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

高三下信息选考练习卷8

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 6 题。

某街道引入的"AI 新同事"社区垃圾分类管理系统，在各社区部署分布式智慧局域网，在关键点位安装带计算能力的摄像终端，提取视频中的异常事件并上传到服务器，并在服务器部署了某 AI 大模型进行本地化分析。该系统能实时智能识别垃圾桶满溢、垃圾暴露、违规投放等异常事件，服务器生成告警工单派发至保洁员手机 APP。同时，AI 可自动生成涵盖问题类型、高发时段、清运频率的日报、周报，为社区精准管理提供数据依据。

1. 下列关于该系统中数据的说法，不正确的是 C

A. 告警数据在计算机内部都是以二进制形式存储的

B. 摄像终端采集到的数据主要为非结构化数据

C. 保洁员手机 APP 中收到的告警工单是未经任何加工的原始数据

D. 系统识别的"高发时段"等数据，可作为社区调整清运人力与频次的依据

2. 下列关于该系统组成和功能的说法，正确的是 C

A. 摄像终端、服务器和保洁员的手机 APP 均属于该系统的硬件

B. 部署在本地用于智能识别的 AI 大模型属于系统软件

C. 该系统最大的局限性是对算力、电力等外部环境的依赖 还有两点局限性：软件自身缺陷、数字鸿沟

D. 该系统对数据的处理全部在服务器端完成

3. 为保障"AI 新同事"系统安全稳定运行，下列措施中，不合理的是 B

A. 对摄像头采集的视频流进行加密后传输

B. 增加服务器内存容量以便定期备份视频数据 外存（硬盘或辅存）容量

C. 为本地服务器部署防火墙，并安装杀毒软件

D. 仅将摄像头捕捉到的异常事件数据传至服务器，以减轻服务器压力

4. 下列关于该系统网络技术的说法，正确的是 D

A. 该系统的网络资源不包括硬件资源

B. 从采集数据到服务器生成工单的过程无需网络软件支持

C. 保洁员手机可以通过 RFID 技术连接局域网

D. 服务器网络故障会影响保洁员手机 APP 接收告警工单

5. 该系统的下列应用中，运用了人工智能技术的是 D

A. 将告警工单即时推送到保洁员手机 APP 中

B. 统计"违规投放"等异常事件的高发时段

C. 为不同类型的告警工单在 APP 中标注不同的颜色

D. 从视频画面中识别出"违规投放"、"垃圾暴露"等异常事件

6. 该系统的摄像终端将采集到的画面存储为 BMP 格式文件，下列说法不正确的是 C

A. 图像数字化过程是将模拟信号转换为数字信号

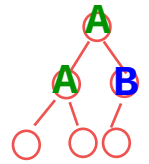
B. 颜色位深度会影响该图像文件的容量大小

C. 图像画面越复杂，其存储容量就越大

D. 为减轻网络传输负荷，将 BMP 格式图像转换为 JPEG 格式

7. 某算法的部分流程图如第 7 题图所示，若数组元素 a[0]至 a[6]依次存放 8,6,9,7,7,6,5，执行这部分流程后，输出 t 的值为 B

A. 25　　B. 18　　C. 13　　D. 11



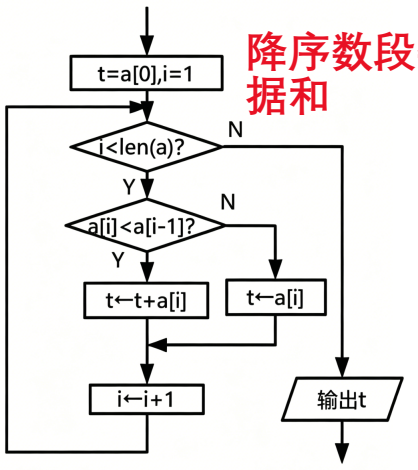
B一定在最后
A可以在第2、4位

8. 某完全二叉树有 A、B、C、D、E、F 六个节点，其中 A 节点的度为 2，B 节点的度为 1，则该完全二叉树的中序遍历结果可能是

A. DEFACB　　B. DACBEF　　C. ACDFEB　　D. EDFABC　　A

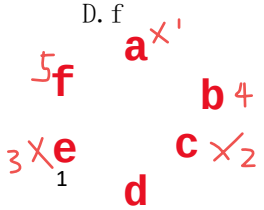
9. 有一个循环队列，队首到队尾的元素依次为 a,b,c,d,e,f。现进行如下操作：先将队首元素出队，再将新的队首元素出队后再入队，重复以上两个步骤，直到队空，则最后一个出队的元素为

A. b　　B. d　　C. e　　D. f　　B



降序数段数据和

第 7 题图



10.有如下 Python 程序段：

```
result, flag = "" , 1
for ch in s:
    if flag == 1 and "a"<=ch<="z":
        result += chr(ord(ch)-32)
    elif ch =="#":
        flag = 1 - flag
    elif flag == 0:
        result += ch
```

flag值为1，只对小写字母操作，其他字符不做操作

若字符串 s 的值为"ab#cd#23e#!"，执行该程序段后，变量 result 的值为 A

A."ABcdE!"　　B."AbcdE!"　　C."ABCDE"　　D."aBcdE!"

