绝密★考试结束前

技术（一）

本试卷分两部分，第一部分信息技术，第二部分通用技术。满分 100 分，考试时间 90 分钟。

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、多选、错选均不得分）

阅读下列材料，回答第 1 至 3 题。

某社区健康驿站配备智能体检一体机，集成身高、体重、血压、X 光片等 20 余项检测功能，支持人脸识别登录、数据云端同步及健康报告生成。设备每日采集约 200 名居民的数据，与社区卫生服务中心平台实时对接。驿站同时引入中医体质辨识大模型，基于 1100 万条中医知识图谱数据，为居民提供个性化养生建议。

1. 下列关于数据与信息的说法，正确的是 B

A.数字是该系统唯一的数据表现形式

B.人脸识别信息属于个人敏感信息

C.该系统处理的都是结构化数据

D.提供的养生建议对不同人具有相同价值

A. 数据表现形式不限于数字

C .X 光片等是非结构化数据

D.养生建议对不同人价值不同。

2. 关于该系统中各项功能，其中运用了人工智能技术的是 C

A.身高体重等检测功能　　B.数据云端同步

C.中医体质辨识大模型　　D.打印健康报告

3. 每日采集约 200 名居民的数据，现要用二进制对每位居民进行编码，则所需的二进制位数最少是 D

A.5　　B.6　　C.7　　D.8

$2^7=128 < 200 < 256=2^8$，因此需要8 位二进制编码

阅读下列材料，回答第 4 至 6 题。

某城市交通管理中心通过路侧传感器和车载设备实时采集道路车流量、车速等数据，上传至云端服务器，并保存到数据库中。服务器中的 AI 模块对数据进行分析后，将结果呈现在指挥中心大屏，并推送至交警和公交司机的移动终端 APP。一旦检测到异常拥堵或事故，AI 持续提供绕行路线建议，交警可结合建议指挥现场交通疏导。

