

高三上技术选考练习卷 4

第一部分：信息技术（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 1-4 题：

某智慧物流仓储信息系统工作流程如下：货车驶入仓库时 RFID 读写器自动识别货物电子标签，完成信息录入并上传服务器存入数据库。自动分拣机器人通过 Wi-Fi 接收服务器指令，将货物运送到指定货架，运送途中机器人会通过摄像头识别并避开障碍物。管理员可通过电脑或手机 App 实时查看库存量数据、监控机器人状态。

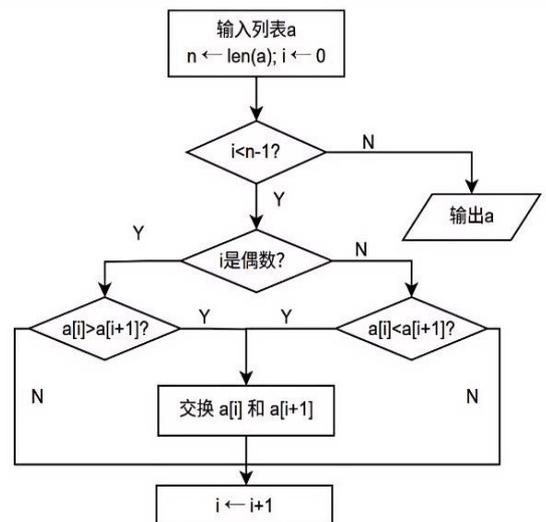
- 下列关于该仓储系统中数据的说法，不正确的是 **D**
 - 扫描货物电子标签可以获取货物批次、数量等信息
 - 货物库存量数据经过分析处理后，可用于预测销售趋势
 - 货物库存量数据在计算机中以二进制编码方式存储
 - 货物相关信息录入系统后，其价值不会随时间变化
- 为保障该智慧仓储信息系统的安全，下列措施中不合理的是 **B**
 - 对数据库中的数据进行异地备份
 - 管理员为方便管理，把自己的生日设置为 App 登录密码
 - 在系统局域网与因特网之间设置防火墙
 - 对不同层级的管理员设置不同的数据访问权限
- 下列关于该系统组成的说法，正确的是 **A**
 - 自动分拣机器人中包含中央处理器
 - 管理员手机上安装的该系统 App 属于系统软件
 - 机器人的摄像头不属于系统的传感器
 - 该系统中的硬件仅包括机器人和服务器
- 该系统的下列应用中，主要体现人工智能技术的是 **C**
 - RFID 读写器自动录入货物信息 **传感技术**
 - 该系统手机 App 自动获取机器人状态 **计算机技术**
 - 机器人通过图像识别避开障碍物 **人工智能技术**
 - 机器人通过 Wi-Fi 与服务器通信 **通信技术**

阅读下列材料，回答第 5-6 题：**人工智能技术（联结主义）**

某在线教育平台为学生提供在线课程视频。学生通过浏览器观看视频时，系统会记录学生的学习时长，暂停、快进等操作数据，并存储在服务器中。

- 下列关于该在线教育平台，说法正确的是 **D**
 - 平台升级时学生需要更新浏览器
 - 浏览器与服务器之间的数据传输是单向的 **双向**
 - 课程视频只能通过计算机网络传输
 - 课程视频通过网络实现了数据共享
- 有一段长 10 分钟的未压缩课程视频（不考虑音频），每秒 25 帧，每一帧图像的像素数量为 1024*768，颜色位深度为 16 位。压缩后文件大小为 250MB，则压缩比约为 **B**
 - 20:1
 - 90:1
 - 180:1
 - 720:1

- 某算法的部分流程图如图所示。执行这部分流程，若输入的 a 列表为 a=[3,1,4,2,5,7,6]，则输出 a 列表的值是 **B**
 - [3,1,4,2,5,7,6]
 - [1,4,2,5,3,7,6]
 - [1,2,3,4,5,6,7]
 - [1,3,2,4,5,7,6]



第 7 题图

i	3,1,4,2,5,7,6
0 偶数	1,3,4,2,5,7,6
1 奇数	1,4,3,2,5,7,6
2 偶数	1,4,2,3,5,7,6
3 奇数	1,4,2,5,3,7,6
4 偶数	1,4,2,5,3,7,6

