

2025 学年第一学期浙南名校联盟十月联考

高三年级技术学科 试题

考生须知：

1. 本卷共 12 页满分 100 分，考试时间 90 分钟；
2. 答题前，在答题卷指定区域填写班级、姓名、考场号、座位号及准考证号并填涂相应数字。
3. 所有答案必须写在答题卷上，写在试卷上无效。
4. 考试结束后，只需上交答题卷。

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 2 题。

某电商平台支持用户线上下单、线下扫码自提，平台会记录用户订单信息、商品库存数据、商品评价数据，并为会员推送个性化优惠活动。

1. 下列关于数据的说法，不正确的是 **C**
  - A. 数据可视化可以直观体现商品销售趋势
  - B. 该平台中的数据一般以文件的形式在计算机中存储
  - C. 通过消费数据实时分析精准推送活动体现了批处理计算 **实时数据处理采用流计算**
  - D. 平台对用户订单数据的分析用于优化商品摆放位置体现数据的价值性
2. 下列措施中，不能有效提升该电商平台数据安全的是 **D**
  - A. 限制单个账号的登录设备数量
  - B. 为不同用户设置不同的访问权限
  - C. 部署防火墙防止非法访问服务器
  - D. 增加电商平台的数据存储容量

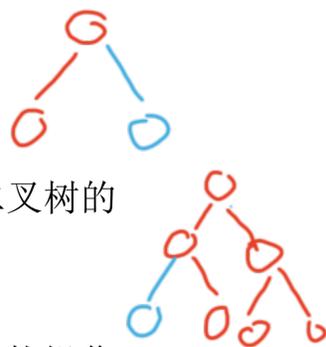
阅读下列材料，回答第 3 至 7 题。

某智能家居控制系统包含智能门锁、温湿度传感器、远程控制 APP 等。智能门锁通过指纹识别验证用户身份，温湿度传感器实时采集室内环境数据并传输至云端服务器，APP 通过语音助手接收用户指令并远程操控家电（如调节灯光亮度）。

3. 在用户通过语音助手远程调节灯光亮度的过程中，没有用到的技术是 **C**
  - A. 控制技术
  - B. 网络技术
  - C. 射频识别技术
  - D. 人工智能技术
4. 下列关于该系统中语音助手功能和设计的说法，正确的是 **A**
  - A. 语音助手接收的指令需转换为二进制形式才能被设备识别
  - B. 语音助手的所有功能必须依赖云端服务器才能实现 **不是所有功能都需要云端**
  - C. 新增方言识别功能无需修改代码 **新增功能肯定需要修改代码**
  - D. 语音助手的响应速度仅由网络延迟决定 **网络延迟会影响响应速度，但不是唯一**
5. 下列关于该系统中硬件和数据传输的说法，正确的是 **B**
  - A. 该系统的硬件包含智能门锁、温湿度传感器和远程控制 APP **APP是应用软件**
  - B. 智能门锁的指纹识别设备属于输入设备
  - C. 数据从传感器传输至云端不需要网络协议实现
  - D. 手机 APP 与云端通信时，数据只能通过移动通信网络传输 **还有计算机网络**
6. 将语音助手采集的用户指令存储为 WAV 格式音频文件，下列说法不正确的是 **B**
  - A. 采集用户指令是模数转换过程
  - B. 用户指令的音量会影响音频文件存储容量 **存储容量与采样频率、量化位数、时间有关**
  - C. 将该音频转换成 MP3 格式可以节省存储空间
  - D. 采样频率过低可能导致指令语音模糊不清

7. 智能门锁的指纹识别功能基于深度学习算法实现，下列说法不正确的是 C

- A. 指纹识别模型的训练需人工标注指纹样本的类别
- B. 指纹识别功能使用了数据驱动的人工智能方法
- C. 识别时需收集指纹的关键特征与指纹的相关知识 **不需要指纹的相关知识**
- D. 模型训练时使用的样本数量越多，识别准确率不一定越高



8. 现有某二叉树，向其中添加一个节点 x 后，该二叉树是完全二叉树，且节点 x 在新二叉树的前序遍历序列中处于第 3 个位置。则原二叉树的节点数量最少与最多分别是 A

- A. 2 6
- B. 3 6
- C. 2 7
- D. 3 7

9. 栈初始为空，元素“松”“柏”“竹”“梅”“兰”依次入栈，经过一系列入栈、出栈操作后，栈又为空。已知“梅”是第 2 个出栈的元素，则第 3 个出栈的元素不可能是 A

