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

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。在每小题给出的四个选项中，只有一个符合题目要求）

阅读下列材料，回答第 1 至 3 题：

某健身房推出“智能健身”模式：馆内健身设备通过人脸识别用户，在运动中自动采集用户的身高、体重、体脂率等身体指标。健身设备会根据用户的身体状况、语音指令智能调节设备。以跑步机为例，系统实时采集用户的心率、卡路里消耗量、跑步姿势等数据，与用户语音互动交流，运动结束后系统自动在“健身房 APP”给用户推送运动报告。

1. 以下关于“智能健身”系统中数据与信息的叙述，不正确的是 A

A. 跑步机采集的运动图片、语音指令都属于结构化数据 非结构化

B. 该系统推送的运动报告有助于个性化服务，体现了数据的价值性

C. 实时采集心率、卡路里消耗量以数值呈现，是数据的一种表现形式

D. 该系统在用户运动中自动采集各项身体指标，体现了数据的时效性

2. 该健身房拟新增以下功能，其中运用了人工智能技术的是 B

A. 会员等级自动升级

B. 系统语音实时纠正跑步姿势

C. 定期发送运动提醒短信

D. 运动数据同步至社交平台

判断姿势不正确属于AI

文本转语音：不一定有AI

3. 以下关于该系统中数据存储和处理的叙述，正确的是 C

A. 系统中的数据以十六进制方式编码后才能存储在计算机中

B. 将采集到的运动姿势图片存储为 bmp 格式更节省存储空间 jpg

C. 系统采集用户的跑步姿势的过程经历了数据的采样和量化

D. 健身设备发出声音与用户交流的过程是模数转换过程

阅读下列材料，回答第 4 至 6 题：

学校部署自助水果售卖系统：用户刷脸验证后打开水果柜，柜内摄像头自动识别所选水果，关门后电子秤按重量差计算价格并自动扣费，结算终端同步显示金额。消费记录实时上传服务器，用户可通过浏览器查询账单。管理员使用“售卖系统 APP”监控库存及时补货。

4. 下列关于该信息系统的描述，不合理的是 B

A. 消费记录同步至服务器，体现系统的数据存储与传输功能

B. 系统中数据的收集和输入都由电子秤、摄像头等设备实现，无需用户输入 查询需要输入数据

C. 购买水果的学生、老师和系统管理员，都属于该系统中的用户 ✓

D. 该系统降低了人工成本，但网络故障会影响系统的正常使用

5. 下列关于该信息系统中软硬件的说法，正确的是 D ✗

A. 该系统的硬件包括结算终端、摄像头、数据库等

B. 服务器的存储容量不会影响系统性能

C. 该系统只采用了 B/S 架构 c/s也有

D. “售卖系统 APP”是一款应用软件

有正有负的数中，连续和最大值

6. 下列关于该系统网络技术与数据安全的说法，正确的是 C

A. 水果售卖机不需要遵循 TCP/IP 协议

B. 水果售卖机可借助 RFID 技术接入互联网

C. 该系统使用的人脸数据属于个人敏感信息

D. 学生和系统管理员可设置相同访问权限

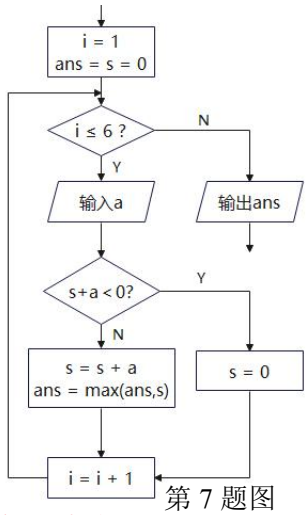
7. 某算法的部分流程图如第 7 题图所示，执行该流程时，若依次输入 2，-5，12，-4，11，-3，下列说法正确的是 C

A. 执行这部分流程后，输出的结果为 16 19

B. s 的最终值等于 ans 的最终值 s=16

C. “s = s + a”共执行了 5 次 ✓

D. 该算法流程实现了双循环结构 单循环结构



第 7 题图

8. 某二叉树只含有度为 2 和度为 0 的节点，若该二叉树中含有 5 个叶子节点，则该二叉树上的边的总数为 B

n0=5 $n_0=n_2+1$ 度为2的节点4个，度为0的5个，边为 节点总数-1：5+4-1=8

A. 7　　B. 8　　C. 9　　D. 10

9. 队列 Q 中的元素依次为 1,2,3,4，栈 S 为空。约定：P 操作是队首元素出队后入栈，B 操作是队首元素出队后入队，T 操作是指栈顶元素出栈后入队。执行操作 PPBT 后，队列 Q 中的元素依次为 A