4. 下列关于该信息系统组成与功能的说法，正确的是 D

A.该系统中的用户指的是交警和公交司机

B.系统中数据收集和传输功能都由车载设备完成

C.系统的数据输出功能都由指挥中心大屏实现

D.提供绕行路线建议体现了数据加工处理功能

B.数据收集由路侧传感器和车载设备共同完成，传输功能也依赖两者将数据上传至云端服务器，并非仅由车载设备承担。

C.数据输出功能既通过指挥中心大屏呈现结果，也会推送至交警和公交司机的移动终端 APP。

5. 下列关于该信息系统中硬件和软件的说法，不正确的是 A

A.该系统硬件包括传感器、车载设备、数据库等

B.服务器中一定包含运算器和控制器

C.可通过优化 AI 模块算法来提升数据分析能力

D.移动终端 APP 属于应用软件

数据库是软件不是硬件

6.下列有关网络系统的说法，正确的是 C 范围太广，局域网不够用

A.该系统可以在局域网中实现

B.系统硬件搭建完成后无需网络联通测试

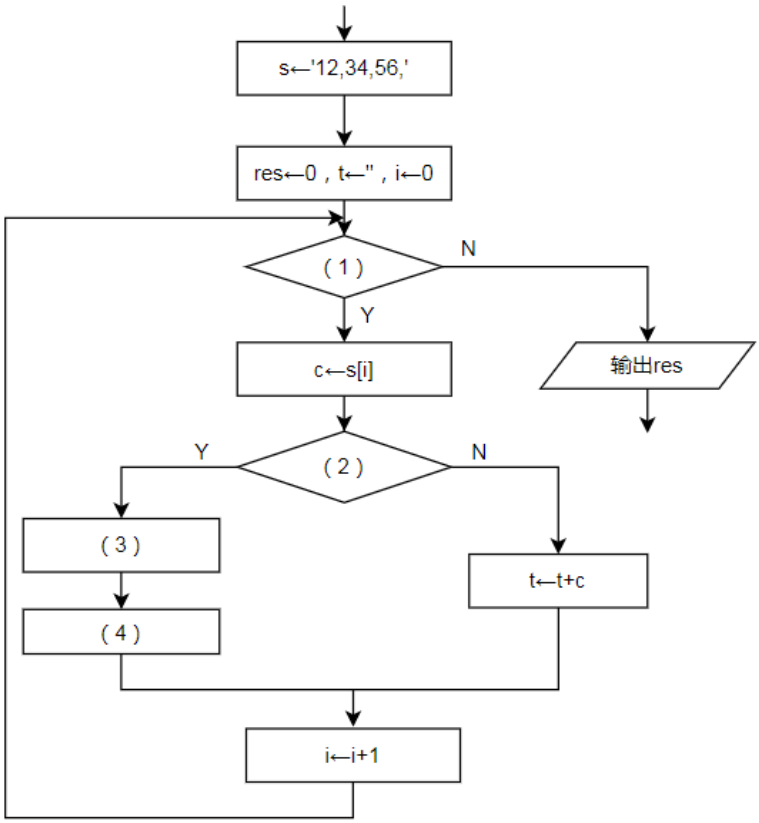
C.数据上传云端服务器可能用到 5G 通信

D.移动终端接入网络可以没有 IP 地址

7. 输入多个以逗号分隔的数字字符串，如“12,34,56,”，设计算法取出数字串并求和，结果是12+34+56=102，部分流程图如第7题图所示，表达式有 A

①i≤len(s)？ ②c=',’？ ③i≤len(s)-1？ ④t←’’ ⑤res←res+int(t)

⑥c≠',’？ ⑦t←0



第 7 题图

以 d 开头的出栈序列有：
d,e,c,b,a;
d,c,e,b,a;
d,c,b,e,a;
d,c,b,a,e，共4种,其实就是

则（1）～（4）处表达式序号依次为

A. ③②⑤④ B. ①⑥⑤④

C. ①②⑤④ D. ③⑥⑤⑦

8. 元素 a、b、c、d、e 按序入栈，在所有的出栈序列中，以元素 d 开头的出栈序列个数为 C

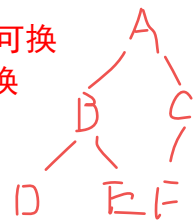
A.2 B.3 C.4 D.5

9. 某完全二叉树包含节点 A、B、C、D、E、F，其中 A 和 B 节点的度为 2，则该完全二叉树的中序遍历结果不可能是 D

A.DBEAFC B.CADBEF

C.CBDAFE D.EFABCD

DEFC位置可换
AB位置可换



10. 数组元素 a[0]～a[5]为随机正整数，执行如下 Python 程序段后，a[0]～a[5]的值可能是 A

```
for i in range(5):
    for j in range(5,i,-1):
        x=a[j]%2
        y=a[j-1]%2
        if x<y or x==y and a[j]<a[j-1]:
            a[j],a[j-1]=a[j-1],a[j]
```

后一个数

前一个数

x<y,表示后偶前奇，需要交换

x==y and a[j]<a[j-1],表示同奇同偶时且降序时交换。

结果：前偶后奇，各自升序

A.8, 1, 3, 5, 7, 9　　B.1, 2, 4, 6, 6, 8　　C.6, 4, 4, 2, 2, 5　　D.1, 7, 9, 2, 6, 8

11. 有如下 Python 程序段：

```
def remove(a, k, i):
    if i >= len(a):
        return a[:k + 1]
    if a[i] != a[k]:
        k += 1
        a[k] = a[i]
    return remove(a, k, i + 1)
a=["A","P","P","P","A","A","A","C","C"]
result = remove(a,0,1)
print(result)
```

程序功能是去除连续重复元素，保留一个，结果为["A","P","A","C"]。

程序运行后输出结果是 C

A.["A", "P", "C"]　　B.["A", "P"]

C.["A", "P", "A", "C"]　　D.["P", "C"]

12. 给定一个 n×n 的矩阵（以 n=3 为例），其中每行和每列都是升序排列的，矩阵元素用列表 matrix 存储。编写如下 Python 程序，找出矩阵中的第 k 小元素，运行结果如第 12 题图所示。

```
matrix=[[1,5,9],[10,11,13],[12,13,15]]
k=int(input("请输入数字(1~9)："))
left = matrix[0][0]
right = matrix[-1][-1]
while left <= right:
    mid = (left + right) // 2
    cnt = 0
    i, j = n - 1, 0
    while i >= 0 and j < n:
        if matrix[i][j] <= mid:
            cnt += i + 1
            j+=1 ①
        else:
            i-=1 ②
    if cnt < k:
        left = mid + 1
    else:
        right = mid - 1
print("第",k,"小元素是：",  ③  )
```

[[1, 5,9],
[10,11,13],
[12,13,15]]

