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

高三下信息技术选考练习卷（三）

一、选择题（本大题有 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 2 题。

某银行客户在柜台或 ATM 机上存、取款后，可通过手机银行 APP 实时查询余额变化、交易明细等数据。

1. 下列关于数据的说法，正确的是 D

A. 存、取款数额相同时余额未发生变化，不会产生新的数据

B. 若干年前的交易明细没有价值

C. 通过手机银行 APP 查询的数据都是非结构化数据

D. 国家监管部门分析客户的操作有助于防范电诈的发生

2. 下列操作中，不能有效提升系统数据加密措施或防入侵能力的是 B

A. 对客户的信息进行加密存储

B. 当账户变动时发送短信通知客户

C. 仅通过官方渠道下载 APP

D. 定期修改管理员密码

向客户发送短信提醒属于服务功能，主要用于提升客户的体验感，并不直接增强系统本身的数据安全防护能力

阅读下列材料，回答第 3 至 6 题。

某城市智慧交通系统实现红绿灯动态配时，车道旁的摄像头实时分析采集到的图像中行人和车辆的数量，并上传至服务器，服务器根据数量调整绿灯时间；路口设备使用 RFID 技术实时获取网联公交车、救护车的编号等数据，实现优先通行；该城市智慧交通系统还可以根据历史车流数据预测交通情况，实现红绿灯配时提前优化。

3. 下列应用中体现人工智能技术的是 A 联结主义

A. 摄像头实时分析图像获取人流量

B. 将行人和车辆的数量上传给服务器

C. 服务器发送指令改变绿灯时长

D. 路口设备使用 RFID 技术获取救护车编号

4. 下列关于该信息系统组成与功能的说法，不正确的是 C

A. 信号灯颜色改变，说明系统对数据进行了加工与处理

B. 红绿灯动态配时可以提高出行效率，节能减排

C. 机动车车牌号属于该系统中的软件 数据

D. 行人属于该信息系统的用户

5. 下列关于该系统中网络的说法，正确的是 D

A. 服务器必须部署在局域网中才能正常工作　　B. 网络摄像头工作时无需获取 IP 地址

C. 公交车只能通过 5G 联网　　D. 仅上传采集到的数据可有效节省带宽

6. 若将摄像头采集的图像存储为未经压缩的 BMP 格式文件，下列说法不正确的是 B

A. 图像采集模块的位深度会影响画质

B. 采集到的数字图像是矢量图形 位图图像

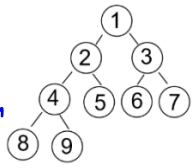
C. 图像中行人和车辆的数量与文件存储容量无关

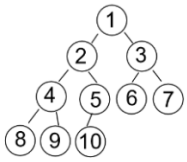
D. 为节省存储空间，可将 BMP 格式转换为 JPEG 格式

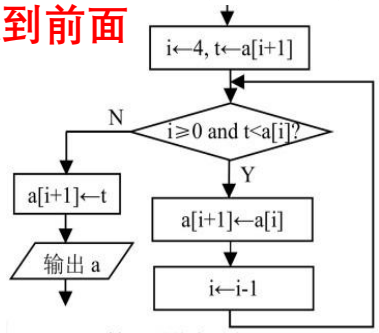
7. 某算法的部分流程图如图所示，数组 a 的值为[4,1,3,5,6,2]，执行这部分流程后，输出 a 的值为 D

A. [1,2,3,4,5,6]
B. [4,1,3,5,6,2]
C. [4,2,1,3,5,6]
D. [4,1,2,3,5,6]

将a[5]中的2以插排方式插入到前面
注意：前面不是完全有序







第 7 题图

8. 循环队列 Q 从队首到队尾的元素依次为"p","y","t","h","o","n"。约定：U 操作是指元素出队，H 操作是指元素出队后再入队。经过 UHHUHHUH 系列操作后，队列 Q 中队首到队尾的元素依次为 A

