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

技术（三）

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6题：

智能CT影像系统是基于计算机断层扫描（CT）技术，结合人工智能、大数据和网络通信的医疗诊断系统。该系统能够高效采集、存储和分析患者的CT影像数据，并利用AI算法辅助医生进行病灶检测（如肿瘤、炎症等），提高诊断准确性和效率。

1. 关于智能CT影像系统中数据的说法，正确的是 A

A. CT影像数据能用于诊断疾病，体现了数据的价值性✓

B. CT影像数据分析处理后不再以二进制形式存储

C. CT影像数据是结构化数据 图片、音频、视频属于非结构化数据

D. CT影像数据仅存在于医院服务器中

2. 关于信息安全与信息社会责任，下列行为合适的是 C

A. 同一科室医生共用一个系统账户

B. 未经同意即公布病人的检测情况

C. 定期备份检测数据

D. 随意剪辑病人CT影像

3. 下列关于该信息系统组成的说法，不正确的是 D

A. CT扫描仪属于输入设备

B. 病人和医生都属于信息系统用户

C. 智能CT影像系统软件是应用软件

D. 服务器的存储器容量不会影响系统性能

4. 该系统中人工智能是基于神经网络方法实现的，下列说法正确的是 C

A. 当AI的诊断结果和医生的诊断结果冲突时，应以AI的诊断结果为准

B. 该人工智能需要预先手工设置所有疾病的知识库

C. 提升服务器性能、优化AI算法、提升CT拍摄质量都有助于提高人工智能诊断准确率

D. 该人工智能进行病症诊断时，需要原始的训练数据进行比对

建立知识库属于符号主义，AI诊疗（识别影像数据）一般为联结主义

5. 下列技术中，最可能用于连接 CT 扫描仪和医院服务器的是 B

A. 3G

B. 网线

C. 蓝牙 距离10米内

D. RFID

提升服务器性能　目前有争议

1.认为不能提升准确率　（慢慢还是能算出结果？）

2.认为可以提升准确率（一些大模型，内存越大，可以运行的大模型就更精确）

6. 将CT影像数据存储为未经压缩的BMP格式文件，下列说法不正确的是 B

A. 影像数据采集需经过采样、量化和编码过程 ✓

B. 病人病情越复杂，得到的单张图像文件的存储容量越大 ✗ BMP图像：存储容量与图片内容无关

C. 将BMP格式转换成JPG格式有助于节省存储空间

D. CT影像共256色，则BMP文件量化位数为8位

7. 某栈初始有 3 个元素，经过一系列入栈、出栈操作后，栈变为空，若元素出栈顺序为 1,2,3,2，则栈初始从栈底到栈顶的元素值不可能是 B

栈外可能也有元素

2
1
3 ✗

A. 3,2,1

B. 3,1,2 ✗

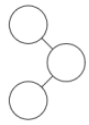
C. 2,2,1

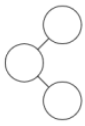
D. 2,3,2

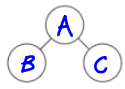
8. 某二叉树前序遍历结果为 ABC，后序遍历结果为 CBA，其树形结构图不可能是 D

前序、中序 可确定二叉树
中序、后续 可确定二叉树
前序、后续 不能确定二叉树

A. 

B. 

C. 

D. 

9. 有如下 python 程序段：

```
dic={}
s="a12b4c563"
t=0
for i in s:
    if "0"<=i<="9":
        t=t*10+int(i)
    else:
        dic[i]=t
        t=0
```

则程序运行后，dic["b"]的值为 B　　遍历字符串，遇到字母时记录前面数字

A. 2

B. 12

C. 4

D. 563

10. 下列 python 程序段用于输出 a,b,c 三个变量中第 2 大的值C

```
if a>b:
    if b>c:
        print(b)
    elif a>c:
        print(c)
    else:
        print(a)
else:
    if c>b:
        print(b)
    elif a>c:
        print(c)
    else:
        print(a)
```

a<=b

1.逻辑检查

a<=b c<=b

X

a<=c a<=b c<=b

2.直接代入法检查

X

程序存在漏洞，下列数据能测试出问题的是

A. a=1 ; b=2 ; c=3

B. a=1 ; b=2 ; c=1

C. a=1 ; b=3 ; c=2

D. a=3 ; b=1 ; c=2

11. 有如下 python 程序段：A

st ed

4 3 1 7 6

```
def sorts(a,st,ed):
    if st==ed:
        return
    for i in range(ed,st,-1):
        if a[i]>a[i-1]:
            a[i],a[i-1]=a[i-1],a[i]
    sorts(a,st+1,ed)
sorts(a,0,4)
```