- A. 松
- B. 柏
- C. 竹
- D. 兰

10. 有如下 Python 程序段：

```

a = [0]*10
c = 0
n = int(input())
while n > 0:
    a[n % 10] += 1
    if a[n % 10] == 1:
        c += 1
    n //= 10
print(c)

```

**统计n中出现的数字的个数  
投桶：存在性验证**

运行该程序后，输出结果为 5，则键盘输入内容为 B

- A. 975357
- B. 876985
- C. 808890
- D. 4326479

11. 定义如下函数，返回 key 在列表 data 中的下标，若 key 不存在，则返回-1。

```

def search_key(data, key):
    i, j = 0, len(data) - 1
    while i <= j:
        m = (i + j) // 2
        for pos in [m, m + 2]:
            if i <= pos <= j and data[pos] == key:
                return pos
        if data[m] < key:
            i = m + 1
        else:
            j = m - 1
    return -1

```

**对分一次，完成m、m+2两个位置上数值的查找比对**

**以升序的方式找**



若调用函数查找 key = 22 时返回结果为-1，则 data 可能是 D

- A. [35, 30, 25, 22, 20, 15, 10]
- B. [10, 15, 20, 22, 25, 30, 35]
- C. [35, 22, 30, 25, 20, 15, 10]
- D. [10, 22, 15, 20, 25, 30, 35]

12. 使用列表 d 模拟链表结构，头指针为 head，每个节点包含数据区域（0~9 的整数）和指针区域。链表的节点按逆序存储一个非负整数，例如链表 3->2->1 表示数值 123。现需要将该数值加 c (0<=c<=9)，并将结果以同样的链表结构存储，头指针仍为 head。

```

c = int(input())
head = 0
p = head
prev = -1

```

```

while p != -1:
    d[p][0] += c
    c = d[p][0] // 10
    d[p][0] %= 10
    prev = p
    p = d[p][1]
if c > 0:
    

```

**c存放进位值**

**最高位有进位，需要添加一个节点，加在链尾，前面链表遍历结束时，p==-1,prev是链尾节点**

实现该功能的程序段如下，方框中应填入的正确代码为 **C**

A. d.append([c, prev]) d[prev][1] = len(d) - 1	B. d.append([c, head]) head = len(d) - 1
C. d.append([c, p]) d[prev][1] = len(d) - 1	D. d.append([c, -1]) head = len(d) - 1

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

13. 某快递站点记录包裹的配送方式，用“0”表示上门配送，“1”表示驿站自提。现需从一天的配送记录中，找出最长的一段连续记录，要求这段记录中上门配送和驿站自提的包裹数量恰好相等。请回答下列问题：

(1) 若给定记录为 [0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0]，则最长符合条件的连续记录长度为 10 ▲

(2) 实现上述功能的 Python 程序如下，请在划线处填入合适的代码。

# 读取包裹的配送方式数据，存入列表 nums，代码略。

① n=len(nums)

```

for i in range(n):
    if nums[i] == 0:
        nums[i] = -1

```

d = {}

cur = ans = 0

```

for i in range(n):
    cur += nums[i]
    if cur==0 ②:

```

**cur存放每次的前缀和，cur为0时配送和自提正好相等**

```

    ans = i+1
    if cur not in d:
        d[cur] = i
    else:
        tmp = i-d[cur] ③

```

**相同的前缀和在之前已经出现过，第二次出现说明两段之间的累计和为0，可以统计长度**

```

    if tmp > ans:
        ans = tmp

```

print(ans)

14. 某商场搭建了智能客流监测系统，通过摄像头传感器记录每小时的客流数据（单位：人），由智能终端汇总后经 5G 网络上传至服务器数据库。服务器实时分析客流高峰时段，当每小时客流超过阈值时，自动推送客流高峰提醒至商场管理人员；用户可通过公众号输入日期查询当日客流趋势及各楼层分布图表。请回答下列问题：