A. ont　　B. hon　　C. tno　　D. noh

n0=5
n2=4
n1=0或1

9. 已知一棵完全二叉树有 5 个叶子节点，下列关于该二叉树的说法正确的是 C

A. 度为 1 的节点一定有有 0 个　0或1个　　B. 最多有 9 个节点　9个或10个

C. 第 4 层可能有 3 个节点　　D. 深度为 5　4

10. 对于数组 a= ["6","8","1","2","1"]，分别执行以下两个程序段，说法正确的是 C

```
def quick(a):
    if len(a) <= 1:
        return a
    L="";M="";R="";p=a[0]
    for x in a:
        if x<p:
            L+=x
        elif x==p:
            M+=x
        else:
            R+=x
    return quick(L)+M+quick(R)
```

快排思想
升序排序

甲程序段

```
n=len(a);flag=True;i=0
while i<n-1 and flag:
    flag=False
    for j in range(n-i-1):
        if a[j]<a[j+1]:
            a[j],a[j+1]=a[j+1],a[j]
            flag=True
    i+=1
```

冒泡排序，降序

乙程序段

A. 两个程序段都采用了递归算法　　B. 两个程序段都将数组 a 降序排序

C. 甲程序段中加框处代码可以修改为 p=a[-1]　　D. 乙程序段中加框处代码可以修改为(n-i,i,-1)

11. 有如下 Python 程序段：用于分界的值不一定是a[0]

```
import random
a=[random.randint(1,100) for i in range(10)]  #随机生成 10 个 1~100 之间的整数
st=[0]*len(a);top=f=-1
for i in range(len(a)):
    while top>=0 and f*a[i]<f*st[top]:
        top-=1
    top+=1;st[top]=a[i]
    f=-f
print(st[0:top+1])
```

构建单调栈代码

i为偶数时，f=-1,构建从前大后小的序列
i为奇数时，f=1,构建从前小后大的序列
最后一次i=9,所以最后两个数前小后大

执行该程序段后，输出结果不可能是 D

A. [65, 78, 13, 92, 75, 85]

B. [27, 27, 88]

C. [41]

D. [6, 6, 7, 7, 88, 67, 90, 5]

12. 列表 d 存储了升序链表的节点（节点数大于 0），每个节点包含数据区域和指针区域，head 为头指针。现要仅保留链表中数据值在[start, end]范围内的节点。实现该功能的 Python 部分程序段如下，方框内应填入的正确代码为 C

```
q = p = head
while p != -1:
    if d[p][0] >= start and d[p][0] <= end:
        q = p
        p = d[p][1]
    else:
        [          ]
```

1.链表删除，通常需要知前驱

2.本题特殊　头部删除：更新head

尾部删除，需要知道前驱

○→○→○→○→○→○→○→-1

升序链表中，如果存在小于start的节点，一定是从头开始，如果存在大于end的节点一定延续到链尾

A.
`if head!=p:`
`  head=d[p][1]`
`else:`
`  d[q][1]=-1`
`p=d[p][1]`
	B.
	`if d[p][0]>end:`
	`  head=d[p][1]`
	`else:`
	`  d[p][1]=-1`
	`p=d[p][1]`

C.
`if d[p][0]<start:`
`  head=d[p][1]`
`else:`
`  d[q][1]=-1`
`p=d[p][1]`
	D.
	`if d[p][0]<start:`
	`  head=d[p][1]`
	`else:`
	`  d[p][1]=-1`
	`  p=d[p][1]`

删头

如果d[p][0]>end

直接将前驱q节点设为链尾

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. 小明模拟搭建智慧图书馆座位预定系统。智能终端连接安装于每个座位上的红外传感器和座位指示灯，红外传感器探测人体是否就座，探测结果控制指示灯的颜色（座位空闲显示绿色，否则显示橙色），并通过无线通信方式向服务器发送座位的状态信息，服务器统计当前座位状态，用户可以通过浏览器查询座位使用情况并预定座位。请回答下列问题：

（1）在搭建该监测系统时，每个智能终端连接的传感器和执行器数量合理的是 B▲（单选，填字母：A. 只能连接 1 个传感器和 1 个执行器 / B. 可以连接多个传感器和多个执行器）。