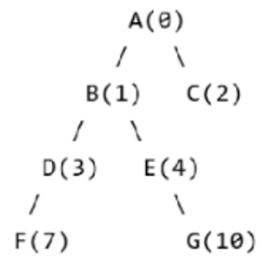
- 下列有关数据结构的说法正确的是 **C**
 - 常见的线性关系数据结构有数组、队列、栈、~~树~~
 - 栈是一种操作受限的线性表，具备“先进先出”的特点
 - 链表在频繁插入和删除元素时，算法效率比数组高
 - 二叉树结构中，每个节点的父节点最多可以有 2 个

每个节点（除根节点外）只有一个父节点

9. 某二叉树用一维数组表示，如图所示。

下标	0	1	2	3	4	5	6	7	8	9	10
元素	A	B	C		D	E				F	G

第9题图



则该二叉树的后序遍历结果为 **A**

- A. FGDBECA B. FGDEBCA C. FDBGCEA D. FDGBECA

10. 有如下 Python 程序段：

```
def fun(s1, s2):
    p = 0
    for i in range(len(s1)):
        if s1[i] == s2[p]:
            p += 1
        if p == len(s2):
            return True
    return False
```

寻找s1中与s2相同的字符个数，相对位置也要相同，若s2的字符都在s1中出现且相对位置相同，则返回True

s1 和 s2 分别取下列值时，调用函数得到的结果与其他三项不同的是 **B**

- A. "12345", "135" **T** B. "36425", "346" **F** C. "98765", "876" **T** D. "24680", "246" **T**

11. 有如下 Python 程序段：

```
def b_search(a, key, left, right):
    while left <= right:
        mid = (left + right)//2
        if a[mid] > key:
            right = mid-1
        else:
            left = mid+1
    return ① left
```

二分查找，找插入key的位置
返回插入的位置99%都是left(不管相等时往前找还是往后找)

```
for i in range(1, len(a)):
    key = a[i]
    pos = b_search(a, key, 0, i-1)
    for j in range(②):
        a[j+1]=a[j]
    a[pos] = key
```

插入排序，将1~len(a)-1位置的值逐个用二分查找的方式找到在0~i-1范围中插入的位置，然后用for循环实现从i-1~pos位置的值批量后移

为实现数组 a 升序排序，则程序划线①②处应填入的语句为 **C**

- A. ①left ②i-1, pos, -1 B. ①right ②i-1, pos, -1 C. ①left ②i-1, pos-1, -1 D. ①right ②i-1, pos-1, -1

12. 有如下 Python 程序段：

```
stks, cnt = [], [0]*5
for i in range(len(a)):
    if i%3 != 2:
        i=0, 1, 3, 4, 6, 7
        if cnt[a[i]] == len(stks):
            stks.append(a[i])
        else:
            cnt中的值是栈的索引号
            stks[cnt[a[i]].append(a[i])
            cnt[a[i]] += 1 投桶
    else:
        i=2, 5, 8
        val = stks[-1].pop()
        if not stks[-1]:
            stks.pop()
        cnt[val] -= 1
```

按规则将元素分组入栈，每3个元素出栈一次

若列表 a 为 [2, 2, 0, 3, 1, 4, 4, 1, 3]，程序运行结束时 stks[0] 不包含 **A**

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

i	stks=[]	cnt=[0,0,0,0,0]
i=0 入	stks=[[2]]	cnt[2]=1
i=1 入	stks=[[2],[2]]	cnt[2]=2
i=2 出	stks=[[2]]	cnt[2]=1
i=3 入	stks=[[2,3]]	cnt[3]=1
i=4 入	stks=[[2,3,1]]	cnt[1]=1
i=5 出	stks=[[2,3]]	cnt[1]=0
i=6 入	stks=[[2,3,4]]	cnt[4]=1
i=7 入	stks=[[2,3,4,1]]	cnt[1]=1
i=8 出	stks=[[2,3,4]]	cnt[1]=0

操作类型	具体规则
入栈	对数字 x=a[i]: ① 若 cnt[x] == len(stks) (无可用子栈) - 新建子栈 [x] 并加入 stks; ② 否则 - 把 x 追加到 stks[cnt[x]] 子栈末尾; ③ cnt[x] += 1 (更新计数)。
出栈	① 从 stks 最后一个子栈弹出末尾元素 val; ② 若弹出后子栈为空 - 删除该空栈; ③ cnt[val] -= 1 (回退计数)。

