## 台州市 2023 届高三第二次质量评估试题

技术

2023.04

信息命题: 天台中学 郑莎娜 温岭中学 朱海波 审题: 黄岩中学 黄 辉 通用命题: 黄岩二高 陈 湛 台州一中 徐 政 审题: 三门县教师发展中心 王小军 考生须知:

- 1. 本卷满分 100 分, 考试时间 90 分钟;
- 2. 答题前,在答题卷指定区域填写班级、学号和姓名,考场号、座位号写在指定位置;
- 3. 所有答案必须写在答题纸上,写在试卷上无效;
- 4. 考试结束后,只需上交答题纸。

## 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,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、错选、多选均不得分)
- 1.下列关于数据和信息的说法,正确的是
  - A. 图形符号不是数据
- C. 信息一般会随着时间的推移而变化
- 2.下列关于大数据的说法,不正确的是
  - A. 处理大数据时,一般采用分治思想
- 阅读下列材料,回答第3至6题。
- B. 模拟信号可以直接被计算机处理
- D. 人们对同一条信息的分析结果一定是相同的
- B. 社交网络数据可采用图计算进行处理

B. 该电子班牌不可外接存储设备

D. 电子班牌使用了人工智能和 RFID 技术

C. 利用大数据,可以对天气情况进行预测 D. 随着时间的流逝,流数据价值一般会随之提高

学校建设了智能班牌信息系统,服务器配备 UPS 不间断电源,每个教室门口悬挂电子班牌,该 班牌能展示班级文化、显示课程表:能兼容一卡通系统.刷卡进行余额查询、食堂订餐:能进行无 接触测温、人脸签到;还可以进行家校联系,家长可以在校外登录该系统查看学生信息。电子班牌 的部分参数如下表所示:

存储容量: 2GB, 可外接 SD 卡	USB 接口: 内置 USB2. 0 高速接口 2 个
操作系统: Linux/安卓(Android)	网络接口: 10/100M 以太网,内置 wifi
触摸屏: 16.5 英寸、分辨率 1360*768	识别:射频卡识别<0.2秒
摄像头: 130 万像素	音效: 语音提示

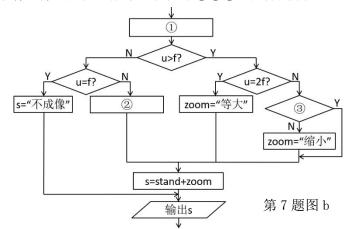
- 3.下列关于电子班牌的说法,不正确的是
  - A. 电子班牌属于硬件设备
  - C. Linux 和安卓都属于系统软件
- 4.下列关于该信息系统的说法,正确的是
  - A. 该系统由硬件、软件、用户和网络组成

  - B. 该系统的用户就是全校所有学生
  - C. 该信息系统不会感染计算机病毒
  - D. 服务器配备 UPS 不间断电源是因为信息系统对外部环境有依赖性
- 5.下列关于该信息系统数据的说法,正确的是
  - A. 无接触测温功能利用传感器采集温度数据
  - B. 电子班牌与外界只能通过 USB 接口进行数据传输
  - C. 电子班牌播出语音提示是模数转换过程
  - D. 定期备份服务器中的数据可以有效提升数据的保密性
- 6.下列关于该信息系统中网络的说法,不正确的是
  - A. 该系统不需要连接因特网

- B. 该系统可依靠网络实现数据共享
- C. 电子班牌进行网络通信需要遵循 TCP/IP 协议
- D. 电子班牌可以通过有线或无线的方式连接到网络
- 7.根据第7题图 a 所示的凸透镜成像规律, 第7题图 b 所示流程图中①②③处应分别填入



第7题图a



A.(1)stand="" zoom="" B.①stand="倒立" zoom="放大" C.(1)stand=""

D.①stand="倒立"

②stand="正立"

②stand="正立"

zoom="" ②zoom="放大"

②zoom="放大"

zoom="放大"

(3)u>2f?