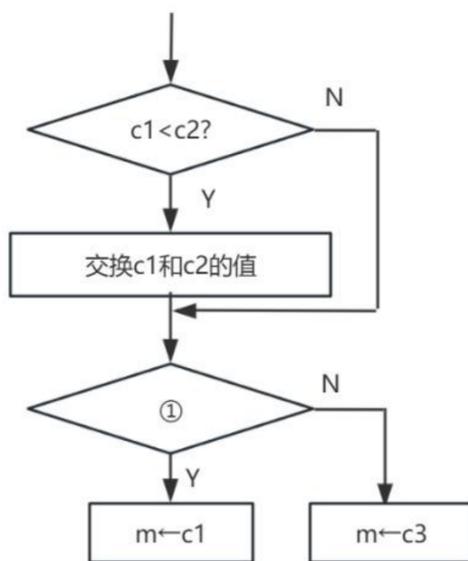
(1) 该系统中，摄像头、智能终端、服务器与公众号之间数据的传输关系 C ▲ (单选)

- A. 摄像头 → 智能终端 → 服务器 → 公众号
- B. 摄像头 → 智能终端 ↔ 服务器 → 公众号
- C. 摄像头 → 智能终端 → 服务器 ↔ 公众号

(2) 下列关于该系统功能的说法, 正确的是 AC (多选)

- A. 智能终端负责客流数据的采集和初步汇总
- B. 服务器仅用于存储数据, 不参与数据分析
- C. 公众号查询历史客流数据需要访问数据库
- D. 智能终端向管理人员推送客流高峰提醒

(3) 智能终端每小时采集 3 次客流数据 ( $c_1$ 、 $c_2$ 、 $c_3$ ), 以最大值作为当前客流量判断是否超过阈值, 流程图如图所示, 图中①处应填入  $c_1 > c_3$ ?



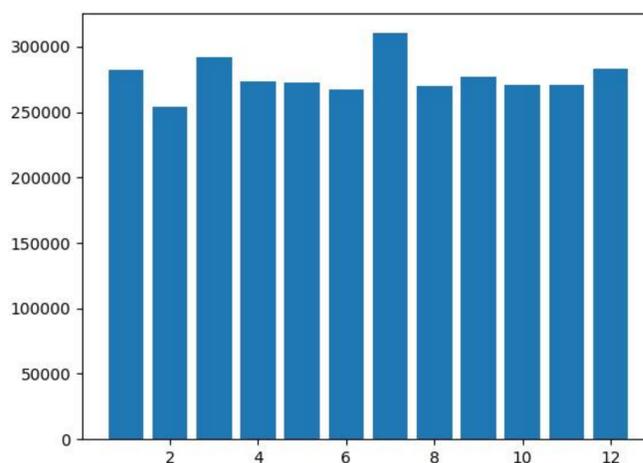
参考25年6月真题答题格式:  
 在服务器端根据实时客流量向通风设备发送控制指令;  
 在智能终端增加(烧录)接受服务器端控制指令的代码;  
 在智能终端增加(烧录)根据客流量(阈值)控制通风设备的代码;

(4) 现需增加自动通风设备, 根据客流量自动调整, 在智能终端接入通风设备后, 还需对软件部分作多处修改。请用文字描述其中 1 处修改建议。

(5) 将三层商场数据导出, 存于 flow\_data.xlsx 文件中, 如图 a 所示。现要统计该商场每月客流量变化, 并绘制如图 b 所示的柱形图。实现上述功能的部分 Python 程序如下, 划线处应填入的语句为 月。

1	楼层	月	日	客流
2	一	1	1	3660
3	一	1	2	3939
4	一	1	3	4981
5	一	1	4	3691
1091	三	12	26	2299
1092	三	12	27	4270
1093	三	12	28	4547
1094	三	12	29	2678
1095	三	12	30	3413
1096	三	12	31	4719

第 14 题图 a



第 14 题图 b

```

import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel("flow_data.xlsx")
g = df.groupby(月, as_index = False).客流.sum()
# 设置绘图参数, 显示如图第 14 题图 b 所示的线形图, 代码略

```

(6) 统计每个楼层每月的客流总量, 找出客流最大的楼层及相应的月份。

```
df["楼层"] = df["楼层"] + df["月"].astype(str) # astype(str)将数字列转换为字符
```

```
print("第" + g[0] + "层在" + g[1:] + "月份客流最大")
```

方框中应填入的语句依次为 1、3、5 (限选 3 项)

- ① `g = df.groupby("楼层", as_index = False).客流.sum()`
- ② `g = df.groupby("月", as_index = False).客流.sum()`
- ③ `g = g.sort_values("客流", ascending = False, ignore_index = True)`

- ④g = g.sort\_values("客流", ascending = True, ignore\_index = True)
- ⑤g = g.at[0, "楼层"]
- ⑥g = g.at[0, "月"]

