

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

高三上信息周练8

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 2 题:

某市举办人工智能展, 展会汇聚全球顶尖的人工智能企业和专家, 展示最新的技术成果和应用案例, 搭建一个产学研合作平台。市民、团体、媒体线上注册后方可参加线下活动, 同步开展直播及全媒体传播。

- 下列关于该活动中数据和信息的说法, 正确的是 **C**
 - 同一活动内容的数据只有经过数字化才能被加工 **计算机中**
 - 活动内容中图像和文本是**数据**的表现形式
 - 活动内容通过多种媒介传播有助于信息的共享
 - 同一活动信息对不同的人的价值是相同的
- 关于信息安全与信息社会责任, 下列行为合适的是 **B**
 - 团体负责人将注册的参会人员名单公开在社交媒体分享
 - 媒体在直播时对涉及未公开专利的技术展示画面进行模糊处理
 - 市民在展会现场随意连接未加密的公共 WiFi 传输个人敏感信息
 - 某专家未经授权将其他企业的核心算法代码在论坛上演示

共同WiFi加密或未加密
尽量不要连接

阅读下列材料, 回答第 3 至 6 题:

某驿站引入智慧快递系统: 快递员通过人脸识别进入驿站, 扫描包裹电子标签存入货架, 货架自动生成取件码发送给客户并上传信息。客户扫码或录入手机号, 智能门禁核验信息后开门, 同时驿站激活包裹位置提醒灯。高拍仪录入取件信息后, 提醒灯熄灭, 门禁自动开启。站内有无死角云监控系统, 管理员可通过监管系统 APP 实时查看各项数据。

- 下列关于该信息系统功能的说法, 不正确的是 **B**
 - 高拍仪、扫描包裹电子标签设备能为系统收集和输入数据
 - 该系统工作过程中无数据输出
 - 门禁系统与服务器能进行双向数据传输
 - 通过人脸识别快递员的功能属于人工智能技术的应用



4. 使用 9 位二进制生成取件码, 如某货物编码为 000010100, 取件码是 1-3-5。

0	0	0	0	1	0	1	0	0
货架号 9				层号 3		列号 4		

则货物编码为 011001101 的取件码是 **D**

- 3-1-4
- 1-13-13
- 6-1-5
- 7-2-6

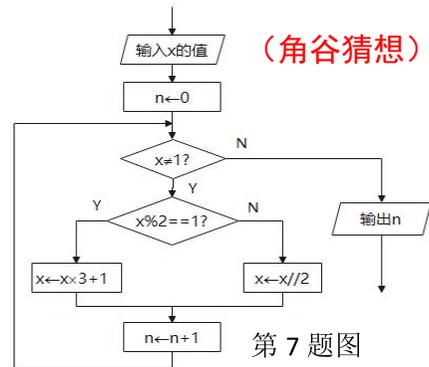
设计传感器:
摄像头
RFID读卡器
高拍仪
...

- 下列关于该信息系统中硬件的说法, 正确的是 **D**
 - 位置提醒灯是系统中的控制器 **执行器**
 - 系统中的传感器都用于获取图像数据
 - 驿站的门 **不属于** 系统的硬件
 - 智能货架和服务器可能使用 WIFI 方式通信

6. 下列关于该信息系统中软件的说法正确的是 **D**
- A. 监管系统 APP 属于系统软件 **应用软件**
 - B. 各项软件安装后 ~~不~~需要定期维护和管理
 - C. 云监控功能只需软件支持就能正常工作
 - D. 包裹的相关数据存储于数据库中

7. 某算法的流程图如第 7 题图所示，下列说法正确的是 **A**

- A. 若输入的 x 值是 5，则输出 n 的值为 5 ✓
- B. “ $x \neq 1?$ ”和“ $x \% 2 == 1?$ ”的执行次数一定相同
- C. 若输入的 x 是个偶数，则“ $x \leftarrow x \times 3 + 1$ ”将一定不会被执行
- D. 若输入的 x 是个奇数，则“ $x \leftarrow x \times 3 + 1$ ”与“ $x \leftarrow x // 2$ ”执行次数相同

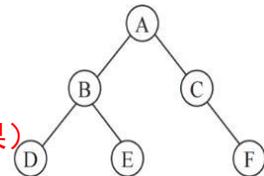


8. 有后缀表达式 “3 4 6 * + 5 2 * +”，现利用栈计算该表达式：从左向右扫描，遇到数字时，数字入栈；遇到运算符时，两个元素出栈，用运算符计算，所得结果入栈，如此反复操作，直到扫描结束，栈的深度至少是 **B**