A. 4,3,2　　B. 4,2,3　　C. 4,3,1　　D. 4,2,1

操作	Q=[1,2,3,4]	S=[]
P	[2,3,4]	[1]
P	[3,4]	[2,1]
B	[4,3]	[2,1]
T	[4,3,2]	[1]

10. 有如下 Python 函数：

```
k = 1
ch = s[0]
for i in range(1, len(s)):
    if ch == s[i]:
        k += 1
    else:
        if k == 0:
            ch = s[i]
            k = 1
        else:
            k -= 1
```

字母消消乐

ch记录消除后(k=0)时的首1个字符

变量 s 分别取下列值并运行后，变量 ch 的值不为"a"的是 C

A. "aaasd"　　B. "asdsa"　　C. "asads"　　D. "ssdaa"

11. 有如下 Python 程序：

```
import random
i , j = 0 , len(a)-1
key = random. randint(1,47)
while i<=j:
    mid = (i+j)//2          # 语句①
    if a[mid] <= key:       # 语句②
        i = mid + 1
    else:
        j = mid - 1
print(j)
```

语句①的次数=循环次数=查找次数=查找层数

0 1 2 3 4 5 6 7

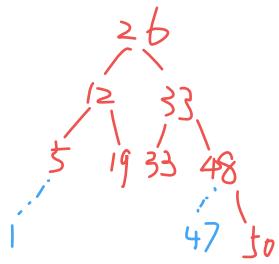
当 a 为[5,12,19,26,33,33,48,50]，运行程序，下列说法正确的是 B

A. 语句①执行的次数可能为 4 3

B. 语句①改为“mid = (i+j+1)//2”，程序结果不会改变 j|i 不变

C. 语句②改为“a[mid]<key”，程序结果不会改变 找相同数不同

D. 若生成的 key 是 33，则输出结果是 4 5 例如26，33等



12. 有如下 Python 程序段：

```
import random
k = random. randint(2,5)
i = s = cnt = 0
m = -1
for j in range(len(a)):
    s += a[j]
    if a[j] % 2 == 1:
        cnt += 1
    while (j - i + 1) > k or cnt > 1:
        if a[i] % 2 == 1:
            cnt -= 1
        s -= a[i]
        i += 1
    m = max(m, s)
```

滑窗问题，滑窗大小为k

#i是窗口左边界，s是当前窗口的和，cnt是当前窗口的奇数个数

#j是窗口右边界，遍历数组

#将a[j加入当前窗口的和

#当窗口长度>k或 奇数个数>1时，收缩左边界

#如果移出的a[i]是奇数

窗口长度小于等于k，且区间内只有1个奇数的和的最大值

#左边界移动

k=5，实际最大10,6,1,8,2，m=27

3,10,6,1,8此区域有2个奇数

若 a 为[3, 10, 6, 1, 8, 2, 5, 4]，运行程序后 m 的值不可能的是 D

k=4

A. 16 k=2 10,6 B. 19 k=3 3,10,6 C. 25 10,6,1,8 D. 28

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

13. 某地举办排球淘汰赛，参赛队伍从 1 号开始编号。每轮比赛，参赛队伍按编号从小到大依次安排比赛（如下表所示）胜者晋级，若末尾队伍无对手，则该队伍直接晋级。通常情况下，能力值高的队伍会战胜能力值低的队伍；如果**能力值相近**，则双方**均有机会**获胜。

小明编写了一个 Python 程序，模拟比赛过程并预测冠亚军，若参赛队伍能力值**相差不超过 5**，随机选择一个队伍获胜。

例如有 4 支队伍，能力值与淘汰赛程如下表所示：

第一轮	第二轮	最终结果
1 号队伍：97	1 号队伍：97	1 号队伍（冠军）
2 号队伍：86		
3 号队伍：65	4 号队伍:73（亚军）	
4 号队伍：73		

（1）在上述例子中增加 2 支队伍，5 号队伍能力值为 83，6 号队伍能力值为 76，则本次比赛的亚军为__▲5__号队伍。

（2）实现上述功能的代码如下，请在划线处填入合适的代码。

```
import random
#队伍编号与对应能力值存储在列表 s，s = [["1 号队伍", 97],["2 号队伍", 86],……]
while len(s) > 1:
    nxt = []  #建立一个空列表 nxt
```

```
for i in range(_____①_____):
    if abs(s[i][1] - s[i + 1][1]) <= 5:
        win = random.choice([s[i], s[i + 1]]) #随机选择一个获胜
    elif _____②_____:
        win = s[i]
    else:
        win = s[i + 1]
    nxt.append(win)      #将 win 添加到列表 nxt 的末尾
    if _____③_____:
        nxt.append(s[-1])    #将列表 s 最末元素添加到 nxt 的末尾
    s = nxt
print("冠军：",s[0][0])
```