二、非选择题（本大题共3小题，其中第13题8分，第14题9分，第15题9分，共26分）

13. 小张设计了一个股票价格模拟分析算法来监测并提醒买入信号.给定股票价格序列，首先识别该序列的所有波峰（局部最大值）和波谷（局部最小值），例如：当 $a[i]$ 分别大于 $a[i-1]$ 和 $a[i+1]$ ，则 $a[i]$ 为波峰，反之当 $a[i]$ 分别小于 $a[i-1]$ 和 $a[i+1]$ ，则 $a[i]$ 为波谷。如果出现以下任何一个条件，则触发提醒买入信号：

条件一：出现连续3个递增波峰，第3个波峰触发买入信号。

条件二：出现大于第一个波峰的波谷，该波谷触发买入信号。

请回答下列问题：**3个波峰不是递增100,104,103**

(1) 若股票价格序列为[95, 100, 98, 92, 95, 102, 104, 100, 103, 98]，**B**▲____（单选，填字母：A.是/B.否）会触发买入信号。

(2) 实现买入信号检测功能的 Python 程序如下，请在划线处填入合适的代码。

输入当前股票价格序列 prices，代码略

```
n = len(prices)
```

```
peaks, valleys = [], []
```

```
for i in range(1, n-1):
```

```
    pre = prices[i-1]
```

```
    cur = prices[i]
```

```
    nex = prices[i+1]
```

```
    if cur > pre and cur > nex 或 pre < cur < nex:
```

```
        peaks.append([i, cur])
```

```
    elif cur < pre and cur < nex:
```

```
        valleys.append([i, cur])
```

```
buy_signal = False
```

```
pos = 0
```

```
if len(peaks) >= 3: 2, len(peaks)
```

```
    for i in range(②):
```

```
        if peaks[i-2][1] < peaks[i-1][1] < peaks[i][1]:
```

```
            buy_signal = True
```

```
            pos = i
```

```
            break
```

```
if not buy_signal and peaks and valleys: # 若条件1未触发，且波峰波谷存在
```

```
    first_peak_price = peaks[0][1]
```

```
    for i in range(len(valleys)):
```

```
        if valleys[i][1] > first_peak_price:
```

```
            buy_signal = True
```

```
            pos = i
```

```
            break
```

输出是否触发买入信号，代码略

14. 某校建设智慧节能监测系统，实现对多幢教学楼用电量实时监测。每幢楼的智能电表每小时采集一次用电量数据，每幢楼的智能终端内置5G通信模块，通过移动通信网络发送数据到服务器进行处理，若服务器监测到某教学楼用电量超过阈值，通过智能终端控制声光报警器发出警报，处理后存储到数据库。管理员可通过浏览器查看实时和历史用电量数据。请回答下列问题：

(1) 智能终端将用电量数据上传到服务器和浏览器访问 Web 服务器的过程可以使用同一种应用程序协议 **A** ▲____（单选，填字母：A.正确 / B.错误）。 **http协议**

(2) 现希望用电异常发出警报时，管理员检查完毕后能够一键关闭警报，以下位置适合设置按键的是 **A** ▲____（单选，填字母：A.浏览器页面中 / B.声光报警器上 / C.智能电表上）。

