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

高三上信息选考练习卷9

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、错选、多选均不得分。）

阅读下列材料，回答第 1 至 3 题：

某学校搭建了一个智慧体育系统，当系统通过摄像头检测到学生举手时，采用人脸识别技术确认学生身份，记录并存储学生的运动照片和运动时长。每隔一段时间，系统经数据整理和分析后形成学生运动建议。

联结主义关键词
数据训练、数据驱动、神经网络
算法模型先要进行数据训练，得到数据训练的结果文件
真正识别的时候不需要数据训练
数据训练的这些图片，不需要事先定义特征

1. 下列关于该系统中数据与信息的叙述，正确的是 D
   A. 学生的信息仅由图片的形式呈现
   B. 系统中的数据都是结构化数据
   C. 系统将学生甲识别成了学生乙，这一信息不具有任何价值
   D. 系统分析学生运动数据的过程中会产生新的信息

2. 该系统人脸识别技术是基于神经网络方法实现的，下列说法不正确的是 A
   A. 人脸识别时需要学生出现在摄像头前，形成了混合增强智能的形态
   B. 训练该人工智能模型不需要定义人脸五官位置和特征等知识 ✓
   C. 该人脸识别的过程是联结主义人工智能的应用 ✓
   D. 优化人脸识别的算法可以提高识别的准确度 ✓

3. 系统拍摄的某张照片为 600*800 像素、256 色、BMP 图像格式，经下列操作后，图像存储容量改变的是 B

8位
bmp：未压缩
jpg：有损压缩

   A. 将图像转换成 8 位灰度模式并保存
   B. 将图像转换并保存为 JPEG 格式
   C. 将图像旋转 90° 并保存 总像素未变化
   D. 降低图像亮度并保存 bmp大小与内容无关

阅读下列材料，回答第 4 至 6 题：

某博物馆启用了在线票务系统，参观者可使用手机注册并登录该系统的 App，选择参观场馆和时间并购买门票。门票数据上传至服务器，同时参观者可将门票二维码以图片形式下载至手机。系统管理员可通过 App 调整并发布博物馆的服务安排计划。

信息系统缺陷（必然存在）：
外部环境依赖
本身有安全隐患
数字鸿沟

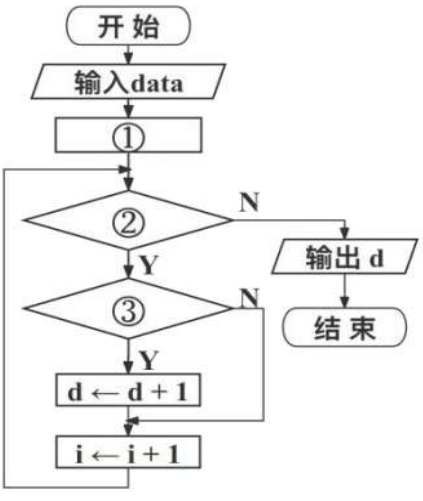
4. 下列关于该信息系统功能和应用的描述，不正确的是 C
   A. 系统可通过二维码扫描仪读取参观者门票数据
   B. 参观者可使用 App 通过移动通信网络登录该系统
   C. 系统配备 UPS（不间断电源）即可克服对外部环境的依赖 ✗
   D. 系统中的门票数据可为管理员调整服务安排计划提供依据

5. 下列关于该信息系统组成的说法，正确的是 A
   A. 管理员属于该系统的用户 ✓
   B. 该系统的 App 属于系统软件 应用软件
   C. 该系统的硬件仅指服务器+网络设备
   D. 用户手机中的数据都是该系统的数据 ✗ 部分数据

6. 下列措施不能提升该系统安全性的是 D
   A. 要求参观者实名注册 App 账号
   B. 要求用户账号只供自己使用
   C. 每隔一段时间检测系统的安全漏洞
   D. 给管理员与参观者分配相同的权限 ✗