（2）智能终端通过 URL 向服务器传输数据时，需要知道 C▲（单选，填字母）。

A. 智能终端的 IP 地址　　B. 智能终端的 CPU 型号

C. 服务器中对应路由名称　　D. 服务器中数据库名称 安全行考虑，数据库名称保密

（3）现需增加图书馆内空气温湿度监测功能，在智能终端接入温湿度传感器后，必须要对软件部分做的修改有 AB▲（多选，填字母）。

A. 增加智能终端程序中向服务器上传温湿度传感器数据的代码

B. 修改服务器端程序，增加接收温湿度传感器数据并写入数据库的代码

C. 增加接收温湿度传感器数据的新的路由与视图函数的代码 不是必须

D. 修改智能终端程序中服务器地址及端口号 不需要修改

```
@app.route(/input)
def get():
    request.args.get(temp)
    request.args.get(hum)
```

一个路由中，一并读取

也可以分开提交到不同的路由

宝典p53


```
errno, resp = Obloq.get("input?x=" + str(temp) + " & y=" + str(hum), 10000)    #发送数据
```

（4）若用户未能在规定时间内到达，则系统会扣除信用分 t(每位用户原始信用分为 600)，每超时一次扣 100，实现该功能的代码为 t= max(0,t-100)，下列程序段中与其作用相同的是 AD （多选，填字母）。 t=50?

A. t-=100 if t<0: t=0	B. if t<0: t=0 else: t-=100	C. if t>100: t-=100 if t<0: t=0	D. if t>100: t-=100 else: t=0

没有提到正常运行 t可能出现负数 50分也需要扣分

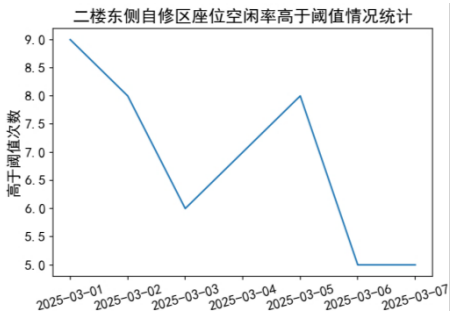
（5）系统运行后，发现某座位没有人坐但是指示灯长期保持橙色。若指示灯本身无故障，且其与智能终端的连接也无故障，请用文字描述可能的原因。（注：回答 2 个可能原因，1 项正确得 1 分。） 传感器故障、智能终端控制LED代码有误、传感器被遮挡 IOT？ xxx连接异常？

14. 智慧图书馆座位预定系统已采集了一年的数据，座位空闲率高于 0.6 的为低效区域。现要对这些数据进行分析，请回答下列问题：

（1）小明将某一周内图书馆各区域每小时的座位空闲率导出，存入 output.xlsx 文件中，如第 14 题图 a 所示，现要找出座位空闲率最高的区域，并统计该区域每日空闲率高于 0.6 的小时数量，绘制如第 14 题图 b 所示的线形图，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①②③（单选，填字母）。

时间	区域	座位空闲率
2025-06-01 10:00:00	一楼东南自修区	0.252
2025-06-01 10:00:00	二楼东侧自修区	0.766
2025-06-01 10:00:00	三楼环廊	0.423
2025-06-01 10:00:00	四楼环廊	0.460
2025-06-01 11:00:00	一楼东南自修区	0.696
2025-06-01 11:00:00	二楼东侧自修区	0.544
2025-06-07 19:00:00	四楼环廊	0.454
2025-06-07 20:00:00	一楼东南自修区	0.342
2025-06-07 20:00:00	二楼东侧自修区	0.770
2025-06-07 20:00:00	三楼环廊	0.226
2025-06-07 20:00:00	四楼环廊	0.445