指定区间内的冒泡排序

若列表 a 初值为[4,3,1,7,6]，则运行程序后，a 的值变为

A. [7,6,4,3,1]

B. [1,3,4,6,7]

C. [1,3,4,7,6]

D. [4,7,6,3,1]

将数组分割为最多 2 个子数组，找到所有分割方式中“最大子数组和的最小值”

12. 有如下 python 程序：

```
def can(x):
    s=0;t=1
    for num in nums:
        s+=num
        if s>x:
            t+=1
            s=num
    if t>2:
        return False
    else:
        return True
left=max(nums)
right=sum(nums)
while left<=right:
    mid=(left+right)//2
    if can(mid):
        right=mid-1
    else:
        left=mid+1
print(left)
```

left　　mid　　right

max(nums)　　21　　sum(nums)

10　　32

判断是否能将数组分割为不超过 2 个子数组，且每个子数组的和都不超过 x

对分方式不断尝试x的值

x的值能否把列表分为2段，

这里改3、4...就是求分为3段、4段时最小的x

x的值小了

（left=mid+1）

段数多了，x值增加

段数不够，x值减少

x的值大了

（right=mid-1）

max(nums)=10，最大为sum(nums)=32。

二分查找过程：mid=21 → 可分割为[7,2,5]和[10,8]（和=14和18），can成立,r=20

mid=15（l=10,r=20） → 可分割为[7,2,5]和[10,8]（和=14和18），can不成立,l=16

mid=18（l=16,r=20） → 可分割为[7,2,5]和[10,8]（和=14和18），can成立。r=17

mid=17（l=16,r=17） → 可分割为[7,2,5]和[10,8]（和=14和18），can不成立。l=18

最终输出left=18。

若数组 nums=[7, 2, 5, 10, 8]，则程序运行后输出值为 C

A. 8　　B. 15

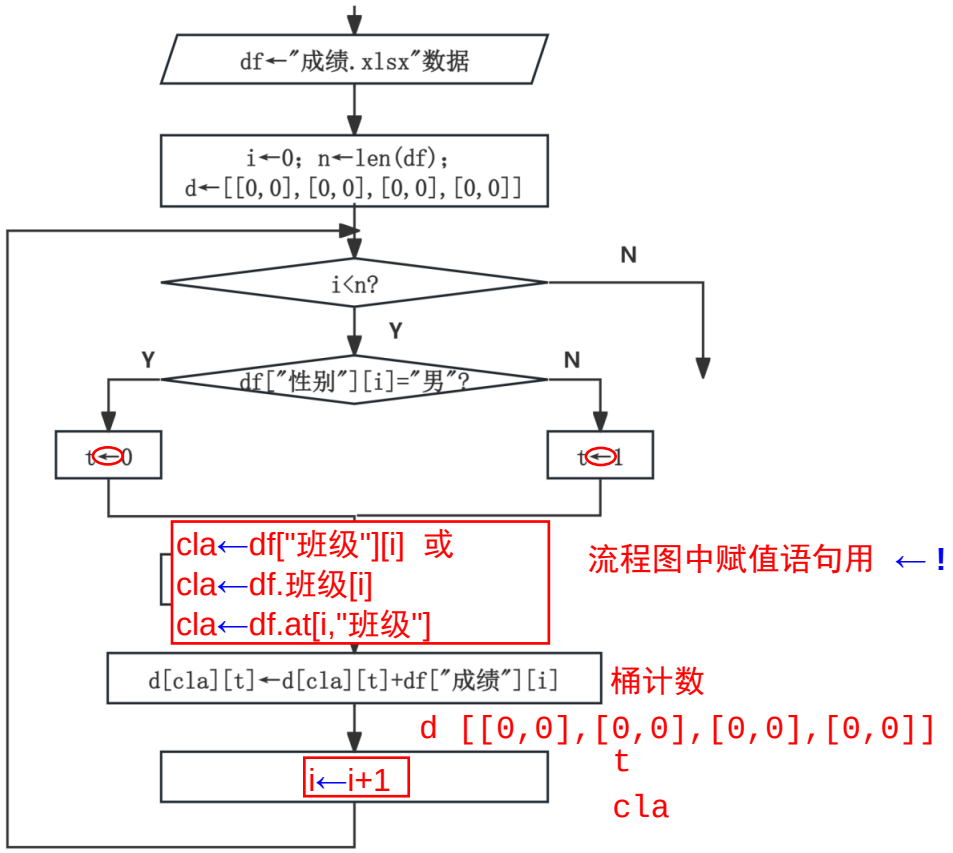
C. 18　　D. 21

二、非选择题（本大题共 3 小题，其中第 13 小题 6 分，第 14 小题 10 分，第 15 小题 10 分，共 26 分）

13. 某校 4 个班（编号 0~3）的趣味比赛结果存储在“成绩.xlsx”文件中，部分数据如第 13 题图 a 所示。现要分别计算各班男女生的总成绩，部分算法如第 13 题图 b 所示。

班级	性别	成绩
0	男	55
0	女	18
2	女	32
3	男	13
1	男	52
2	女	15
3	男	26
2	女	31
3	男	33
0	女	26

第 13 题图 a



第 13 题图 b

（1）请在第 13 题图 b 中①②处填入合适内容。