(3)u < 2f?

(3)u>2f?

(3)u < 2f?

第8题图

8.如第8题图所示的二叉树,根节点为0,每个节点的左子节点为0,右子节点为1,每一条从根到 叶子的路径都组成一个二进制数。例如:从根到叶子 a 的路径组成二进制数 011,转换为十进制 数是3。若某完全二叉树共有13个节点,则它能表示的最大十进制数是

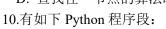
A.3

**B.4** 

C.5

D.6

- 9.下列关于单向链表的说法正确的是
  - A. 必定有头指针和尾指针
  - B. 每个节点都有一个后继节点
  - C. 删除一个节点, 需要修改两个指针
  - D. 查找任一节点的算法时间复杂度为 O(n)



$$m = 2$$
;  $lst = [7, 5, 4, 3, 1, 6, 3]$ 

for i in range(len(lst) - 1):

for j in range(len(lst) - 1, i, -1):

if lst[j] < lst[j-1]:

lst[j], lst[j - 1] = lst[j - 1], lst[j]

if  $i \ge m$  and lst[i] != lst[i - 1]:

break

执行该程序段,加框处语句被执行的次数是

A.2

B.3

C.4

D.6

11.有如下 Python 程序段:

def fac(n):

ans = 1

for i in range(2, n + 1):

ans \*= i

return ans



```
def Com(n, m):
    return fac(n) // fac(n - m) // fac(m)
print(Com(5,3))
执行该程序段后,下列说法不正确的是
```

- A. Com()函数运用了递归思想
- В. fac()函数一共被调用了 3 次

C. 输出结果是 10

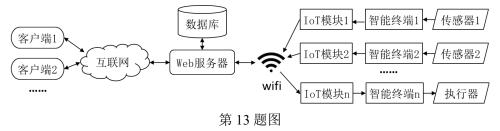
D. 将 Com(5,3)改为 Com(5,2), 运行结果不变

12.列表 a 和列表 b 均有 5 个从小到大排列的整数元素,且列表 a 的最后一个元素大于列表 b 的最后 一个元素。有如下 Python 程序段:

```
i = 0; j = len(a) - 1; c = 0
for key in b:
     while i \le j:
         m = (i + j) // 2; c += 1
         if key < a[m]:
              j = m - 1
         else:
              i = m + 1
    a = a[:i] + [key] + a[i:]
    i += 1; j = len(a) - 1
执行该程序段后, c 的值至少是
                                                   C.10
A.5
                           B.6
                                                                            D.20
```

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.学校搭建了井盖安全实时监测系统,为井盖安装了智能终端、水平传感器和 IoT 物联网模块。各 个井盖的智能终端每隔 10 秒钟就会推送自身倾斜角度给服务器,管理人员可以在学校或家里通 过浏览器查看监测信息。该系统结构示意图如第 13 题图所示。



请回答下列问题:

- (1) 该系统网络应用软件的实现架构是\_\_\_\_\_(单选,填字母: A.B/S 架构 B.C/S 架构)。
- (2) Web 服务器端程序采用 Flask Web 框架开发, 部分代码如下。

# 导入模块,设置 IP 地址和端口号,代码略

```
app = Flask( name )
@app.route('/')
def hello():
    # 从数据库中查询所有记录到 data 中,代码略
    return render template('view.html', data=data)
@app.route("/input", methods=['GET'])
def insert():
```