第 14 题图 a



第 14 题图 b

```
df=pd.read_excel("output.xlsx")
df["日期"]=df["时间"].astype(str).str[:10]    #获取时间列中的年月日
df1=df.groupby("区域",as_index=False)["座位空闲率"].mean()  根据区域求空闲率平均值
df2= ①A    找空闲率最高的区域，降序
oid=df2.at[0,"区域"]
df_qy= ②E    筛选（对原始数据重新筛选）
df3=df_qy[df_qy.座位空闲率>0.6]
df4= ③C    统计每日空闲率高于0.6的小时数量
plt.plot(df4["日期"],df4["座位空闲率"])
#设置绘图参数，显示如第 14 题图 b 所示的线形图，代码略
```

程序中①②③处可选代码有：AEC

A. df1.sort_values("座位空闲率",ascending=False,ignore_index=True)
B. df1.sort_values("座位空闲率",ascending=True,ignore_index=True)
C. df3.groupby("日期",as_index=False).座位空闲率.count()
D. df1.groupby("时间").座位空闲率.mean()
E. df[df.区域==oid]
F. df2[df2.区域==oid] ?

（2）通过提升硬件和服务，更多用户预约座位空闲率最高的区域。将该区域最新一周的座位空闲率存储于 data 列表中，要求改小某个高于 0.6 的值，求最长连续序列，其中每个值都小于 0.6（若有多个长度相同的子序列，选择最早出现的）。示例执行结果如第 14 题图 c 所示。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。

```
示例数据：[0.666, 0.544, 0.653, 0.478, 0.462, 0.837, 0.429]
最佳修改位置：2
最长子序列长度：4
子序列起始位置：1
```

第 14 题图 c

bug:如果都小于0.6
max没有赋值
2.代码效率低

```
length=0;start=0;pos= -1
for i in range(  ①  ):
    if data[i] > 0.6:
        mdata = data[:]       # 复制列表 data 数据
        mdata[i] = 0.5        # 修改为小于 0.6 的值
        curlen = 0;max = 0;start_pos = 0;temp = 0
        for j in range(len(mdata)):
            if mdata[j] < 0.6:
                if curlen == 0:
                    ②
                curlen += 1
                if curlen > max:
                    max = curlen
                    start_pos = temp
            else:
                curlen= 0
        if  ③ :
            length=max
            start=start_pos
            pos=i
```

len(data)

检测所有数据，枚举法把所有大于0.6的都改一次

第一次出现小于0.6，记录开始位置存入

temp=j

max>length

#输出最佳修改位置 pos、最长子序列长度 length 和子序列起始位置 start，代码略

15. 某医院有 n 个手术室，除紧急手术外，使用手术室都需要提前若干天预约。每天零点前系统对每个手术室当天的预约按开始时间进行升序排序，并将当天零点之后新增的预约手术插入到已排序好的数组中（时间已转换为分钟），如第 15 题图 a 所示。为提高预约成功率，系统升级，用户输入预估手术时长，系统自动统计当天各手术室空闲时间，显示存在指定预约时长的手术室的开始和结束时间，程序执行后如第 15 题图 b 所示。编写 Python 程序模拟该系统功能。

```
输入当日使用手术室的新预约：宫腔镜手术,OR1,15:00,16:40
预约成功！
{'OR1': [['膝关节置换术', 'OR1', 480, 540], ['痔疮手术', 'OR1', 600, 720], ['宫腔镜手术', 'OR1', 900, 1000]], 'OR2': [['冠状动脉搭桥术', 'OR2', 570, 810]], 'OR4': [['肺移植手术', 'OR4', 610, 1000]], 'OR3': [['肠道肿瘤切除', 'OR3', 900, 1050]]}
```

第 15 题图 a

系统升级后

```
请输入预估手术时间(分钟)：30
手术室OR1的空闲时间：  [['09:00', '10:00'], ['12:00', '15:00'], ['16:40', '17:30']]
手术室OR2的空闲时间：  [['08:00', '09:30'], ['13:30', '17:30']]
手术室OR3的空闲时间：  [['08:00', '15:00']]
手术室OR4的空闲时间：  [['08:00', '10:10'], ['16:40', '17:30']]
```