请输入数字(1~9)：5
第 5 小元素是： 11

第 12 题图

第一次

统计<= mid 的元素个数时，利用矩阵的行列有序性，从左下角开始用扫描：

对于①处：当 matrix[i][j] <= mid 时，说明j列中从0行到i行的元素都小于等于mid，cnt统计比mid小的个数就是i+1个。已经统计完当前列的符合条件元素，需要向右移动一列，即j+=1。

对于②处：当matrix[i][j]>mid时，说明当i行第j列及其右边的元素都大于 mid，取小的需要向上移动一行，即i-=1。

对于③处：通过不断调整 left 和 right 的范围，二分查找结束时 left指向首个满足cnt>=k 的值，即第k小元素，所以此处应填left。

相等时往前找

right | left

```
            right = mid - 1
    print("第",k,"小元素是：",_____③_____)
```

划线①②③处应填入的代码为：B

A.①j += 1 ②i -= 1 ③right

B.①j += 1 ②i -= 1 ③left

C.①j -= 1 ②i += 1 ③right

D.①j -= 1 ②i += 1 ③left

二、非选择题（本大题共 3 小题，其中第 13 小题 7 分，第 14 小题 10 分，第 15 小题 9 分，共 26 分）

13. 设计一个智能鱼缸氧气监控系统，实时监测两个鱼缸的含氧量（单位：mg/L）。当某个鱼缸的含氧量低于 5mg/L 且氧气泵关闭状态持续 10 分钟时，自动开启该鱼缸的氧气泵；当含氧量高于 8mg/L 且氧气泵开启状态持续 5 分钟时，自动关闭氧气泵。系统每分钟检测一次含氧量，含氧量每分钟随机波动±0.5mg/L。初始时两个鱼缸的含氧量为 4.0-9.0mg/L 之间的随机值，氧气泵初始状态均为关闭，每分钟输出一次各鱼缸的当前含氧量和氧气泵状态（开启/关闭）。

请回答下列问题：当含氧量在5~8mg/L时，氧气泵状态清0

（1）已知鱼缸监控系统前续状态如下：

鱼缸 1：含氧量 5.2 mg/L，氧气泵关闭（持续关闭时间：15 分钟）

鱼缸 2：含氧量 4.8 mg/L，氧气泵关闭（持续关闭时间：8 分钟）

系统每分钟检测一次，后续检测到的含氧量变化为：

第 1 分钟：鱼缸 1 含氧量：4.9，鱼缸 2 含氧量：4.6

第 2 分钟：鱼缸 1 含氧量：5.1，鱼缸 2 含氧量：4.4

问：在第 2 分钟检测完成后，两个鱼缸的氧气泵状态分别是？B（单选，填字母）

A.鱼缸 1：开启，鱼缸 2：关闭　　B.鱼缸 1：关闭，鱼缸 2：开启

C.鱼缸 1：关闭，鱼缸 2：关闭　　D.鱼缸 1：开启，鱼缸 2：开启

（2）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。

```
import time
import random
#两个鱼缸的初始含氧量，存在 oxy 列表中，代码略
pump = [0] * 2              #氧气泵状态 (0 关/1 开)
timer = [0] * 2             #计时器，以分为单位
while True:
    #显示当前时间，代码略
    for i in range(2):
        #模拟鱼缸的含氧量变化，随机波动±0.5mg/L，代码略
        if oxy[i] < 5:
            timer[i]①+= 1
            if timer[i] >= 10 and pump[i]==0:
                pump[i] = 1
                print(f"鱼缸{i+1} 含氧量低！启动氧气泵")
        elif oxy[i]>8②and pump[i]==1      或 oxy[i] > 8 and pump[i]:
            timer[i] += 1
```

```
            if timer[i] >= 5:
                pump[i] = 0
                print(f"鱼缸{i+1} 含氧量正常，关闭氧气泵")
        else:
            timer[i]=0                    =                    0
                ③
        #显示状态
        if pump[i]:
            status = "开"
        else:
            status = "关"
        print(f"鱼缸{i+1}: 含氧量 {oxy[i]:.1f} mg/L | 氧气泵: {status}")
    time.sleep(60)          #每隔 1 分钟检测一次
```

14. 某中学部署智慧教室环境监测系统，每个教室配备传感器实时采集教室内的湿温度、空气质量（PM2.5 浓度）数据，智能终端每隔 5 分钟通过 IoT 模块将传感器数据上传并保存至服务器数据库。若数据出现异常时，服务器将通过智能终端自动触发教室新风设备。管理员可通过浏览器登录系统查看实时数据与历史趋势。请回答下列问题：