（2）下列代码也能实现本功能的是 ▲B （单选，填字母）。

A. df1=df.groupby("班级")
df2=df1.groupby("性别").sum()

B. df["组"]=df["班级"].astype(str)+df["性别"]
df1=df.groupby("组").sum()
#注：astype(str)将数字列转换为字符串列

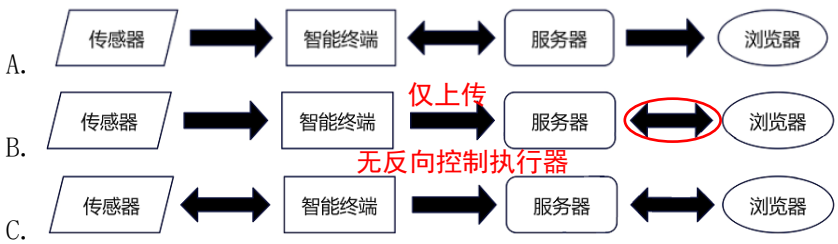
班级	性别	成绩
0	男	55
0	女	18
2	女	32
3	男	13
1	男	52
2	女	15
3	男	26
2	女	31
3	男	33
0	女	26

→

班级	性别	成绩	组
0	男	55	0男
0	女	18	0女
2	女	32	2女
3	男	13	3男
1	男	52	1男
2	女	15	2女
3	男	26	3男
2	女	31	2女
3	男	33	3男
0	女	26	0女

14. 某高中搭建教室温度监控系统，该系统能监测该校 n 个教室（分别编号 0~n-1）的温度。每个教室装有温度传感器和智能终端，传感器采集的数据由智能终端经 IoT 模块发送到 Web 服务器，用户通过浏览器可快速查询某编号教室的平均温度。请回答下列问题：

（1）本系统中传感器、智能终端、服务器与浏览器之间数据的传输关系，其中合理的是 ▲B （单选，填字母）。



（2）某个智能终端的部分代码如下：

```
class=102
while True:
    temp=pin0.read_analog()
    Obloq.get("con?id="+str(class)+"&temp="+str(temp),10000)
    sleep(1000)
```

超时时间

con?id=102&temp=30

若获取温度为 30，则其提交数据到 Web 服务器的 URL 为 http://192.10.152.197/ ▲ 。

（3）服务器上的程序具有功能 a：获取并存储各个智能终端传递来的教室编号和温度；功能 b：获取浏览器传递来的教室编号后快速查询该教室的平均温度。请在程序中划线处填入合适的代码。

功能 a 部分代码如下：

队链结构

```
head=[-1]*n ; data=[] ; i=0
while True:
    #获取智能终端传递的编号和温度，保存在 id 和 temp 中,代码略
    data.append([id,temp,head[id]])
    head[id]=i 或 head[id]=len(data)-1
    i+=1
```

新数据不断插在链表头节点之前
最终：链尾为最早数据，链头为最晚数据

若最早数据为链表头结点（如果单链）

```
head=0
while True:
    data.append[x,x,len(data)+1]
data[-1][2]=-1
```

必须加上

功能 b 部分代码如下：

```
#获取浏览器传递来的查询编号，保存在 id_find 中，代码略
p=head[id_find]
m=s=ans=0
while p!= -1:
    s+=data[p][1]
    m+=1
    p=data[p][2]
if m!=0:
    ans=s/m
```