0,len(s)-1,2

错误：(0,len(s)-1,2)

for i in range(0,3)
等价
for i in [0,1,2]
i最后1次值为2

方法2:
len(s)//2`
2*i 2*i+1

s[i][1]>s[i+1][1]

或 s[i][1]-s[i+1][1]>5

错误：abs(s[i][1]-s[i+1][1])>5

奇数队伍中
最后1支队伍处理

len(s)%2==1 或 len(s)%2!=0 或 i+3==len(s) 或i+3 >=len(s)

14. 小张为学校搭建教室噪音监测系统。系统通过声音传感器采集教室中的声音数据，通过 Wi-Fi 将数据传输到 Web 服务器，若噪音过高，服务器通过智能终端控制教室音响发出提醒。教师可通过浏览器登录系统，查看各个教室各时段的相关数据。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__▲A__（单选，填字母：A. 声音传感器与智能终端/B. 智能终端与服务器/C. 服务器与教室音响）

（2）小张用浏览器查看数据页面，页面动态显示最新的数据及其采集时间。系统正常工作一段时间后，发现该页面不再变化，刷新后仍不变，该现象可能是__▲C__出现故障（单选，填字母：A. 声音传感器/ B. 服务器/ C. 智能终端）。

（3）智能终端上的程序具有如下功能：每隔 2 分钟从传感器获取 1 次声音数据值，并将声音值传输到服务器端存储。部分程序如下：

```
while True:
    #myid 保存设备编号,temp 保存声音数据
    temp=pin0.read_analog()
    errno, resp = Obloq.get("input?id="+str(myid)+"&val="+str(temp), 10000)
    # 其他代码略
```

读取接在0管脚(pin0)的值

通过观察智能终端上的程序，下列说法正确的是__▲AC__（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. 服务器端程序中存在路由"/input"

B. 向服务器传输数据可以不知道服务器的地址和端口

无需知道智能终端的地址和端口

C. 每个终端设置不同的设备编号用于区分所在教室

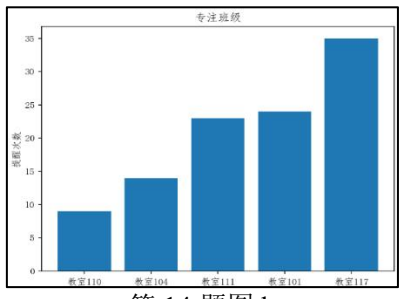
D. 执行器接在智能终端的 pin0 引脚上

传感器接pin0

（4）小张将系统中某天各教室的数据导出到文件 data.xlsx 中，部分数据如第 14 题图 a 所示。现要统计发出提醒次数（噪音值>60）最少的 5 个班级，并绘制柱形图（如图 b 所示）。

	A	B	C
1	时间	教室	噪声值
2	7:00	教室101	38
3	7:00	教室102	78
4	7:00	教室103	62
5	7:00	教室104	29
7410	21:30	教室114	59
7411	21:30	教室115	68
7412	21:30	教室116	53
7413	21:30	教室117	74

第 14 题图 a



第 14 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（填字母）。

```
import pandas as pd
import matplotlib. pyplot as plt
df=pd. read_excel("data. xlsx")
df=df. drop("时间",axis=1)      #删除“时间”列
df1 = ____①____
df1 = ____②____
df2 = ____③____
df2 = ____④____
plt. bar(df2. 教室,df2. 噪声值)
#设置绘图参数，并显示如图 b 所示的柱形图，代码略
```