7. 已知非降序列表 data 由 n 个整数组成（n>0），求 data 中不同整数的个数。实现该功能的算法流程图如第 7 题图所示，则①②③处的表达式应为 C

相等或者升序



第 7 题图

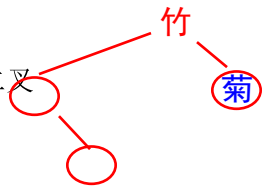
   A. ①d←0,i←0 ②i<n? ③data[i]<data[i+1]?
   B. ①d←0,i←0 ②i<n-1? ③data[i]<data[i+1]?
   C. ①d←1,i←1 ②i<n? ③data[i-1]<data[i]? ✓
   D. ①d←1,i←1 ②i<n-1? ③data[i-1]<data[i]?

3, 7, 4, 1, 2, 6 3 2

8. 队列 Q 从队首到队尾的元素依次为 3,7,4,1,2,6。重复执行如下操作，直至队列剩余一个元素：队首元素 h 出队，若队尾元素 t 大于 h，则将 t-h 的值入队，否则不入队。队列中最后剩余的元素是 C

A. 0　　B. 1　　C. 2　　D. 3

9. 已知某二叉树包含 3 个节点，其前序遍历结果为“竹梅兰”，若增加一个节点“菊”，新二叉树的中序遍历为“梅兰竹菊”，则该二叉树的后序遍历为 B



A. 梅兰菊竹　　B. 兰梅菊竹　　C. 竹菊兰梅　　D. 菊梅竹兰

10. 对于任意正整数 n（$n \leq 10^{100}$），调用函数 cal (n) 返回的结果为 n 的各位数之和，要实现上述功能，如下程序段中，加框处的代码不能为 D

梅 兰 竹 菊

```
def cal(n):
    if n < 10:
        return n
    return [          ]
```

A. cal(n//10)+n%10　　B. cal(n//10)+cal(n%10)

C. cal(n//100)+n%10+n//10%10　　D. cal(n//100)+cal(n%100)

11. 有如下 Python 程序段：

```
from random import randint
b = randint(2,3)                    #randint(2,3)随机生成 2 或 3
for i in range(len(a)-b):
    for j in range(b, len(a)-i):
        if a[j]%b > a[j-1]%b:
            a[j], a[j-1] = a[j-1], a[j]
```

若序列 a 的长度为 6，运行该程序段后，a 的值可能为

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

12. 有如下Python程序段：

```
def max_w(nums, k):
    dq=[]
    result=[]
    for i in range(len(nums)):
        while len(dq)>0 and nums[dq[-1]]<nums[i]:
            dq.pop()          #去除dq最后一个元素
        dq.append(i)          #为dq添加一个元素i
        if len(dq)>0 and i-dq[0]>k:
            dq.pop(0)         #去除dq第一个元素
        if i>=k-1:
            result.append(nums[dq[0]])
    return result
print(max_w(nums,k))
```

若nums为[1,3,-1,-3,5,3,6,7]，k为3，运行该程序段后，输出的结果是 A

A. [3,3,5,5,6,7]　　B. [-1,-3,-3,-3,3,3]　　C. [3,3,-1,5,5,6]　　D. [1,3,3,3,5,5,6,7]

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

13.有两个仅包含数字字符的字符串 s1 和 s2，它们的长度分别为 n 和 m。现要从这两个字符串中各取若干字符，字符数总和为 k（$k \leq n+m$），重新拼接这些字符，在保持所取字符在原字符串中相对顺序不变的情况下，组成一个最大的字符串。例如，有字符串 "392" 和 "768"，从其中取出 4 个字符能组成最大的字符串为 "9782"。请回答下列问题：