11.有如下 Python 程序段：

```
top1 = len(s1)-1 ; top2= -1 ; k=0
while top1!= -1:
    temp = s1[top1]
    top1-=1
    k+=1
    while top2 != -1 and temp < s2[top2]:
        top1 += 1
        s1[top1] = s2[top2]
        top2-=1
    top2 += 1
    s2[top2] = temp
```

s2构建递增栈，构建过程中出栈的元素入s1栈

已知栈 s1、s2 最大容量均为 n，s1 有 n 个元素入栈，s2 初始为空，若 s1 初始值为['110','98','105','99','33']，下列关于该程序段的说法，正确的是 D

A.该算法的时间复杂度为 O(2^n) $O(n^2)$

B.执行该程序段后，s2 的值为['33','98','99','105','110']['105','110','33','98','99']

C.对任意 s1 的值，变量 k 的值一定大于 n　也可能等于n

D.执行该程序段后，s1 的值为['99','98','33','33','33']

12.现有一个 m 行,n 列的数字矩阵，如 a=[[4,3,2,-1],[3,2,1,-1], [1,1,-1,-2], [-1,-1,-2,-3]]矩阵中的元素每行每列都是按照非升序排列，如第 12 题图所示，实现统计该矩阵中小于 0 的数字个数的 Python 程序段如下，划线处应填入的正确代码是 A

```
def f(a):
    m = len(a)
    n = len(a[0])
    i,j=0,n-1
    cnt =0
    while i< m and j>= 0:
        if a[i][j]<0:
            ①
            ②
        else:
            ③
    return cnt
```

4	3	2	-1
3	2	1	-1
1	1	-1	-2
-1	-1	-2	-3

第 12 题图

i是行号，从上往下
j是列号，从左往右
如果a[i][j]<0,意味着i行下面的同一列中的值都小于0，可以批量把m-i个负数一起统计，即cnt=m-i

A. ①cnt += m-i　②j -= 1　③i += 1

B. ①cnt += n-j　②j -= 1　③i += 1

C. ①cnt += 1　②i += 1　③j -= 1

D. ①cnt += m-i-1　②i += 1　③j -= 1

二、综合题

13.某厂房部署火情预警和喷淋系统，在厂房的多个监测位置采用智能终端连接烟雾传感器、温度传感器，每 5 分钟采集一次环境烟雾浓度和温度数据，通过 IoT 模块以无线通信方式将采集的数据传输至服务器，并存储到数据库中。服务器根据数据判断，若有火情危险将通过终端控制执行器发出预警信号和启动消防喷淋，用户可通过浏览器查看系统数据。请回答下列问题：

(1)下列符合硬件连接的是_____B_____(单选)。

A.传感器和执行器可连接在同一智能终端的同一引脚上

B.多个监测位置可使用多个智能终端连接传感器采集环境数据

(2)关于该系统中数据传输与处理的说法，正确的有_____CD_____(多选)。(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.传感器与智能终端的数据传输是双向的

B.该系统中服务器与数据库之间的数据传输是单向的

C.服务器可以分别处理来自智能终端和浏览器的数据

D.系统正常运行一段时间后，若服务器损坏，用户将无法查看系统的历史数据

(3)系统运行后，在没有火情的情况下，执行器经常发出预警信号，为适当减少系统对火情状态的误判，下列做法有效的有____BC____(多选)。(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.提高传感器采集数据的频次　　B.适当调整烟雾浓度或温度的报警阈值

C.对采集数据进行多条件联合判断　　D.增加服务器的存储容量

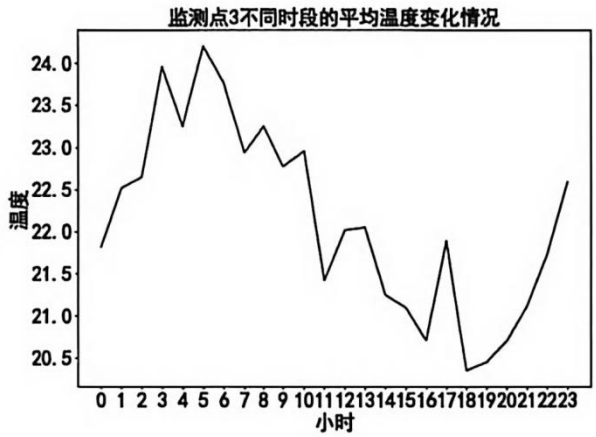
(4)系统正常运行一段时间后，若某个监测点 IoT 模块与智能终端的连接断开，请写出 2 种可能引发的问题。

