

高三上信息选考练习卷12

第一部分 信息技术（共 50 分）

一、选择题（本大题共 12 小题，每小题 2 分，共 24 分。在每小题给出的四个选项中，只有一个符合题目要求）

阅读下列材料，回答第1至3题：

某医院的自助系统，患者可利用手机在线挂号、缴费、查看病历、查看明细等服务。系统能智能分析患者CT影像，自动识别病灶并为医生生成初步诊断参考报告。

1. 下列关于该自助系统中数据和信息的说法，正确的是 **D**
 - A. 系统中的数据只有文字这一种表现形式
 - B. 患者的相关信息均存储在患者手机中
 - C. 患者 CT 影像与文本报告的编码方式一样
 - D. 利用数据库管理，有利于数据共享与追踪
2. 下列措施中，不能有效提升该信息系统数据安全的是 **A**
 - A. 程序编写人员预留后门以便远程操控
 - B. 对存储的患者病历信息进行加密处理
 - C. 患者与医生登录系统时使用动态口令
 - D. 安装防火墙并记录数据库的访问操作日志
3. 下列关于该系统中人工智能技术的说法，正确的是 **B**
 - A. AI 的诊断结果可直接作为最终的临床诊断
 - B. 模型训练过程中，常使用卷积神经网络提取图像特征
 - C. 智能识别病灶时，需要人工标注 CT 影像数据
 - D. 从图像输入到诊断输出，需要事先人工构造知识库

阅读下列材料，回答第 4 至 7 题。

某新能源汽车的车载智能系统集成北斗高精度定位、5G 支持的车联网通信技术、NFC 识别模块、蓝牙技术，实现导航、音乐播放等功能。摄像头与传感器实时监测驾驶员和车辆状态，识别危险驾驶模式，触发语音提醒，并将视频传输至平台。借助手机，用户走近车辆时，车门会自动解锁，还能查看车辆定位，行驶轨迹等。

4. “用户走近车辆时，车门自动解锁”，实现此功能，下列选项中最适合的是 **C**
 - A. 加速度传感器
 - B. 距离传感器
 - C. 蓝牙
 - D. NFC
5. 关于该系统的组成与功能的说法，正确的是 **C**
 - A. 系统数据的采集都是由摄像头实现
 - B. 平台采集到的视频属于结构化数据
 - C. 该系统中的用户包括驾驶员
 - D. 智能系统的运行不需要系统软件的支持
6. 关于该系统的网络及网络技术的说法，正确的是 **A**
 - A. 视频数据传输需要 IP 协议的支持
 - B. 车载终端与平台的数据传输不需要传输介质
 - C. 平台的网络故障会导致新能源汽车无法行驶
 - D. 用户只能通过 5G 网络查看车辆定位
7. 关于该系统采集的视频在存储和传输时的说法，不正确的是 **D**
 - A. 摄像头的分辨率会影响采集视频的清晰度
 - B. 存储在车载终端的视频是经过编码后的二进制数字
 - C. 不同视频格式在画面质量、文件大小方面存在差异
 - D. 为了节省存储空间，视频应采用 AVI 格式

AVI 一般默认是未经压缩的视频文件，文件比较大

8. 某包含 6 个节点的完全二叉树，其根节点在中序遍历序列中的位置记为 x，在后序遍历序列中的位置记为 y，则 x-y 的值为 **C**
- A. -1 B. 1 C. -2 D. -3

9. 循环队列的存储空间为 6，head 指向队首元素所在位置，tail 指向队尾元素后一个位置，初始时队列为空，head=0，tail=0。现在依次执行以下操作：入队 a、b、c，出队一个元素，入队 d、e、f，出队两个元素，入队 g、h。此时队列 head 和 tail 为 **A** 入8个，出3个
- A. head = 3, tail = 2 B. head = 2, tail = 3
C. head = 1, tail = 2 D. head = 2, tail = 1

10. 如下程序实现的功能为去除字符串中的重复字符，对于重复出现的，仅保留首次出现位置的字符。

```
s = input("输入字符串")
```

```
i = 1; j = len(s)
```

```
while ① i <= j :
```

```
    if s[i] in s[:i]:
```

```
        s = s[:i] + s[i+1:j]
```

```
        j = len(s)
```

```
        i -= 1 ②
```

```
    i += 1
```

```
print("去重后字符串: ",s)
```