（4）运行一段时间后，管理员发现服务器仅无法获取编号 4 教室的温度数据，系统中造成上述问题的原因可能是 ▲ （经检测，各器件之间的连接均正常）

倾向写硬件方面原因
1.编号4教室的智能终端故障
2.编号4教室的Iot故障
3.编号4教室的智能终端和Iot通讯故障
（注意：未特别指出是编号4教室的答案不给分）
软件方面：
服务器获取4号教室的温度数据代码有误
xxx代码有误 不详细 ✗

15. 扫雷游戏规则如下：一个 n*n 大小的棋盘中，共有 m 个方格上为雷。游戏开始时，所有方格为未揭示状态，选择方格来揭示它的内容，若为雷则游戏失败；若为数字，则表示周围一圈 8 个方格中（边界上方格的周围格子数会小于 8）雷数量，例如第 15 题图 a 所示。如果被揭示的方格为数字 0，那么该方格会展开，继续揭示周围一圈方格。玩家揭示雷则游戏失败，揭示所有非雷方格才游戏胜利。

（1）某棋盘如第 15 题图 b 所示，已有 4 个格子被揭开，可推测 A 点内容为__▲A__（单选，填字母：A.雷 / B.0 / C.1 / D.2）。

雷	雷	1
2	3	2
0	1	雷

第 15 题图 a

1	2	
	3	
1		

第 15 题图 b

（2）编写扫雷游戏程序，定义 outner 数组用于存储游戏的显示界面，仅包含“#”和数字，“#”表示方格未揭示，数字表示已揭示方格的周围雷点数。打印显示界面并判断游戏是否胜利的 pri 函数如下：

```
n=m=10
outner=[["#" for i in range(n)]for j in range(n)]
def pri():
    cnt1=cnt2=0
    for i in range(n):
        for j in range(n):
            print(outner[i][j],end=" ")
            if "0"<=outner[i][j]<="8":
                cnt1+=1
            if outner[i][j]=="#":
                cnt2+=1
        print()
    if [          ]:
        print("恭喜，你胜利了")
        return True
    else:
        return False
```

游戏胜利条件为揭示所有非雷方格

加框处代码用于判断游戏是否胜利，下列语句正确的是__▲AC__（多选，填字母）。

A. cnt1+m==n*n　　B. cnt1+cnt2==n*n　恒成立

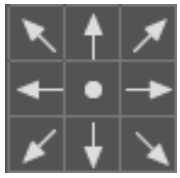
C. cnt2==m　　D. cnt2<m

（3）实现上述功能的部分 Python 程序如下，其中 inner 数组用于存储各个方格的周围雷点数，若值为-1 表示为雷，请在划线处填入合适的代码。

```
inner=[0 for i in range(n)]for j in range(n)]
def generate():      #生成初始棋盘
    #随机生成 m 个雷点位置，统计并生成 inner 数组的值，代码略
def judge(x,y):      #根据选中的位置，判断结果
    if inner[x][y]==-1:
        print("抱歉，你失败了")
        return False
    elif inner[x][y]!=0:
        outner[x][y]=str(inner[x][y])
    else:
        que=[[-1,-1]for i in range(n*n)]
        que[0]=[x,y]
        head=0
        tail=1
        while head<tail:
            cur=que[head]
            outner[cur[0]][cur[1]]=str(inner[cur[0]][cur[1]])
            w=[[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]]
            for step in w:
                next=[cur[0]+step[0],cur[1]+step[1]]
                if 0<=next[0]<n and 0<=next[1]<n and next not in que:
                    x=inner[next[0]][next[1]]
                    if x==0:
                        que[tail]=next
                        tail+=1
                    outner[next[0]][next[1]]=str(x)
            head+=1
    return True
generate()
while not pri():
    x=int(input("请输入点击节点的横坐标(0~n-1):"))
    y=int(input("请输入点击节点的纵坐标(0~n-1):"))
    if not judge(x,y):
        break
```

整数类型

若当前值大于0，表示周围一圈有雷，应在outner上显示数字值



当前方格向周围8个方格移动时x，y变化情况

不能越界

防止重复访问：已经访问过，则已经在que列表数据

利用队列，广度优先搜索（广搜）

		0	
0		0	
0	0	0	
0		0	