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

9. 某二叉树如第 9 题图所示，若删除其中的一个叶子节点，则新二叉树的中序遍历结果不可能是 **DBEACF** (删除DEF后结果)

- A. EBACF ✗
- B. DBACF
- C. DBEAC
- D. BEACF



10. 数组元素 $a[0]$ 至 $a[n-1]$ 依次存放着 n 个数据，现需要使数组元素降序排序，实现该功能的部分程序段如下。

```
for i in range(1, n):
    k = a[i]
    j = i - 1
    while j >= 0 and                     (1)                     :
                            a[j+1]=a[j]                    
        j -= 1
                            a[j+1]=k                    
```

划线处可选代码有：① $a[j] < k$ ② $a[j] > k$ ③ $a[j+1] = a[j]$ ④ $a[j] = a[j+1]$
⑤ $a[j] = k$ ⑥ $a[j+1] = k$

则(1)(2)(3)处填入的正确代码应为 **B**

- A. ①③⑤
- B. ①③⑥
- C. ②④⑤
- D. ②③⑥

11. 若甲、乙程序段输出结果相同，则甲程序段加框处的正确代码为 **A**

<pre>def fun(n): if n == 10: return 1 else: return 必定有: fun(n+1) print(fun(1))</pre> <p style="text-align: center;">甲程序段</p>	<pre>r = 1 n = 10 for i in range(n-1, 0, -1): 循环9次 r = (r+1)*2 print(r) ((1+1)*2+1)*2... +1 后乘2</pre> <p style="text-align: center;">乙程序段</p>
---	---

- A. $2 * (\text{fun}(n+1) + 1)$
- B. $2 * \text{fun}(n+1) + 1$
- C. $2 * (\text{fun}(n) + 1)$
- D. $2 * \text{fun}(n) + 1$

$\text{fun}(9) = 2 * (\text{fun}(10) + 1)$

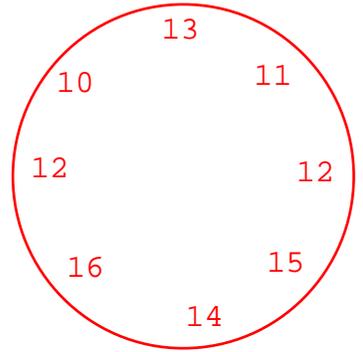
f:
fun:
function
(函数)

12. 有如下 Python 程序段: C

```
q=[13, 11, 12, 15, 14, 16, 12, 10]
n=len(q);head, tail=0, n-1
while (tail-head+n)%n>2: 跳出循环, 队列中剩2个元素
    if q[head]>q[(tail-1+n)%n]:
        q[tail]=q[head]
        tail=(tail+1)%n
    head=(head+1)%n
```

执行该程序段后, `q[head:tail]` 中的元素依次为

- A. 13, 15, 16 B. 14, 15, 16 C. 15, 16 D. 13, 15



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

13. 某单位打印室有 4 台打印机 (编号 0~3), 根据打印任务的紧急程度分为加急任务和普通任务, 两种任务的分配规则如下:

△ 任务甲为加急任务, 优先分配:

- 先选择当前空闲的打印机 (同空闲状态则编号小的优先)
- 若无空闲打印机, 则选当前任务剩余时间最短的打印机 (同剩余时间则编号小的优先)

△ 任务乙为普通任务, 按打印机累计打印份数分配, 选择累计打印份数最少的打印机 (同打印份数则打印机编号小的优先)

若甲 打印机0

若乙 打印机1

(1) 某时刻各打印机状态:

打印机 0: 空闲 (累计 200 份)	打印机 1: 忙碌, 剩余 8 分钟 (累计 160 份)
打印机 2: 忙碌, 剩余 6 分钟 (累计 160 份)	打印机 3: 忙碌, 剩余 8 分钟 (累计 200 份)

此时到达一个打印任务, 下列选项中任务和分配的打印机相匹配的是 B (单选, 填字母)

- A. 乙, 打印机 0 B. 乙, 打印机 1 C. 甲, 打印机 2 D. 甲, 打印机 3

(2) 以下 python 代码段实现打印任务的分配, 请在划线处填入合适代码。

(3) 加框处代码有误, 请改正。

`#[False, 0, 200]` 表示打印机状态 (空闲), 当前任务剩余时间, 累计打印份数

`printer=[[True, 6, 200], [True, 5, 150], [True, 8, 150], [True, 3, 220]]`

`#`获取 task 值, 如 `[True, 20]` 表示任务属性 (甲) 与需打印份数, 代码略

`def assignmen_task(printer, task):`

 if task[0]: 如果是优先任务 (甲任务)

 k=0

 从编号小的打印机开始扫描

 for i in range(len(printer)):

 if printer[i][0]==False 或 not printer[i][0]

 更新打印机状态与累计打印份数, 代码略

 return printer, i 无等号

 elif printer[i][1]<printer[k][1]

 k=i

 更新打印机状态与累计打印份数, 代码略

 return printer, k

先选择当前空闲的打印机 (同空闲状态则编号小的优先)

若无空闲打印机, 则选当前任务剩余时间最短的打印机 (同剩余时间则编号小的优先)

监测点	时间	传感器	值
1号	xxx	温度	xxx
1号	xxx	湿度	xxx
1号	xxx	pH	xxx
...