（1）若字符串 s1 的值为 "506"，s2 的值为 "503"，则从其中取出 5 个字符能组成的最大字符串为 56503

（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。

```
def select(s, d): 此函数可推断：整体算法采用枚举算法
    #返回由字符串 s 中 d 个字符组成的最大字符串，其中字符的相对顺序不变，代码略
#读取 s1、s2、n、m、k，代码略
res=""①
for p in range(k+1):
    if p <= n and k-p <= m:                                    选哪个3？
        x = select(s1, p)                                      x = "836"
        y=select(s2,k-p)                                       y = "438"
        tmp = ""
        i, j = 0, 0   归并的过程中，碰到相等的数？
        while i < len(x) or j < len(y):      x[i:]>y[j:] 或x[i:]>=y[j:]
            if j >= len(y) or i < len(x) and (_____③_____)):
                tmp += x[i]        或x[i]>y[j] or x[i]==y[j] and x[i+1:]>y[j+1:]
                i += 1             或 x[i]>y[j] or x[i]==y[j] and x[i:]>y[j:]
            else:
                tmp += y[j]
                j += 1
        if tmp > res:
            res = tmp
print("组成的最大字符串为：",res)
```

14.小智设计了一款智能大棚监测系统，在每个大棚设置一个环境监测点。每个监测点的智能终端连接传感器，每隔 1 小时采集 1 次温度和光照数据，通过无线通信方式上传至服务器；服务器分析数据并判断有异常情况时，通过监测点智能终端控制执行器调节温度或光照。用户可通过浏览器查看实时与历史环境数据。请回答下列问题：

（1）该系统服务器中的数据__B__（单选，填字母：A. 需要传输至传感器 / B. 可以传输至智能终端 / C. 只能传输至浏览器）　传感器→智能终端

（2）为使系统能采集到更准确的环境数据，下列做法不合适的是__C__（单选，填字母：A. 增加传感器的数量 / B. 更换灵敏度更高的传感器 / C. 增加传感器的采集间隔时长）应减少采集间隔时间，取平均值

（3）该系统服务器端程序采用 Flask Web 框架编写，服务器的 IP 地址是 192.168.1.100，端口是 8000，部分功能页面规划如下表所示。

序号	访问地址	功能说明
1	/look?d=20251104	查看某日的历史数据
2	/input?t=27&s=600	提交传感器数据

某监测点智能终端的 IP 地址是 192.168.1.1，若该监测点某次采集到的温度 t 为 27，光照数据 s 为 600，则该监测点将数据提交到服务器的 URL 为 http://192.168.1.100:8000/input?t=27&s=600

（4）系统正常运行一段时间后，发现某监测点无法调节温度。假设仅传感器、执行器这两个设备可能存在故障，更换新的传感器后发现温度仍然无法调节，再更换新的执行器后，温度可正常调节。根据上述现象，说明__B__（选填：A.传感器/B.执行器）一定存在故障。在上述操作的基础上，请你描述判定原来另一个设备是否存在故障的方法。可能1个坏，或全坏　此时系统正常工作　确定执行损坏　题干：在更新后能正常工作的基础上，如何判断另一个设备是否存在故障

（5）小智将某日 8 个监测点的温度数据导出到文件 data.xlsx 中，部分数据如第 14 题图 a 所示。若温度数据低于 22 或超过 28，则认为温度异常。现要统计每个监测点温度异常的数据条目数量，按数量升序排列后绘制如第 14 题图 b 所示的柱形图。　（有变化且是环境真实值）

更换为原传感器后改变环境温度，浏览器查看实时温度是否发生变化，如有变化，则传感器正常，反之则原传感器故障。

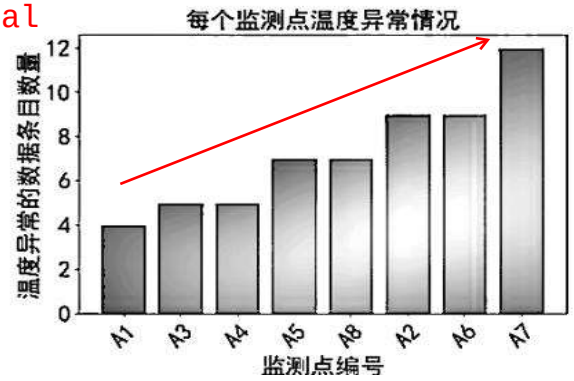
更换为原传感器后温度可正常调节，则说明传感器未故障；更换为原传感器后温度不可正常调节，则说明传感器故障等