划线处应填的正确代码为 **B**

- A. ①i<=j ②i-=1 B. ①i<j ②i=1 C. ①i<=j ②j-=1 D. ①i<j ②j-=1

11. 定义如下函数

```
def dsearch(data, key):
```

```
    i = 0; j = len(data) - 1
```

```
    while i <= j:
```

```
        m = (i+j)//2
```

```
        if key > data[m]:
```

```
            i = m + 1
```

```
        else:
```

```
            j = m - 1
```

```
    return i
```

当 key 值为 9，使用以下 data 值，调用函数后，返回结果与其他几项不同的是 **D**

- A. [1,3,6,10,11,12] B. [1,3,6,9,10,11,12]
C. [1,3,6,9,9,11,12] D. [1,3,9,9,10,11,12]

12. 使用列表 d 模拟链表结构（节点数 n>0 且 n 为偶数），每个节点含数据区域和指针区域，head 为头指针。按数据区域首字母分为 A 类、B 类节点。要将链表从如图 a 所示“ABAB...”的顺序，调整为如图 b 所示“BABA...”的顺序。实现上述功能 Python 程序段如下：

```
p = head
head = d[p][1] 更新head 单独处理
```

```
while p != -1:
```

```
    q = d[p][1]
```

```
    r = d[q][1]
```

```
    d[q][1] = p ①
```

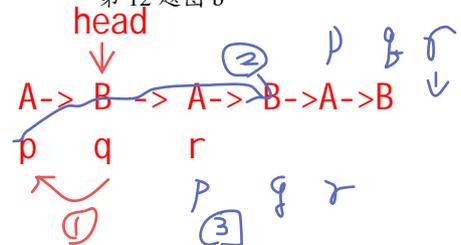
方框中应填入的正确代码为 **B**

	数据域	指针域
head→0	A35	3
1	B26	-1
2	A30	1
3	B28	4
4	A33	5
5	B27	2

第 12 题图 a

	数据域	指针域
0	A35	5
1	B26	2
2	A30	-1
head→3	B28	0
4	A33	1
5	B27	4

第 12 题图 b



- A. if d[q][1] == -1:
 d[p][1] = -1
 else:
 d[r][1] = d[p][1] ~~X~~
 p = d[p][1] ~~X~~
- B. if r == -1:
 d[p][1] = -1
 else:
 d[p][1] = d[r][1] ~~X~~
 p = r ~~3~~
- C. if d[q][1] == -1: **等同于 r == -1**
 d[p][1] = -1
 else:
 d[r][1] = d[p][1] ~~X~~ **d[p][1] = d[r][1]**
 p = r
- D. if r == -1:
 d[p][1] = -1
 else:
 d[p][1] = d[r][1]
 p = d[p][1]

二、非选择题 (本大题共 3 小题, 其中第 13 小题 8 分, 第 14 小题 9 分, 第 15 小题 9 分, 共 26 分)

13. 某小组搭建噪音监测系统采集若干自习室的声音数据。每个自习室部署一个监测点, 智能终端连接传感器。每隔 1 分钟采集一次声音, 若未超过阈值 50 分贝用 0 值表示, 超过的用实际的分贝值表示, 数据通过 5G 模块上传至服务器。当收到连续 3 分钟及以上的数据都超过阈值, 服务器通过智能终端控制蜂鸣器发出声音, 直到收到停止的指令, 蜂鸣器停止发声, 此过程视为 1 次预警。小组成员可通过浏览器查看系统数据。请回答下列问题:

- (1) 将未超过阈值的数据用 0 表示是在 ▲ B 处完成 (单选, 填字母)。
 A. 传感器 B. 智能终端 C. 服务器

- (2) 关于该信息系统的搭建, 下列说法正确的是 ▲ B (单选, 填字母)。

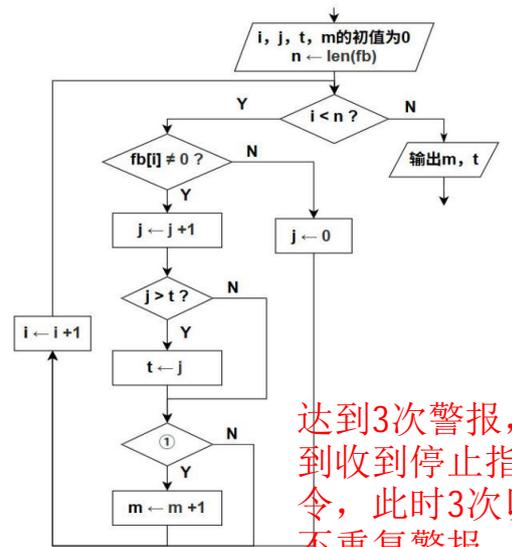
- A. 系统采用 B/S 架构, 其优势是服务器的负荷量轻
 B. 若智能终端连接的 5G 模块损坏, 小组成员与服务器之间的通信不受影响
 C. 智能终端通过 URL 向服务器传输数据时, 需要知道服务器端的数据库文件名 **只需要知道服务器 ip**