监测点	时间	温度	湿度	pH
1号	xxx	xxx	xxx	xxx
...

```

else: 如果是普通任务（乙任务）
    k=0
    for i in range(1, len(printer)):
        if printer[k][2]>=printer[i][2]:
            k=i printer[k][2]>printer[i][2]
            #更新打印机状态与累计打印份数，代码略 无等号
    return printer, k

```

选择累计打印份数最少的打印机（同打印份数则打印机编号小的优先）

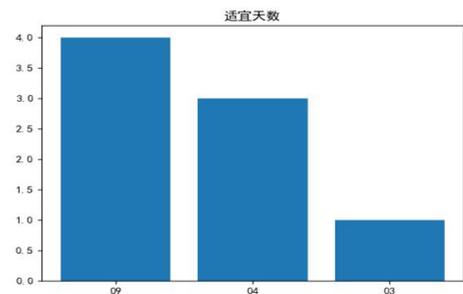
14. 某小组搭建一个多肉生长环境监测系统，采集大棚内的温度、土壤湿度和 pH 值等数据。该系统有若干个监测点，每个监测点均配备智能终端、传感器、执行器和 IoT 模块。智能终端通过 IoT 模块连接 Web 服务器上传各项数据，并从服务器获取阈值。若土壤湿度低于阈值，系统将启动洒水装置；若 pH 值异常，系统将异常信息发送至用户手机。用户可以通过浏览器查看各项数据。请回答以下问题：

- 数据库某数据表记录了各监测点分时段各类传感器上传的数据，则该表至少有 **A 或 B** (单选，填字母：A. 4 B. 5) 个字段。
- 下列关于该信息系统说法，正确的是 **AB** (多选，填字母)。(注：全部选对的得 2 分，选对但不全的得 1 分，不选或有错的得 0 分)
 - A. 各项阈值可以存储在服务器的数据库中 ✓
 - B. 传感器和执行器可以连在不同的智能终端上 ✓
 - C. 用户可以通过浏览器直接查看智能终端中的数据 ✗
 - D. 实时判断 pH 值是否异常的程序只能在服务器端运行 ✗ **可以在服务端，也可以在智能终端**
- 某监测点提交数据到服务器的 URL 为 `http://192.168.1.6:5000/toph?Tm=20&tid=3&t=5.8`，则 Web 服务器的 IP 地址为 **192.168.1.6**
- 系统运行一段时间后，发现有一个监测点的洒水装置一直在工作，请描述出现该现象的可能原因。(回答 2 项，1 项正确得 1 分)
 - 执行器故障、传感器故障
 - 智能终端缺少关闭洒水装置的程序
 - 该处实际湿度一直低于阈值
 - 服务端关闭洒水装置的阈值设置不合理
- 根据资料显示，某多肉植物最适宜的生长温度为 15℃~25℃，小明收集了当地全年每天的气温数据，部分界面如图 a 所示，统计该植物每月适宜生长的天数，按天数由高到低绘制柱形图，如图 b 所示。请在划线处填入合适的代码，完善程序。