sid = request.args.get('id') # 获取传感器编号

svalue = float(request.args.get('angle')) # 获取传感器倾斜角度 # 将传感器编号、倾斜角度和当前时间存入数据库, 代码略 return redirect("/") # 跳转到主页 if name == ' main ': app.run(host=ip, port=port) 该系统网站主页被访问时,执行的视图函数名是 ; 若传感器的编号为 2,倾斜角度值为 16.5,则提交数据到 Web 服务器的 URL 为: http://192.168.1.106:8080/ (单选)。 A. id=2&angle=16.5 B. input?sid=2&svalue=16.5 C. sid=2&svalue=16.5 D. input?id=2&angle=16.5 (3) 学校警务室安装了智能终端和警铃,当有井盖出现连续多次倾斜角度大于阈值,警务室的智 能终端就会接收到打开警铃报警的指令。则该指令最可能是由 (选填:服务器/客户端 /传感器)发出。警铃属于第 13 题图中的 (4) 在系统正常运行一段时间后,发现不能查看其中一个井盖的最新监测信息了。结合第 13 题图, 简要说明系统中可能造成上述问题的原因 (不考虑黑客或病毒的原因)。(注: 回答 2 项, 1 项正确得 1 分) 14.高三年级的学生名单按班级顺序存放在"student.csv"文件中,每班学生从1开始连续编号,如第 14 题图 a 所示。小明收集了某次直播讲座的各班签到文件(以班号为文件名,如第 14 题图 b 所 示是高三1班的签到文件"1.csv"中的内容,按签到时间升序排列)。 🧻 student.csv - 记事本 文件(F) 编辑(E) 格式(O) **1.csv** - 记事本 班级,编号,姓名 |请输入直播开始时间: 08:00:00 文件(F) 编辑(E) 格式(O) 查看(V) 1班, 1, 裴\*烨 1 班 迟到: 4 缺席: 0 date, time, id, name 1班, 2, 徐\*轩 1班, 3, 张\*蝶 2月8日,07:51:36,28,陈\*诺 2月8日,07:51:50,10,郑\* 2月8日,07:52:01,12,张\* 2 班 迟到: 1 缺席: 0 1班, 4, 奚\*静 1班, 5, 车\*丹 3 班 迟到: 2 缺席: 3 4 班 迟到: 4 缺席: 4 2月8日,07:52:06,5,车\*丹 1班-6. 何\*欣 5 班 迟到: 0 缺席: 0 2月8日,07:52:15,14,杨\*如 2月8日,07:52:48,32,范\*轩 8班, 1, 王\*豪 8班, 2, 陈\*翔 8班, 3, 卢\*锋 6 班 迟到: 3 缺席: 4 2月8日,07:53:05,6,何\*欣 2月8日,07:53:11,21,张\*任 7 班 迟到: 2 缺席: 1 8 班 迟到: 0 缺席: 1 8班, 4, 王\*雄 2月8日,07:53:26,31,潘\*耀 8班, 5, 陈\*鹏 2月8日,07:55:08,29,陈\*妮 第14 题图 a 第14题图 c 第 14 题图 b 编写 Python 程序分析签到数据,请回答下列问题: (1)编写自定义函数 convert,功能为:转换时间格式,如将"01:01:00"转换为 3660。函数代码如下, 请在划线处填入合适的代码。

def convert(s):

(2)编写自定义函数 cal,功能为:返回某班签到学生编号列表和迟到人数,多次签到的以最早时间为准。函数代码如下,请在划线处填入合适的代码。

def cal(i, late time): #参数 i 为班号, late time 为直播开始时间

df = pd.read csv(str(i) + '.csv')

arr = []

late = 0

for j in df.index:

if df['id'][j] not in arr:

arr.append(df['id'][j])

if convert(\_\_\_\_\_\_) > convert(late\_time):

late += 1

return arr, late # 依次返回签到学生编号列表和迟到人数

(3) 主程序。统计所有班级准时、迟到和缺席人数,输出结果如第 14 题图 c 所示。程序代码如下。 import pandas as pd

df = pd.read\_csv('student.csv') tot = df.groupby('班级').编号.[max()

late time = input('请输入直播开始时间:')

d=[0]\*3 #存储准时人数、迟到人数和缺席人数

for i in range(1, 9):

arr, late = cal(i, late time)

d[0] +=\_\_\_\_

d[1] += late

d[2] += tot[i - 1] - len(arr)