第 15 题图 b

binary

（1）定义 bsearch(new,array)函数，参数 new 与参数 array 的每个元素均由手术名称、手术室编号、手术开始时间、手术结束时间 4 项构成，函数功能是查找新增预约插入数组的合适位置，定义 conflict 函数，函数功能是判断新增预约是否与数组内元素存在冲突，若新预约的时间和当前预约的时间重叠，则判断为存在冲突。请在划线处填入正确的答案。

①若图 a 中，将新预约修改为：“宫腔镜手术，OR1，9:30，12:30”，则有▲（填写：有/没有）存在冲突。

570~750

```
def conflict(new,array,mid):  #冲突检查函数，若存在冲突返回 True，否则返回 False
    #代码略
def bsearch(new,array):       #二分插入函数，查找新增预约插入数组的合适位置
    left,right=0,len(array)
    while left<=right:
        mid=(left+right)//2
        if conflict(new,array,mid):
            return "fail"          #插入失败
        elif new[2]>array[mid][2]:
            left=mid+1
        else: <=
            right=mid-1
    return left ②
```

{'OR1': [['膝关节置换术', 'OR1', 480, 540], ['痔疮手术', 'OR1', 600, 720], ['宫腔镜手术', 'OR1', 900, 1000]], 'OR2': [['冠状动

new：[手术名称、手术室编号、手术开始时间、手术结束时间]

二分查找确定插入位置
此时插入都是发生在left中

或right+1

（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。

```
'''当天的手术室预约存入 res 数组，数组中的每个元素包含手术名称、手术室编号、手术开始时间、手术结束时间 4 个数据项（时间已转换为分钟，如“08:00”转换为 480），例如[["膝关节置换术","OR1",480,540]……]；key 为新增的预约在当天的信息，例如["痔疮手术","OR1",600,720]'''
sort(res)     #将预约按手术开始时间升序排序
pos={}
for j in range(len(res)):          #将已排序的预约按预约手术室分类
    if res[j][1] in pos:
        pos[res[j][1]] +=[res[j]]
    else:
        pos[res[j][1]]=[res[j]] ①
#输入当日使用手术室的新预约 key，代码略
ins=bsearch(key,pos[key[1]])
if ins=="fail" :
    print("预约冲突！")
else:
    print("预约成功！")
    pos[key[1]] = pos[key[1]][:ins] + [key] + pos[key[1]][ins:]
```

投桶

```
{'or1':[[x,x,x,x],[...]]
 'or2':...
}
```

不能用append

对应手术室预约内容更新

（3）系统升级后的部分 Python 程序如下，请在划线处填入合适的代码。

```
'''
输入手术室早、晚开放时间，分别存储在变量 open、close 中，如 open=480，代表早上 8：00
开放，代码略
'''
def cf(n):  #将时间 683 转换成“11:23”的形式
    #代码略
OR=["OR1","OR2","OR3","OR4"]
min=int(input("请输入预估手术时间(分钟)："))
for i in OR:
    a=pos[i]          ②
    b=[]
    for j in range(len(a)):
        if j==0 and a[j][2]-open>min :
            b.append([cf(open),cf(a[j][2])])
        if j==len(a)-1:
            if close-a[j][3]>min:
                b.append([cf(a[j][3]),cf(close)])
        elif j>0          ③:
            b.append([cf(a[j][3]),cf(a[j+1][2])])
    if b :
        print("手术室"+i+"的空闲时间：",b)
```

实现图b的功能：输入预估时间，然后整理手术室的空闲时间

```
{'or1':[[x,x,x,x],[...]]
 'or2':...
}
```

i是“OR1”“OR2”等

第一个时段判断：a[j][2]-开放时间>预估时间，那么前面这段时间(开门时间~第1台手术开始时间)可以成为空闲时间

说明空闲时间要>min

最后一个时段判断（最后手术结束到关门时间>30），可以成为空闲时间

/len(a)>=2 and a[j+1][2]-a[j][3]>min

中间时段判断：后一个开始时间-前一个结束时间>预估时间，就符合要求