张收集了 1896-2024 年历届奥运会奖牌数据，并将其存储在 data.xlsx 文件中，部分数据如第 14 题图 a 所示。将该数据读入到 DataFrame 对象 df 中，请回答下列问题：

xxxxx连接异常？
智能终端故障？
传感器故障？

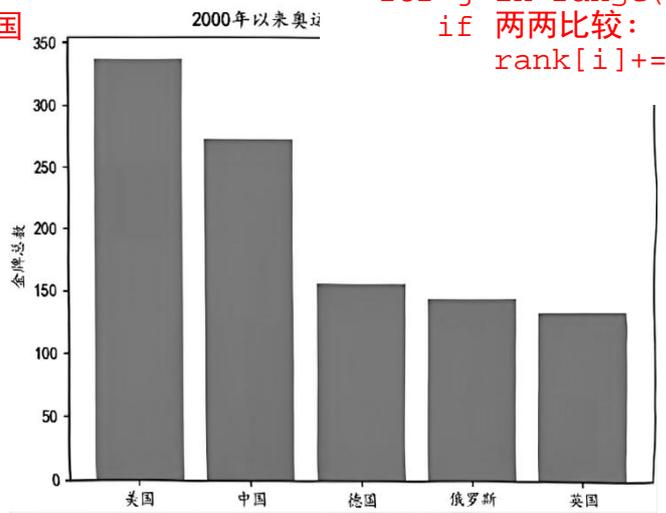
1	奥运会年份	国家	金牌	银牌	铜牌	总数	年份
2	1896年雅典奥运会	美国	11	7	2	20	1896
3	1896年雅典奥运会	希腊	10	18	19	47	1896
4	1896年雅典奥运会	德国	6	5	2	13	1896
367	1952年奥斯陆冬奥会	挪威	7	3	6	16	1952
368	1952年奥斯陆冬奥会	美国	4	6	1	11	1952
369	1952年奥斯陆冬奥会	德国	3	2	2	7	1952
370	1952年奥斯陆冬奥会	芬兰	3	4	2	9	1952
1786	2022年北京冬奥会	白俄罗斯	0	2	0	2	2022
1787	2022年北京冬奥会	西班牙	0	1	0	1	2022
1788	2024年巴黎奥运会	美国	40	44	42	126	2024
1789	2024年巴黎奥运会	中国	40	27	24	91	2024
1790	2024年巴黎奥运会	日本	20	12	13	45	2024
1791	2024年巴黎奥运会	澳大利亚	18	19	16	53	2024

第 14 题图 a

(1) 若要在 df 中新增名次列, 计算当前金牌数的排名(相同金牌数排名相同), 定义 cal 函数, gold 为当前国家的金牌数, gold_list 为同一届奥运会所有国家的金牌数列表。如图 a 中 2024 年巴黎奥运会数据, 美国、中国、日本、澳大利亚的排名分别 1、1、3、4, 请在划线处填写合适的代码。

```
def cal(gold, gold_list):
    rank = 1
    for i in gold_list:
        if i > gold:
            rank += 1
    return rank
```

所有国家金牌数排名:
for i in range(n):
 for j in range(n):
 if 两两比较:
 rank[i] += 1



第 14 题图 b

(2) 统计 2000 年以来奥运会金牌总数最多的五个国家, 降序排序并绘制柱形图, 如第 14 题图 b 所示。实现上述功能的部分 python 代码如下, 请选择合适的代码依次填入划线处(填字母)。

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_excel("data.xlsx")
df1= D ①
df1= B ②
df2= E ③
```

```
df2=df2.head() # 获取前 5 条数据
plt.bar( G ④ )
```

设置绘图参数, 并显示如图 b 所示柱形图, 代码略①②③④处可选代码有:

- A. df1.groupby("国家", as_index=False).sum()
- B. df1.groupby("国家").sum()
- C. df1[df1["年份"]>=2000]
- D. df[df["年份"]>=2000]
- E. df1.sort_values("金牌", ascending=False)
- F. df1.sort_values("总数", ascending=False)
- G. df2.index, df2.金牌
- H. df1.国家, df1.金牌

1	奥运会年份	国家	金牌	银牌	铜牌	总数	年份
2	1896年雅典奥运会	美国	11	7	2	20	1896
3	1896年雅典奥运会	希腊	10	18	19	47	1896
4	1896年雅典奥运会	德国	6	5	2	13	1896

(3) 分析数据后发现不同国家在夏季奥运会(简称奥运会)和冬季奥运会(简称冬奥会)上的表现有较大差异, 程序加框处实现从“奥运会年份”列数据中提取最后三个字符(分别为“奥运会”或“冬奥会”)区分不同奥运会类型的功能, 下列选项可行的有 CD (多选, 填字母)。(注:全部选对的得 2 分, 选对但不选全的得 1 分, 不选或有选错的得 0 分)

```
df.insert(0,"奥运会类型","") # 插入奥运会类型列
```

```
df["奥运会类型"] = df["奥运会年份"][-3:]
```

获取最后三行

- A. df["奥运会类型"] = df["奥运会年份"][-3:]
- B. for i in df: 默认df.columns
df.at[i,"奥运会类型"] = df.at[i,"奥运会年份"][-3:]
- C. for i in df.index:
df["奥运会类型"][i] = df["奥运会年份"][i][-3:]
- D. for i in range(len(df)):
df.at[i,"奥运会类型"] = df.at[i,"奥运会年份"][-3:]