浏览器访问服务器正常

- (3) 系统搭建完成后, 在测试时发现某自习教室在安静的情况下, 蜂鸣器会突然发出预警, 可能的原因有 ▲ ABD (多选, 填字母)。(注: 全部选对的得 2 分, 选对但不全的得 1 分, 不选或有选错的得 0 分)

- A. 传感器存在故障
 B. 蜂鸣器存在故障
 C. 智能终端没有传输数据至服务器
 D. 服务器判定连续噪音的算法存在漏洞

- (4) 将某自习室一天的声音数据存储在列表 fb 中, 统计蜂鸣器发出的预警次数 m 和持续噪音的最长时长 t, 部分流程图如右图所示。图中①处应填入 ▲ j=3? 或 j==3?



达到3次警报, 直到收到停止指令, 此时3次以上不重复警报,

- (5) 系统正常运行一段时间后, 需要新增更人性化的语音提示, 如“同学们, 请保持安静哦”。请增加设备, 并描述如何实现该功能。 ▲ 增加: 扬声器、小喇叭、音响

(注: 保持安静的语音提示文件已存储于智能终端) **功能实现: 扬声器连接到智能终端, 通过服务器发送指令给智能终端控制扬声器播放声音文件。**

14. 小王收集了某城市的天气数据。现要对这些数据进行分析，请回答下列问题：

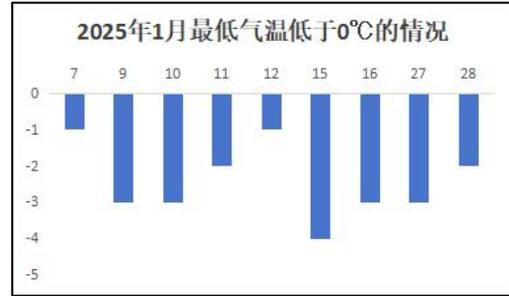
(1) 该城市的天气数据存于 weather.xlsx 文件中，如图 a 所示。要找出最低气温平均值最小的月份，并统计该月最低气温低于 0°C 的情况，绘制如图 b 所示的柱形图。实现上述功能的部分 Python 程序如下，请选择合适的代码填入划线处（单选）。

年	月	日	最低气温	最高气温
2024	10	1	17	22
2024	10	2	15	23
2024	10	3	14	24
2024	10	4	18	25

.....

2025	9	26	25	34
2025	9	27	25	33
2025	9	28	24	34
2025	9	29	23	29
2025	9	30	22	28

第 14 题图 a



第 14 题图 b

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel("weather.xlsx")
df1 = E ①
df2 = df1.sort_values("最低气温",ascending=True)
#将 df2 首行的月份存入 m，代码略
print("平均最低气温最低月份序列为：",m)
df3 = B ②
df4 = df3[df3.最低气温<0]
plt.bar( C ③ )
# 设置绘图参数，显示如图 b 所示的柱形图，代码略
```

- A. df2[df2.月==m]
- B. df[df.月==m]
- C. df4.日, df4.最低气温
- D. df4[日], df4[最低气温]
- E. df.groupby("月", as_index=False).最低气温.mean()
- F. df.groupby("最低气温", as_index=False).月.mean()

(2) 某地 2024 年 10 月至 2025 年 9 月的各月最高气温的平均值存储在列表 w 中。要找出相邻 2 个月，其最高气温的平均值相差最大。如果有多个符合要求的结果，只保留距今最近的那个结果。实现上述功能的部分 Python 程序段如下，请在划线处填入合适的代码。