1.该监测点传感器数据无法实时上器的指令。2.该监测点智能终端无法接收来自服务数据3.用户无法查看该监测点的实时传到服务器。

(5)将某天各监测点的烟雾浓度和温度数据导出到文件 fire_data.xlsx 中。部分数据如第 13 题图 a 所示。现要找出当天平均烟雾浓度最高的监测点，并统计该监测点不同时段的平均温度变化情况，绘制如第 13 题图 b 所示的线形图。

	A	B	C	D	E
1	监测点	时	分	浓度	温度
2	1	0	0	7	22
3	1	0	5	1	20
4	1	0	10	17	24
5	1	0	15	25	21
6	1	0	20	14	19
7	1	0	25	8	22
55	3	23	25	5	22
56	3	23	30	14	20
57	3	23	35	6	20
58	3	23	40	24	24
59	3	23	45	17	21
60	3	23	50	29	22
61	3	23	55	23	21

第 13 题图 a



第 13 题图 b

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)。

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel("fire_data. xlsx") #读取文件
df1 = df.groupby("监测点",as_index=False).浓度.mean()#分组计算
df2=__①__F__
id = df2.values[0][0]
df3 =__②__A__
df4 =__③__D__
#设置绘图参数，选取 df4 中的数据创建图表，显示如图 b 所示的线形图，代码略
```

程序中①②③处可选的代码有：

A.df[df["监测点"]==id]

B.df1[df1["监测点"]== id]
C.df3.groupby("分",as_index=False).温度.mean()
D.df3.groupby("时", as_index=False).温度.mean ()
E. df1.sort_values("浓度", ascending=True) #升序排序
F. df1.sort_values("浓度", ascending=False)

14.该厂房部署了火情预警和喷淋系统，根据历史数据建立基准线，将烟雾浓度超过基准线的采集时刻判定为异常波动点，异常波动可能预示火情风险。现需要对监测点 3 某天的历史数据进行分析，找出异常波动点最密集的连续 1 小时时段(即连续 12 个采集时刻)，并输出该时段的起始时间。如果有多个时段异常点数量相同，取最早出现的时段。
(1)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。

```
'''
读取该天采集的烟雾浓度、温度等数据按时间顺序存储在列表 a 中，如
a=[[0, 0, 7, 22], [0, 5, 1, 20], [0, 10, 17, 24]....., [23, 50, 29, 22], [23, 55, 23, 21]]，每项为[小时,分钟,烟雾浓度,温度]，代码略
'''
#计算烟雾浓度基准线并存储在变量 std 中，代码略
n = len(a)
flag =[0]*n
for i in range (n):
    if a[i][2]> std:
        flag[i]=1
start_time = 0
cnt = 0
for i in range(12):
    ①cnt+=flag[i]
max_cnt = cnt
for i in range(1, n -11):
    cnt = ②cnt-flag[i-1]+flag[i+11]
    if cnt > max_cnt:
        max_cnt = cnt
        ③start_time=i
if max_cnt>0:
    print("异常点最密集时段的起始时间:",a[start_time][0],"时",a[start_time][1],"分")
else:
    print("未发现异常点")
```

第一个小时的异常波动次数作为max_cnt的初值

滑窗求和
i~i+11为滑窗大小

(2)若将程序加框处代码更改为 flag[i] +=1，是否会影响程序功能?__B__ （单选，填字母：A.会/B.不会)

15.某影院推出"智慧观影"系统，该平台具有以下功能：
a)观众进入平台后，可查看影片综合评分排行榜。将本月所有影片按综合评分从高到低排序，并输出排行榜名单。
b)观众选择心仪影片后，系统根据实时状态更新座位信息，为观众呈现可选座位区间。初始状态所有座位均可选，当有观众购买某座位时，系统将该座位从可选区间中移除；当有观众退票时，系统将该座位合并到相邻的可选区间中，或创建新区间。

如某排共 10 个座位，购买座位 3、4、9 后，该排可选区间为[1,2],[5,8],[10,10]；
若座位 4 退票：与区间[5,8]左邻接，此时可选区间为[1,2],[4,8],[10, 10]；
若座位 3 退票：与区间[1,2]右邻接，且与[4,8]左邻接，三个区间合并为[1,8],[10,10]。
(1)某影院每排共 10 个座位，初始状态所有座位均可选。09:15 前座位状态变化记录如下所示(只记录发生变化的座位)。

时间	座位号	排数	状态
08:37	3	5	购票
08:50	5	5	购票
09:00	3	5	退票
09:12	8	5	购票

请根据记录，写出 09:15 时 5 排的可选座位区间(用[起始座位号，终止座位号]形式表示)_____[1,4],[6,7],[9,10]_____。
(2)定义如下函数，参数 b 列表中每个元素包含 4 个数据项，依次为电影名称、上映时间、影片类型、综合得分，参数 n 表示列表 b 的长度，如 b[0]=["天才游戏", "2026-04-04","悬疑/犯罪",8.3]。