(3) 若数据库突发故障不能工作，会引发的现象有 **AD** ▲____（多选，填字母）。

备份角度：建立备份服务器（与主服务器不在同一网络）或备份网络（与主网络独立）

设备更换角度：服务器增加5G模块，通过移动通信网络连接互联网

②功能移植角度：将阈值判断功能移植到各智能终端

③延时上传角度（这个角度并不能解决瘫痪问题）：在智能终端增加数据缓存和备份功能，当网络恢复正常时将数据重新上传。

- A. 管理员无法通过浏览器查看历史数据
- B. 智能电表无法采集用电量数据
- C. 智能终端无法传输数据到服务器
- D. 报警器可以正常工作

智能终端和传感器、执行器之间操作都正常

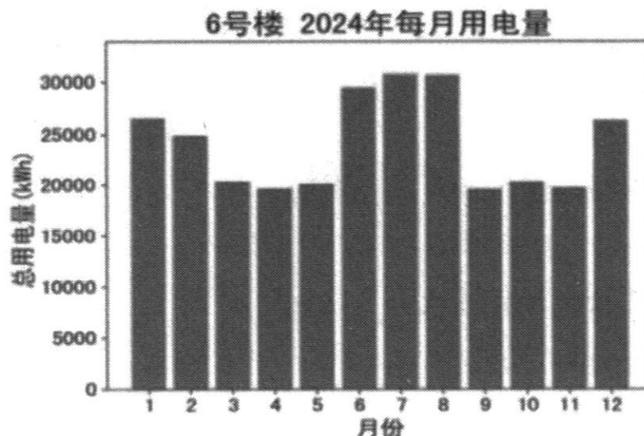
无法存进数据库

(4) 若服务器或服务器所在网络发生故障，所有监测点将无法工作。请针对这个问题，提出一项改进措施，以提高系统的稳定性。

(5) 服务器中存储了2024年全年的用电量详细数据，将其导出到“energy_data.xlsx”文件中，部分数据如第14题图a所示。现需要统计分析2024年3月份的用电数据，找出3月总用电量最高的教学楼，并绘制该教学楼2024年每月用电量的柱形图，如第14题图b所示。

位置	月	日	小时	用电量(kWh)
1号楼	1	1	0	4.54
2号楼	1	1	0	6.72
3号楼	1	1	0	8.34
4号楼	1	1	0	5.41
5号楼	1	1	0	5.8
6号楼	1	1	0	13.18
1号楼	1	1	1	5.94
2号楼	1	1	1	6.95
...

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("energy_data.xlsx")
```

```
df1 = df[df["月"]==3]
```

F ① 按不同楼号进行分组统计每幢楼的总用电量

C ② 按总用电量进行排序，排最上面的就是3月份用电量最多的楼号

```
# 取出 df1 中“位置”列的第一个元素值存入 building，代码略
```

```
print("3月总用电量最高的教学楼是:", building)
```

B ③ 从原数据df中筛选出用电最高的楼的数据

E ④ 按月分类统计该楼每月的用电量总数

```
plt.bar(df1["月"], df1["用电量(kWh)"])
```

```
# 设置绘图参数，显示如图 b 所示的柱形图，代码略
```

程序中①②③④处可选的代码有：

- A. df1 = df1[df1["位置"]==building]
- B. df1 = df[df["位置"]==building]
- C. df1 = df1.sort_values("用电量(kWh)", ascending = False)
- D. df1 = df1.sort_values("用电量(kWh)", ascending = True)
- E. df1 = df1.groupby("月", as_index=False).sum()
- F. df1 = df1.groupby("位置", as_index=False).sum()

15. 某通信系统对一个明文字符串 s 进行加密和解密操作。加密过程包含两个步骤：

步骤一：翻转。将明文 s 中多段连续的区间进行翻转。区间存储在密钥 key1 中，每个元素为[st, ed]，若 st<=ed，表示将 st 到 ed 区间（即：s[st:ed+1]）进行翻转；若 st>ed，表示将 st 到字符串末尾再连接字符串开头到 ed 的完整区间（即：s[st:]+s[:ed+1]）进行翻转。

步骤二：旋转。将步骤一得到结果划分成若干个长度为 L 的连续子串，对每个子串分别循环右移 X 位（包括最后一个长度可能不足 L 的子串），然后重新拼接。密钥 key2 的每个元素表示一次旋转操作（即[L,X]），如果 key2 中有多个元素，则表示进行了多轮旋转操作。例如，明文 s 为“Hello”，密钥 key1 为[[1,4],[3,1]]，

"transform", key1=[[2,5]]翻转区间[2,5](字符a,n,s,f), 结果为f,s,n,a, 变为trfsnaorm
 key2旋转第一轮[4,1]:按长度4分块为[t,r,f,s]、[n,a,o,r]、[m], 各子串循环右移1位后为[s,t,r,f]、
 [r,n,a,o]、[m], 拼得strfrnaom。
 第二轮[8,2]:按长度8分块为[s,t,r,f,r,n,a,o]、[m], 第一个子串循环右移2位后为aostrfrn, 最终密文为
 密文 key2 为[[2,1]], 加密过程如图所示。 aostrfinm

明文 s	Hello
翻转[1,4]后的结果	Holle
翻转[3,1]后的结果	elloH
旋转[2,1]后的密文	leolH

第 15 题图

请回答下列问题:

(1) 若明文 s 为 "transform", 密钥 key1 为 [[2,5]], 密钥 key2 为 [[4,1],[8,2]], 请问加密后得到的密文第一个字符为 **a**。

(2) 有如下 get_length()和 rev()两个函数, 其中 rev()函数加框处的语句有误, 请改正。

get_length()函数求长度为 n 的列表中下标 st 到 ed 区间的长度(含 st 和 ed)