传感器与智能终端连接故障？
 执行器与智能终端连接故障？
 智能终端故障？
 IoT模块故障？

	A	B	C
1	日期	最高气温	最低气温
2	2024-01-01	8	0.1
3	2024-01-02	8	-2
4	2024-01-03	9	-1
			
366	2024-12-30	10	2
367	2024-12-31	9	0

第 14 题图 a



第 14 题图 b

```

#导入相关库，代码略
df = pd.read_excel("2024 年气温数据.xlsx")
df["月份"]=[""]*len(df)

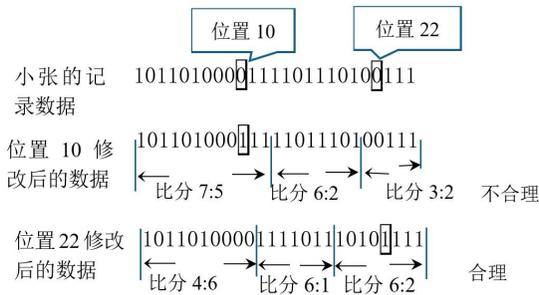
```

```

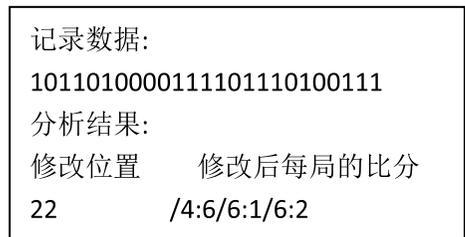
for i in range(len(df)):
    df["月份"][i]=df["日期"][i][5:7]
low=15;high=25
df1= ① B. df[df.最低气温>=low]
df2= ② F. df[df1.最高气温<=high]
A/E? df2= ③ A. df2.groupby("月份",as_index=False).count() #分组计数
df2=df2.rename(columns={"最高气温":"适宜天数"})
df3= ④ D. df2.sort_values("适宜天数",ascending=False) #降序排序
plt.bar(df3.月份, df3.适宜天数)
#设置绘图参数,显示柱形图,代码略
①②③④处可选代码有:
A. df2.groupby("月份",as_index=False).count() #分组计数
B. df[df.最低气温>=low]
C. df[df.最高气温<=high]
D. df2.sort_values("适宜天数",ascending=False) #降序排序
E. df2.groupby("月份").count()
F. df1[df1.最高气温<=high]
G. df2.sort_values("适宜天数")

```

15. 小张和小红参加校园知识竞赛,比赛采用抢答模式。每次抢答正确得1分(用“1”表示),答错或未作答得0分(用“0”表示)。当任意一方得分6分及以上且领先对手2分时,该局赢得一局。例如某局数据“10110101”表示小张得6分、小红得3分,局比分6:3。小尼用“0”与“1”数字字符串记录了小张连续若干局的竞赛得分数据,该数据存在一处错误,错误在连续多个0的最后一个位置。为了找出错误,小尼的处理方法如15题-图a所示,程序运行示例如15题-图b。



第15题-图a



第15题-图b

- 小尼的某次记录为: “1010101000101101111001111”, 按照小尼的修改方式处理后每局的比分为 4:6/6:6:2/6:1。
- 定义 Find_errors(a) 函数, 列表 a 存放小尼记录的比分。函数功能是找出疑似错误位置。

```

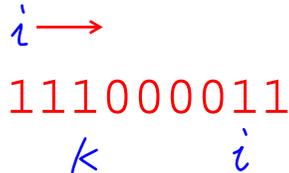
def Find_errors(a):
    head=-1; tail=-1
    link= []; i = 0

```



(2) 定义 Find_errors(a) 函数，列表 a 存放小尼记录的比分。函数功能是找出疑似错误位置。

```
def Find_errors(a):
    head=-1; tail=-1
    link= []; i = 0
    while i < len(a):
        k = i
        while i < len(a) and a[i] == 0:
            i += 1 i在遇到1的位置停下来
        if ▲i-k>=2:
            link.append([i-1, -1]) 记录多个0时，最后1个0的索引位置，并形成链表
            if tail==-1:
                head=len(link)-1
            else:
                link[tail][1]=len(link)-1
                tail=len(link)-1
                如果是单链，代码可以优化，tail可以不用 link[-2][1]=len(link)-1
        i += 1
    return head, link
```



①在划线处填入合适的代码 [[7,1],[17,-1]]

②若列表 a= [1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1], 则 link 的最终值为 ▲ 。

(3) 对每个疑似错误位置分别修改数据，并统计每局比分，请在划线处填入合适的代码。

```
def Accumulated_points(a, head, link):
    ans=[] ; p=head
    while p!=-1:
        f1 = 0 ; f2 = 0 ; flag=True
        k = link[p][0]; a[k] = 1 将索引为k的位置修改为1，说明link[p][0]存储连续个0时最后1个0的索引
        sp = str(k + 1) + "
        for j in range(len(a)):
            if a[j] == 1:
                f1 += 1
            else:
                f2 += 1
            if abs(f1-f2)==2 and (f1>=6 or f2>=6)
            sp += "/" + str(f1) + ":" + str(f2) 比分数据
            f1 = 0; f2 = 0
        if f1!=0 or f2!=0:
            flag=False
        if flag:
            ans.append(sp)
            a[k]=0 本次0的错误修改完毕，恢复列表成原始状态后，尝试修改下一处错误
            p=link[p][1]
    return ans

# 获取小尼记录数据，存入列表 a 中，代码略
head, link=Find_errors(a)
ans=Accumulated_points(a, head, link)
if len(ans)>0:
    #输出修改后的每局比分，代码略
else:
    print("记录数据有误")
```