（5）小智将某日 8 个监测点的温度数据导出到文件 data.xlsx 中，部分数据如第 14 题图 a 所示。若温度数据低于 22 或超过 28，则认为温度异常。现要统计每个监测点温度异常的数据条目数量，按数量升序排列后绘制如第 14 题图 b 所示的柱形图。

监测点编号	时间	温度数据
A1	2025071602	29
A2	2025071601	22
A3	2025071601	22
A4	2025071601	27
A5	2025071601	25
A6	2025071601	28
A7	2025071601	20
A8	2025071601	28
A1	2025071602	26

abnormal

第 14 题图 a



第 14 题图 b

实现上述功能的部分 Python 程序如下：

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel("data.xlsx")
list1 = [0] * len(df)
for i in range(len(df)):
    t = df.at[i,"温度数据"]        #通过行、列标签取单个值
    if t < 22 or t > 28:
        list1[i] = 1 异常为1
df["abnormal"] = list1              #创建列"abnormal"，列内容为 list1
                                    
plt.bar(df.index, df.abnormal)      #绘制柱形图
#设置绘图参数，显示如图第 14 题图 b 所示的柱形图，代码略
```

①请在程序中划线处填入合适的代码。

②方框中应填入的语句依次为 ADC （选 3 项，填字母序列，少选、多选、错选或次序错均不得分）

A. df = df [df ["abnormal"]!=0]　　#筛选

B. df = df[df["abnormal"]==0]

C. df = df.sort_values ("abnormal")　　#升序排序

D. df = df.groupby ("监测点编号").count ()　　#分组计数

E. df = df.groupby ("abnormal").count ()

15.某团购平台提供生鲜下单与配送服务，用户下单时选择期望配送的时段（用从 1 开始的整数表示），平台会按此时段安排配送，也可能依据实际情况延迟配送。平台每个时段最多可完成 n 个订单的配送，同一时段内配送顺序遵循如下规则：根据延迟时长从大到小安排配送；若延迟时长相同或均无延迟，则按订单号从小到大安排配送。当前时段未配送的订单延至下一时段，若某订单延迟超过 m 个时段，就转派至第三方配送。

平台提供订单修改功能，同一个订单号可能对应多条记录，保留最后提交的记录并予以配送。现给定所有订单的数据表，每条记录包含提交时间、订单号和期望配送时段，并按提交时间的顺序排列，订单号按照提交时间从 1 开始编号。

在第 15 题图所示的样例中，n、m 分别为 2、1，根据图中前三行数据计算出需要保留的记录、订单的实际配送时段和转派情况。

提交时间	08:10	08:32	09:20	09:45	10:10	10:12	10:55	11:00	11:21
订单号	1	2	3	4	5	6	7	3	5
期望配送时段	1	1	1	1	2	1	2	3	1
是否保留	是	是	否	是	否	是	是	是	是
实际配送时段	1	1		2		转派	3	3	2

提交时间	08:10	08:32	09:20	09:45	10:10	10:12	10:55	11:00	11:21	12:00
订单号	1	2	3	4	5	6	7	3	5	5
期望配送时段	1	1	1	1	2	1	2	3	1	3
是否保留	是	是	否	是	否	是	是	是	是	是
实际配送时段	1	1		2		转派	3	3	2	4

请回答下列问题：

（1）若在第 15 题图所示的样例中增加一条提交时间为 12:00、订单号为 5、期望配送时段为 3 的记录，那么需要转派的订单数量应为 0

（2）定义如下 filter (tasks) 函数，tasks 列表中每个元素包含 3 个数据项，依次为提交时间、订单号和期望配送时段，列表已按提交时间的顺序排列。函数功能是保留每个订单最后提交的一条记录，并根据订单号对保留的订单进行升序排列。