（1）在搭建该系统时，下列硬件不经过其他硬件设备直接相连的是____C____▲（单选，填字母：A.传感器和服务器/B.智能终端与服务器/C.执行器与智能终端）。

（2）该系统的开发模式为 B/S 架构，并基于 Flask Web 编写该系统服务器程序。

①相比 C/S 架构，B/S 架构的优势为____B____▲。（单选，填字母）

A.对服务器要求较低　　B.客户端无须专门软件　　C.系统的通信开销较低

②智能终端通过 URL 向服务器传递数据时，需要知道____BD____▲。（多选，填字母）

A. 传感器连接智能终端的引脚　　B. 服务器的 IP 地址和端口

C. 数据库类型及文件名　　D. 服务器端对应页面的路由

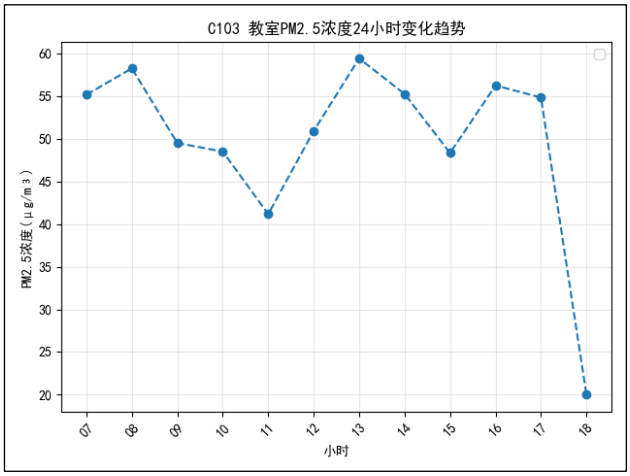
（3）现在要开通班主任和任课老师访问系统查看数据的功能，从安全角度出发，请提出两点建议。

①身份认证：认证方式要安全，强度高，如账号密码不共用，且密码强度要高；刷脸登录等；　②访问控制：设置不同的权限；③局域网访问，不对外开放

（4）将系统中某天的数据导出到文件 data.csv，部分数据如第 14 题图 a 所示。现要根据输入教室编号，绘制该教室各时间段 PM2.5 浓度平均值的线形图（如第 14 题图 b 所示），请在划线处填入合适的代码。

时间段	教室编号	PM2.5浓度	温度	湿度
07:00-07:05	C101	34	24.4	39
07:00-07:05	C102	58	22.6	56
07:00-07:05	C103	59	22.7	42
07:00-07:05	C104	75	23.1	58
07:00-07:05	C105	54	21.2	50
……	……	……	……	……
18:00-18:05	C101	57	24.3	40
18:00-18:05	C102	97	20.7	63
18:00-18:05	C103	20	24.9	64
18:00-18:05	C104	67	24.6	48
18:00-18:05	C105	20	24.6	45

第 14 题图 a



第 14 题图 b

	小时	PM2.5浓度
6	13	59.416667
1	08	58.250000
9	16	56.250000
7	14	55.250000
0	07	55.166667

第 14 题图 c

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('data.csv', encoding='gbk')
class_id = input("请输入教室编号(如 C101): ")
df1 = df[df['教室编号'] == class_id 或 df.教室编号 == class_id]
df1['小时'] = df1['时间段'].str[0:2]          #取出时间段中的'小时'部分
df1 = df1.groupby('小时',as_index=False)["PM2.5 浓度"].mean()
plt.plot(df1["小时"],df1["PM2.5 浓度"])
#设置绘图参数，并显示线形图，代码略
```

（5）为了找出该教室 PM2.5 浓度平均值最高的 5 个时间段（如第 14 题图 c 所示），方框中应填入的语句为 AC▲（多选，填字母）。