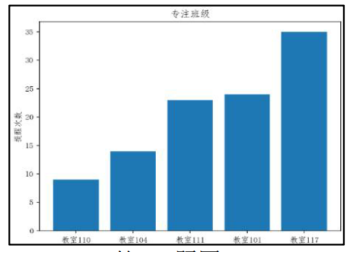
统计发出提醒次数（噪音值>60）最少的 5 个班

D

B

E

G



图表中看出结果是升序

①②③④处可选代码有：

A. df1. groupby("噪声值",as_index=False). count() #分组计数

B. df1. groupby("教室",as_index=False). count()

C. df1[df1["噪声值"]>60] #筛选

D. df[df["噪声值"]>60]

E. df1. sort_values("噪声值",ascending=True) #升序排序

F. df1. sort_values("噪声值",ascending=False)

G. df2. head(5) #获取前 5 条数据

核心问题：解决频繁提醒

调高阈值？ 冷却？ 响过3次后停止？ 仅降低采样频率？

（5）实际使用过程中，教室噪音分贝较大时教室音响频繁提醒，请给出一种可行的解决方法。

服务器记录每个教室每次发出提醒的时间，控制两次提醒的时间间隔

或 降低智能终端采集和上传数据的频率

15. 某市举行编程挑战赛，在限定的时间内，选手完成题目可以获得积分。比赛设有排行榜，显示前 5 名选手的信息，排名方式为：

①按照积分降序显示选手的编号和积分，若积分相同，则编号小的选手排在前面；

②每名选手完成新的题目后，会更新缓冲区 buf 中的相应信息；

③每过一段时间，系统会将 buf 中选手的积分数据合并到排行榜中，再删除排行榜中这些选手的重复数据，生成新的排行榜。

若当前的排行榜如第 15 题图 a 所示，此时 buf 缓冲区如第 15 题图 b 所示，将缓冲区的数据更新到排行榜后，排行榜如第 15 题图 c 所示。

排名	选手编号	积分
1	2	50
2	4	30
3	1	10
4	3	10
5	6	10

第 15 题图 a

选手编号	积分
3	50
4	70
6	20
7	10

5 50

第 15 题图 b

排名	选手编号	积分
1	4	70
2	2	50
3	3	50
4	6	20
5	1	10

5 50

第 15 题图 c

请回答下列问题：

（1）下一时间段有编号为 5 的选手得分 50 分，则排行榜更新后选手 5 的排名为 4▲ 。

（2）sort(b)函数用于对缓存区 b 中的记录按①的要求降序排序，返回一个索引列表 a。

```
#列表 b 存储缓冲区的数据，其中 b[i][0]保存编号、b[i][1]保存积分
def sort(b):
    n = len(b)
    a = list(range(len(b))) #生成列表 a = [0，1，…，n-1]
    for i in range(n - 1):
        k = i
```

等价 a=[i for i in len(b)]

索引下的选择排序比的是值，换的是索引

第一轮排序最值在1号位置，k=j赋值一次，k=1;第二轮，初值k=1，与3号位置比较时赋值一次，k=3;第三轮初值k=2，与3号位置赋值一次;总共3次

初始：[[6,80],[3,100],[7,20],[5,80]

```
    for j in range(i + 1, n):
        if b[a[k]][1] < b[a[j]][1]:
            k = j
        elif  b[a[k]][1] == b[a[j]][1] and b[a[k]][0] > b[a[j]][0]:
            k = j
    a[i], a[k] = a[k], a[i]
return a
```

错误 [2,0,3,1] 不是名次

[1,3,0,2]

原：0 1 2 3

若列表 b =[[6,80],[3,100],[7,20],[5,80]],①函数返回 a 的值为__▲__；②函数中 2 处“k = j”的执行次数共有__3▲__次。虽是索引选择排序，但k=j次数与普通选择排序相同

（3）实现题目要求的部分程序如下，输出如第 15 题图 d 所示，请在划线处填入合适的代码。

```
def insert(d, last):
    global data, head      #可在函数内修改全局变量 data 和 head 的值
    pre = p = last
    while p != -1 and (data[p][1] > d[1] or (_____①_____)):
        pre = p ; p = data[p][2]
    data. append(d)
    idx = len(data) - 1
    if p == head:
        head = idx
    else:
        data[pre][2] = idx
    _____②_____
    return idx
def update(data):
    global head
    #更新 data 列表，去掉重复元素，并返回修改后的 data，代码略
    return data

#排行榜由 data 和 head 构成，且 data[i][0]保存选手编号，data[i][1]保存选手积分
#初始 data = []，head = -1,读入 buf 后执行如下代码
last = head
a = sort(buf)
for i in a:
    buf[i]. append(-1)
    _____③_____
data = update(data)
print("当前排行榜：")
c = 0
p = head
while p != -1 and c < 5:
    print(data[p][0],data[p][1])
    p = data[p][2]
    c += 1
```

成绩相同，序号小排前

data[p][1]==d[1] and data[p][0]<d[0]

新增加的元素在data末尾


```
当前排行榜：
4 70
2 50
3 50
6 20
1 10
```

第 15 题图 d

data[idx][2]=p 或 data[len(data)-1][2]=p

buf降序排序

优化：a[:5] 缓冲区最多加入5个

last=insert(buf[i],last)

下一次插排从buf[i]后面的位置开始遍历，因为buf中的数据已经降序排序

或 insert(buf[i],last) 效率低

或 last =insert(buf[i], head)

或 insert(bufl[i],head)

错误:head=insert(bufl[i],head)