```
def get_length(n, st, ed):
    return (ed-st+n)%n + 1
```

rev()函数将列表 data 中从下标 st 到 ed(含 st 和 ed)的区间进行翻转

```
def rev(data, st, ed):
```

```
    n = len(data)
```

```
    t = get_length(n, st, ed)
```

**只需循环区间长度的一半, 交换前半与后半字符
 get_length(n, st, ed) // 2**

```
    for i in range(t):
```

```
        p = (st+i) % n
```

```
        q = (ed-i) % n
```

```
        data[p], data[q] = data[q], data[p]
```

(3) 实现加密和解密的代码 Python 程序如下, 请在划线处填入合适的代码。

```
def transform1(data, key, mode): #翻转操作
```

```
    if mode == -1: #翻转操作: data为字符列表, key为翻转密钥, mode=1加密/-1解密
```

```
        rev(key, 0, len(key)-1) #解密时先翻转密钥顺序(按密钥逆序执行翻转)
```

```
    for i in range(len(key)):
```

```
        rev(data, key[i][0], key[i][1]) #1:对 data 执行 key[i]的[st,ed]区间翻转
```

```
    return data
```

#循环右移: st/ed为区间首尾, sflen为右移位数

```
def ring_shift(data, st, ed, sflen): #循环移位
```

```
    n = len(data)
```

```
    le = get_length(n, st, ed) # 计算区间长度
```

```
    sflen = sflen % le #处理右移位数超过区间长度的情况
```

```
    if sflen == 0: #右移0位直接返回
```

```
        return data
```

#三次翻转法实现循环右移: 先翻前半段, 再翻后半段, 最后翻整体

```
    rev(data, st, (st+le-sflen-1)%n)
```

```
    rev(data, (st+le-sflen)%n, ed)
```

```
    rev(data, st, ed)
```

```
    return data
```

以 "子串[a,b,c,d,e]右移 2 位" 为例:

目标: [d,e,a,b,c];

步骤 1: 翻转左半段 (前 5-2=3 个): [c,b,a,d,e];

步骤 2: 翻转右半段 (后 2 个): [c,b,a,e,d];

步骤 3: 翻转整个区间: [d,e,a,b,c] (达成右移 2 位)。

key为旋转密钥，mode=1加密/-1解密

```
def transform2(data, key, mode): #旋转操作
    n = len(data)
    if mode == -1: #解密时先翻转密钥顺序(逆序执行旋转)
        rev(key, 0, len(key)-1)
    for i in range(len(key)):
        #按L划分区间:步长为key[i][0](即L)
        for st in range(0, n, key[i][0]):
            ed = min(st+key[i][0]-1, n-1) #计算区间结束下标(处理最后一段不足L的情况)
            le = ed-st+1 #当前子串长度
            if mode == 1: #加密:直接取右移位数x
                shift = key[i][1]
            else: #解密:左移x位等价于右移le-x位
                shift = le-key[i][1]
            data = ring_shift(data, st, ed, shift) #或 shift=-key[i][1]
    return data
# 输入原始明文存储在字符串 s 中; 读取密钥存入 key1、key2 列表。
# key1 每个元素包含 2 个数据项, 依次为待翻转区间的起始和结束位置;
# key2 每个元素包含 2 个数据项, 依次为划分长度和循环右移位数, 代码略。
s_enc = transform2(transform1(list(s), key1, 1), key2, 1)
# 注: list(s)把字符串转换为字符列表
s_dec= transform1(transform2(s_enc, key2, -1), key1, -1)
# 输出密文 s_enc 和解密后的明文 s_dec,代码略。
```

加密 (mode=1) : 按 key1 顺序翻转 → 按 key2 顺序旋转;
解密 (mode=-1) : 先按 key2 逆序旋转 (反向执行密钥 + 移位取反)
→ 再按 key1 逆序翻转。