```
df2 = 
print(df2)
```

A. df1.sort_values('PM2.5 浓度',ascending=False).head()

B. df1.sort_values('PM2.5 浓度',ascending=True).tail()

C. df1.sort_values('PM2.5 浓度',ascending=False)[0:5]

D. df1.sort_values('PM2.5 浓度',ascending=True)[0:5]

（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）

15. 模拟 CPU 进程调度。假设某系统只有一个 CPU 处理进程任务，每一个进程的到达时间、执行时间和运行优先级都是已知的。其中运行优先级用自然数表示，数字越大，则运行优先级越高。进程调度规则如下：

①如果一个进程到达时 CPU 是空闲的，则它会一直占用 CPU 直到该进程结束。除非在这个过程中，有一个比它运行优先级高的进程到达，这个新的运行优先级更高的进程会占用 CPU，而老的只有等待（后面再次运行时只需运行剩余时间）。

②如果一个进程到达时，CPU 正在处理一个比它运行优先级高或运行优先级相同的进程，则这个新到达的进程必须等待。

③一旦 CPU 空闲，如果此时有进程在等待，则选择运行优先级最高的先运行；如果有多个运行优先级相同的进程，则选择到达时间最早的。

已知不同进程有不同的编号，不会有两个相同运行优先级的进程同时到达，且各个进程数据已

经按到达时间从小到大排序，如第 15 题图 a 所示。按照进程结束的时间输出每个进程的进程号和结束时间，如第 15 题图 b 所示。一个进程信息将用一个列表存储，如第 15 题图 a 的 1 号进程存储为[1,1,5,3]。

进程号	到达时间	执行时间	运行优先级
1	1	5	3
2	10	5	1
3	12	7	2
4	20	2	3
5	21	9	4
6	22	2	4
7	23	5	2
8	24	2	4

第 15 题图 a

进程号	结束时间
1	6
3	19
5	30
6	32
8	34
4	35
7	40
2	42

第 15 题图 b

请回答下列问题：

（1）若有 4 个进程的信息如第 15 题表 1 所示：

第 15 题表 1

进程号	到达时间	执行时间	运行优先级
1	1	5	2
2	10	10	1
3	15	5	3
4	16	2	1

结束6
10开始，运行到15，进程3插队，进程3在20结束，进程2继续运行5，25结束

则进程 2 处理的结束时间是____▲ 25____。

（2）函数 quepush 的功能是增加一个进程信息列表 task(依次包括进程号、到达时间、执行时间和运行优先级)到链表队列 que 中，同时让队列中的进程数据保持有序，即运行优先级高的进程排在前面，运行优先级相同则到达时间早的排在前面。请在划线处填入合适的代码。

```
def quepush(que,head,task):
    task.append(-1)     #在列表 task 末尾添加一个元素-1，用于存储链表指针
    que.append(task)    #将某个进程 task 加入到队列 que 中
    i = len(que)-1
    if head == -1:
        head = i
    else:
        p = head
        q = p
        while p!=-1 and (task[3] < que[p][3] or ____①____):
            q=p
            p=que[p][4]
        if p == head:
            que[i][4] = head
            ____②____
        else:
            que[i][4] = p
            que[q][4] = i
    return que,head
```

task[3]==que[p][3] and task[1]>que[p][1]

② head=i 或head=len(que)-1 插头

（3）显示各个进程完成时间的部分 Python 程序如下，请在划线处填入合适的代码。

```
def readfile(filename):
    '''
```

读取进程数据,每项数据包括进程号、到达时间、执行时间和运行优先级，返回所有进程数据列表。代码略。若共读取 2 个进程信息如第 15 题表 2 所示：

第 15 题表 2

进程号	到达时间	执行时间	运行优先级
1	1	5	3
2	10	5	1

```
    则返回 tasks 的值为[[1,1,5,3],[2,10,5,1]]。
    '''
    return tasks
#quepop 函数的功能是从队列 que 中获取队首进程。head 是链表队列 que 的头指针。
def quepop(que,head):
    if head==-1:
        return [],-1,[]
    task = que[head][0:4]
    head = que[head][4]
    return que,head,task
#主程序，计算并显示各个进程的完成时间。
tasks = readfile("task.txt")                    #读取进程序列
que,head=[],-1
now=0                                           #当前时间
for i in range(len(tasks)):
    while head!=-1:
        que,head,task = quepop(que,head)
        if now+task[2]<=tasks[i][1]:    进程不冲突
            now+=task[2]
            print(task[0],now)                  #显示进程号、完成时间
        else:    当前进程未完成，下一个进程进来
            task[2]=_____▲ task[2]-(tasks[i][1]-now)_____
            que,head = quepush(que,head,task)
            break                               #退出循环
    que,head = quepush(que,head,tasks[i])
    now=tasks[i][1]
while head!=-1:
  que,head,task = quepop(que,head)
  now += task[2]
    print(task[0],now)                          #显示进程号、完成时间
```

（4）若使用第 15 题图 a 中的进程数据测试该程序，队列 que 中最多时包含_____▲ 6 _____个进程。