```
def filter(tasks):
    lst = [-1] * 1000                #订单号范围是 1~999
    mx = 0
    for i in range(len(tasks)):
        sid = tasks[i][1]
        lst[sid]=i
        if mx < sid:
            mx = sid
    res = []                         #语句 A
    for k in range(1,mx+1):
        if lst[k] != -1:
            res.append(tasks[lst[k]])
    return res
```

[["08:00", 1, 1], ["08:10", 2, 2], ["08:34", 1, 2], ["08:44", 3, 1], ["09:05", 2, 1],

订单号去重
方式：
利用桶计数方式
记下每个订单最后的出现的索引位置

顺便记录下最大订单号

①请选择合适的语句填入程序中划线处 A （单选，填字母）

A. lst[sid]=i　　B. lst[i]=sid　　C. lst[i]=mx

②若 tasks 为 [["08:00", 1, 1], ["08:10", 2, 2], ["08:34", 1, 2], ["08:44", 3, 1], ["09:05", 2, 1], ["09:20", 4, 3], ["09:30", 3, 2]]，调用 filter (tasks) 函数，则执行到语句 A 时，变量 mx 的值为 4

（3）实现按照规则输出转派订单的数量，以及其余订单实际配送时段的 Python 程序如下，请在划线处填入合适的代码。

```
def solve(tasks, n, m):
    last = 0
    for s in tasks:
        if s[2] > last:
            last = s[2]
    info = [[-1, -1, 0] for i in range(last + 1)]
    nxt = [-1] * len(tasks)
    for i in range(len(tasks)):
        stime = tasks[i][2]
        if info[stime][0] == -1:
            info[stime][0]=i
        else:
            nxt[info[stime][1]] = i
        info[stime][1] = i
        info[stime][2] += 1
    curtime = 1
    t = 1
    num = 0
    ans = []
```

找到最晚的配送时段，存入last

链队 每个时段 起始配送订单、结束配送订单、待处理订单数量

和info链队对应的 指针域

订单期望的配送时段，

① info[stime][0]=i 入队且成为链头

入队，成为队列中的尾

info[1]: [起、终、数量]
info[2]: [起、终、数量]
....

```
while num < len(tasks):
    count = 0
    for p in range(t, curtime + 1):
        if count >= n or p > last:
            break
        while
            ans.append([tasks[info[p][0]][1], curtime])
            count += 1
            info[p][2] -= 1
            info[p][0] = nxt[info[p][0]]
    num += count
    curtime += 1
    while t + m < curtime:
        num+=info[t][2]
        t += 1
return num - len(ans), ans
```

说明num为成功配送+失败而转派的记录总数

轮询从t开始到curtime为止的待处理订单，for循环结束，则curtime+1

超过限定n或超过最晚期望时段

count<n and info[p][2]>0 或count<n and info[0]!=-1

队列中第1个订单加入ans，配送时段为curtime

处理时段p需要完成的配送订单，在当前配送量count未达上限且时段p仍存在待配送订单时，在循环内逐个记录配送订单并更新相关变量

出队

整体思路：
一次while循环
处理一个实际配送时段
处理过程中要兼顾遗留订单
以及超时转派处理

m 超时限制时间

统计延迟超过m个时段、需要转派的订单。num变量用于累计任务总数，此时时段t对应的info[t][2]即为需要移交转派部分对应的余量，该部分订单数也需要累加到 num中

"""读取 n、m；读取订单数据表存入 tasks 列表，每个元素包含 3 个数据项，依次为提交时间、订单号和期望配送时段。代码略。"""

tasks = filter(tasks)

fails, ans = solve(tasks, n, m)

#输出转派订单的数量 fails，以及列表 ans 中订单的实际配送时段，代码略

思路2：
第1次处理 真实时段为1的订单
处理不了加入到2的时段的head
第2次处理 真实时段为2的订单
处理不了的加入到3...
...

info[1]: [起、终、数量]
info[2]: [起、终、数量]
....