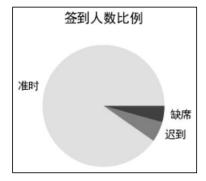
#输出该班级的迟到人数和缺席人数,代码略

(4) 绘制饼图展示全年级的签到人数比例,如第14题图d所示,程序代码如下。

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.pie(d, labels=['准时', '迟到', '缺席'])
plt.title('签到人数比例')
plt.show()

下列说法正确的是 (多选)。

- A. 缺席人数比迟到人数多
- B. 该图能增强数据的解释力和吸引力
- C. 生成该饼图的数据来源于列表 labels
- D. 饼图和环形图都适合展现这类比例关系的数据



第 14 题图 d

15.一批集装箱陆续送达码头,要求叠放到指定的若干位置,每个位置最多可叠放 5 个集装箱,且轻的在上,重的在下。工作人员根据每个送达的集装箱的重量,利用临时位置经过最少的移动次数放到合适的位置,若移动次数相同,优先选择编号小的位置(临时位置不考虑集装箱重量大小关系)。

例如:共有8个集装箱,要求叠放到2个位置。已依次送达6号、2号、3号、5号,叠放情况如第15题图 a 所示。



第15题图 a

新送达的集装箱为7号,重量为28。

- 若叠放到位置 0, 需要移动 3 次:
  - ①2 号→临时位置 ②7 号→位置 0 ③2 号→位置 0
- 若叠放到位置 1, 需要移动 5次:
- ①5 号→临时位置 ②3 号→临时位置 ③7 号→位置 1 ④3 号→位置 1 ⑤5 号→位置 1 叠放到位置 0 的移动次数 < 叠放到位置 1 的移动次数, 所以叠放到位置 0。

编写 Python 程序实现上述功能,请回答下列问题:

(1) 函数 judge(x, st)的功能是:返回 x 号集装箱放入某位置时需移动的次数。列表 st 存储了该位置从下到上已叠放的集装箱编号。请在划线处填入合适的代码。

# goods 存储了已送达集装箱的信息,如: {6:29,2:18,3:20,5:19}。

def judge(x, st):

(2) 当某个集装箱送达时,输入其编号和重量,将该集装箱放到移动次数最少的位置,并输出移动过程。程序执行结果的部分截图如第 15 题图 b 所示。请在划线处填入合适的代码。

```
goods = \{\}
n = 8
k = (n - 1) // 5 + 1
p = [[] \text{ for i in range}(k)]
for i in range(n):
    # 输出当前集装箱叠放情况,代码略
    x = int(input('请输入到达的集装箱编号:'))
    weight = int(input('请输入到达的集装箱重量:'))
    goods[x] = weight
    min steps = 999
    for j in range(k):
        t = judge(x, p[j])
        if ____
            minp = j
            min steps = t
    print('移动过程为:')
    tmp = []
    for j in range( ② ):
        print(p[minp][-1], '-> 临时位置')
```

```
已送达集装箱信息:
{6: 29, 2: 18, 3: 20, 5: 19}
当前集装箱叠放情况:
位置0: [6, 2]
位置1: [3, 5]
请输入到达的集装箱编号: 7
请输入到达的集装箱重量: 28
移动过程为:
2→ 临时位置
7→ 位置0
2→ 位置0
叠放结果:
位置0: [6, 7, 2]
位置1: [3, 5]
```

第 15 题图 b

```
tmp.append(p[minp].pop()) # 删除 p[minp]的最后一个元素,并追加到 tmp p[minp].append(x) print(x, '->', '位置' + str(minp)) while _______: print(tmp[-1], '->', '位置' + str(minp)) p[minp].append(tmp.pop()) # 输出叠放结果,代码略
```

(3) 若送达的集装箱编号和重量依次为 4:20, 2:18, 1:23, 3:20, 7:28, 6:29, 0:17, 5:19, 根据要求叠放到位置 0 和位置 1,则位置 1 从下往上叠放的集装箱编号依次为\_\_\_\_。