```
month=[202410,202411,202412,202501,202502,202503,202504,202505,202506,202507,202508,202509]
imax = 1 ①
for i in range(2, len(w)):
    if abs(w[i] - w[i-1]) >= abs( w[i] - w[0] ② ):
        imax = i
print("最高气温的平均值相差最大的相邻月份中，其第一个月份是：", month[imax-1] ③ )
```

15. 某选课冲突检测系统用于检测每个学生选课过程中可能存在的选课冲突。选课冲突包括时间冲突和先修课程冲突。部分课程有先修课程要求，该要求是指在选该课程前必须已完成或在本学期同时选修这些先修课程。

- 时间冲突：同一时间段的课程之间存在时间冲突；
- 先修课程冲突：如果没有先修课程要求，则该课程不存在先修课程冲突。如果有先修课程要求，需要逐个检查这些先修课程：
 - ①此先修课程涉及的先修课程是否存在先修课程冲突；
 - ②若此先修课程未完成，且本学期未选修，则存在先修课程冲突。

请回答下列问题：

图 a 为某学生的选课信息，包含课程 id、时间段和先修课程，输出如图 b 所示。

课程 id	时间段	先修课程
103	A	102
105	B	101, 104
108	C	105, 107
109	A	无
110	A	无

104

D
第 15 题图 a

已完成课程: [101, 102]
冲突检测结果:
课程103与109在时间段A存在时间冲突 课程103与110在时间段A存在时间冲突 课程109与110在时间段A存在时间冲突
课程105的先修课程要求: 105→104, 缺少先修课程104 课程108的先修课程要求: 108→107, 缺少先修课程107 课程108的先修课程要求: 108→105→104, 缺少先修课程104

保留

不再冲突

不再冲突

第 15 题图 b

- (1) 若某同学在图 a 所示的选课基础之上新选了一门课程 id 为 104、时间段为 D、先修课程无要求的课程，则检测结果中共有 4 ▲ 条冲突。
- (2) 函数 `check_time` 的作用是遍历课程列表数据 `nodes`，找出其中存在的时间冲突。请在划线处填写正确的代码。

```
def check_time():# 检测时间冲突
```

```
    tgroup = {}
```

```
    for node in nodes:
```

```
        if node[1] not in tgroup
```

没有该键的存在，新建键值对

```
            tgroup[node[1]] = []
```

```
            tgroup[node[1]].append(node[0])
```

```
    return tgroup
```

tgroup={"A": [103, 109, 108], "B": [105],}

- (3) 实现其余功能的部分 Python 程序如下，请在划线处填入合适的代码。

```
def find(cid): # 查找课程在 nodes 中的索引
```

```
    i = 0
```

```
    while i < len(nodes):
```

```
        if nodes[i][0] == cid:
```

```
            break
```

```
        i+=1
```

```
    return i
```

如果能找到返回cid在nodes中的下标

找不到i值等于len(nodes)

```
def check_pre(cid): # 检测某课程的先修课程冲突
```

```
    st = [0]*100
```

```
    top = -1
```

```
    top += 1
```

```
    st[top] = [cid, []]
```

入栈，st栈中的元素格式：[课程，[]]

```
    all_path = []
```

```
    result = True
```

```
    while top!=-1 :
```

```

item = st[top] ① item中[先修课程, [后修课程]]
top -= 1 出栈
new_path = item[1] + [item[0]] 第一次执行时new_path=[]+[cid]
i = find(item[0]) 在nodes中找cid, 即找是否有报cid课程
node = nodes[i] 此时cid课程一直是存在的, 获取该课程在nodes中的元素值
pre_courses = node[2] pre_course中存放cid的先修课程
for pcid in pre_courses:
    if pcid in completed: 如果先修课程在已完成课程中, 直接跳过
        continue
    i = find(pcid) 找先修课程本学期是否选修
    if ② i >= len(nodes): 不能用pcid not in nodes, nodes中元素是列表
        all_path.append(new_path + [pcid]) new_path最开始是[cid], 现在加上[pcid]
        result = False 未选修
    else: 如果本学学期已选修
        top += 1 入栈 [先修课程, [后修课程]]
        st[top] = [pcid, new_path] [pcid, [cid]]
return result, all_path

```

```

def check_course(): # 按选课信息检测先修课程冲突
    courses_conf = [] node
    for node in nodes: nodes=[[103, 'A', [102]], .....]
        cid = node[0] cid=103 (cid中是课程)
        if node[2] != []: 如果先修课程不为空
            result, paths = ③ check_pre(cid)或 check_pre(node[0]) 检测先修课程冲突
            if result == False: 先修课程未选修, 冲突 反
                for path in paths:
                    courses_conf.append(["课程冲突", cid, path])
    return courses_conf

# 主程序
# 选课信息存入列表 nodes, 每个元素包含课程 id、开课时间段和先修课程 3 个数据, 代码略
nodes = [[103, 'A', [102]], # 其他课程数据略]
completed = [101, 102] # 已完成课程列表
ct = check_time() 返回类似结果 tgroup={"A": [103, 109, 108], "B": [105], .....}
cc = check_course()
# 输出结果, 代码略

```