15. 某学校设置一间答疑室，现要开发一个答疑室预约系统，学生可在晚自修前提交预约申请，在晚自修期间根据系统分配的答疑时间进行答疑。答疑室预约系统中，共开放  $r$  个可预约时间段，每个时间段最多容纳  $c$  名学生，学生预约申请信息包含：学生学号、预约时间段 ( $0 \sim r-1$ )、所需答疑时长 (1 或 2 个连续时段)；若预约时间段人数未满，成功预约；若预约时间段人数已满，进入等待分配队列；若学生预约 2 个连续时段，需两个时段人数均未满才可成功预约，否则进入等待分配队列；第一轮分配完后，在满足学生所需答疑时长的要求下，从等待分配队列中按先进先出原则、按时间段从早到晚分配到未满的时间段中。若  $r=3, c=2$ ，学生申请和分配情况如第 15 题图所示。

预约申请	时间段 0	时间段 1	时间段 2	申请结果	分配结果
["stu1",1,1]		√		第一轮成功	1
["stu2",0,2]	√	√		第一轮成功	0,1
["stu3",1,2]				第一轮等待，第二轮失败	-1
["stu4",1,1]			√	第一轮等待，第二轮成功	2
["stu5",0,1]	√			第一轮成功	0

第 15 题图

- (1) 若在第 15 题图的预约申请中，新增一条预约申请 ["stu6", 1, 1]，stu6 的分配结果为 2 ▲。
- (2) 定义如下 sort(result) 函数，参数 result 中每个元素由学号、分配结果组成，根据学号对 result 数据进行升序排序。请在程序划线处填上合适的代码。

```
def sort(result):
    n = len(result)
    for i in range(1,n):
        k = result[i]
        j = i - 1
        while j >= 0 and int(k[0][3:]) < int(result[j][0][3:]):
            result[j+1]=result[j]
            j -= 1
        result[j+1]=k
```

插入排序，根据学号  
进行升序排序

边比边移

(3) 答疑室预约功能的 Python 程序如下，请在划线处填入合适的代码。

```
def proc(data, r, c):
    n = len(data); a = [0] * r
    h = t = -1
    for i in range(n):
        data[i].append(i+1)
    data[n-1][-1] = -1
    for i in range(n):
        k, m = data[i][1], data[i][2]
        for j in range(k, k+m):
            if a[j] >= c:
                break
            if a[j] < c:
                pre=i
                a[k] += 1
                a[k+m-1] += m-1
                data[i][3] = k
```

data 中的每个元素增加指针域 (data 中原有 4 个值: 学号、预约时段、答疑时长、申请结果)

k 是预约时段, m 是答疑时长 1 或 2

检查每个预约信息, 从 k 时段开始检测 m 长度, 查看答疑室内人数是否少于 c

前面 for 循环停止时 a[j] < c, 表示 m 个时段都可以安排

#更新各时间段人数

```

else:
    data[pre][①-1]=data[i][-1]    删i节点
    if h == -1:
        h = i
    else:
        data[t][-1] = i
        t = i
for i in range(r):
    p=q=h
    while a[i]<c and q!=-1:
        if data[q][2]==1 or ② i+1<r and a[i+1]<c:
            data[pre][-1]=q
            data[q][3]=i
            a[i]+=1
            a[i+data[q][2]-1]+=data[q][2]-1    #更新各时间段人数
            if q==h:
                h=data[h][-1]
            else:
                data[p][-1]=data[q][-1]
            pre=q
        else:
            p=q
            q=data[q][-1]
'''

```

不能安排的i节点，需要去排队，等待第二轮安排,将i节点从已经安排的链中删除，加到h(头)~t(尾)这条链中

第二轮安排，从h开始，如果能安排的就加入到第一轮安排完结束的节点pre之后

此处有个隐藏条件就是 data[q][2]==2

→将q节点从未安排的链表中删除

当前if成立，q节点可以被安排，则更新已经被安排好的链表尾节点pre

参数 data 列表存放已按提交顺序排列的申请信息，每个元素包含 4 个数据项，依次为学号、预约时间段、所需答疑时长（1 或 2）、分配结果。参数 r 存放时间段数。参数 c 存放最多容纳学生数。

```

proc(data, r, c)
#将成功分配的学生学号和分配结果数据按照学号升序存入 result 中，代码略
sort(result)
print(result)

```