```
def fsort(b, n):
    if n<2: #语句 1
        return b
    for i in range(n-1):
        if b[i][3]<b[i + 1][3]:
            b[i],b[i + 1]=b[i + 1],b[i]
    return fsort(b, n -1)
```

从前往后完成第n个数的排序，降序

下列关于该函数的说法，正确的是_____D_____(单选)
A.该函数功能为将列表 b 中的元素按综合得分升序排序
B.该函数功能为将列表 b 中的元素按上映时间降序排序
C.加框处代码修改为(n-2,-1,-1)，不改变函数功能
D.将语句 1 处代码中 n<2 改为 n<=0，不改变函数功能

降序

(3)系统可以实时处理购/退票信息，为观众呈现当前可选座位区间，其中处理购票功能的函数如下：

```
def order_seat (seat, row, lst):
    '''
    参数 seat 表示可购票座位号，row 表示购票排数，列表 lst 存放每个可选座位区间信息，每个元素包含 5 个数据项，依次为排数，可选座位起始号，可选座位终止号，可选座位数，下一可选座位区间的指针。
    '''
    pre=0
    p = lst[pre][4]
    while p!= -1 and lst[p][0]!= row:
        pre =p
        p = lst[p][4]
    #该排已有可选座位区间，查找 seat 的位置
    while p != -1 and lst[p][0] == row:
        if seat == lst[p][1] and lst[p][3]== 1:
            ①
        elif seat == lst[p][1]:
            lst[p][1]+=1
            lst[p][3]-=1
            return
        elif seat == lst[p][2]:
            lst[p][2]-=1
            lst[p][3]-=1
            return
        elif seat > lst[p][1] and seat < lst[p][2]:
            lst[pre][4]=p+1
            lst[p+1]=[row,seat+1,lst[p][2],lst[p][2]-seat,lst[p][4]]
            lst[p][2]=seat-1
            lst[p][3]=seat-lst[p][1]
            return
        pre=p
        p=lst[p][4]
```

找row行所在的第一个节点

①_____lst[pre][4]=lst[p][4]_____

seat是p节点开始的位置，且该区域只有一个位置，所以位置被售出后，该节点被删除

```
        elif seat == lst[p][1] and lst[p][3]>1:
            lst[p][1]= seat+1
            lst[p][3]-=1
        elif ____②____:
            lst[p][2] = seat-1
            lst[p][3]-= 1
        elif seat > lst[p][1] and seat < lst[p][2]:
            right = lst[p][2]
            lst[p][2]= seat-1
            ____③____
            lst. append([row, seat+1, right, right-seat, lst[p][4]])
            lst[p][4]= len (lst)-1
        pre =p
        p = lst[p][4]
```

seat是p节点开始的位置,该区域不只有一个位置

seat==lst[p][2] and lst[p][3]>1

seat是p节点结束的位置,该区域不只有一个位置

seat在中间，所以p节点要分成两个节点

lst[p][3]=seat-lst[p][1]

```
def add_seat (seat, row, lst):
    #处理观众退票信息，更新当前可选座位区间，代码略
#以下是主程序部分
'''
将影片信息存入 b 列表，每个元素包含 4 个数据项，依次为电影名称、上映时间、影片类型、综合得分，代码略
'''
n = len(b)
res = fsort(b, n)
#输出影片排行榜，代码略
#获取当前场次影厅的总排数存入整型变量 row，每排座位数存入整型变量 tot，代码略
lst = [[0, 0, 0, 0, 1]] #lst[0][4]指向头节点
for i in range(1, row+1):
    lst. append([i, 1, tot, tot, i+1]) #构造链表，初始化每排可选座位区间
lst[-1][4]= -1
'''
读入当前场次的退/购票信息，存入列表 data 中，每个元素包含 4 个数据项，分别是时间、座位号、排号、状态，data 中元素已按时间先后顺序升序排序，代码略
'''
for info in data:
    if info[3]=="购票":
        order_seat(info[1], info[2], lst)
    else:
        add_seat(info[1], info[2], lst)
#输出当前场次可选座位信息，代码略
```