

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

2026 年 1 月浙江省普通高校招生选考科目考试

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 5 题。

某工地采用智能环保喷淋系统，该系统在工地围栏上安装了粉尘传感器，实时采集粉尘浓度数据，数据通过 4G/5G 通信网络传输至服务器，系统根据数据处理结果启停喷淋设备或调整喷水量。当工地出口处的摄像头识别出渣土车时，系统会启动车辆冲洗功能。若监测到故障（如传感器接线故障），服务器会发送短信通知维护人员。管理员可通过移动终端 APP 查看设备运行状况，也可远程控制喷淋设备。此外，系统能根据需要生成运行统计报表和故障记录表。

1. 下列关于该系统中数据的说法，正确的是 **B**

A. 系统中数据的表现形式是单一的

C. APP 中的数据都是未经加工处理的

有数值、文本等

D. 统计报表的分析结果可用于优化系统 ✓

D. 调整喷水量的过程中不会产生数据 **会产生新的数据**
2. 下列关于该系统组成的描述，正确的是 **B**

A. 粉尘传感器的性能对系统运行无影响

C. APP 的运行无需操作系统的支持

摄像头采集图像数据，属于输入设备

B. 摄像头是该系统的输入设备

D. 维护人员不属于该系统的用户
3. 下列关于该系统功能与应用的描述，正确的是 **D**

A. 该系统对外部环境没有依赖

C. 该系统没有数据输出功能

B. 服务器不对故障信息进行处理

D. 识别渣土车可采用人工智能技术
4. 下列关于该系统网络技术的分析，正确的是 **A**

A. 利用 APP 查看故障记录表是网络资源共享的体现

C. 系统中的数据传输均需采用移动通信技术

网络共享：1硬件资源 2软件资源 3数据资源

B. 远程控制喷淋设备指令的发送无需遵循网络协议 **需要协议：如 TCP/IP**

D. 服务器必须部署在工地的局域网中 **也可以采用计算机网络 可以在云端或局域网中**
5. 下列关于该系统安全与防护的做法，不合理的是 **A**

不同

A. 对所有用户设置相同的数据访问控制权限

B. 对故障记录表等重要数据定期备份

C. 对传感器和网络设备定期检查

D. 对系统漏洞及时修复

6. 某 256 色、BMP 格式的位图图像由扫描照片获得。下列关于该图像的说法不正确的是 **B**

- A. 扫描照片是图像数字化的过程

C. 图像存储容量与像素数量有关

D. 该图像可以存储为 JPEG 格式
- 256色需要8位存储**

B. 每个像素至少用 32 位二进制数表示

7. 某算法的部分流程图如第 7 题图所示，若 n 的值为 8 数组元素 a[0]至 a[n-1]依次存放 2,3,4,8,5,4,9,6，执行这部分流程后，输出 c 的值为 **D** **升序冒泡排序单轮循环，c 记录最后一次交换的位置**

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

8. 队列初始为空，字符“甲”“乙”“丙”“丁”“戊”“己”依次入队的规则为：若队列为空，字符入队，否则队首元素出队后再入队，然后字符入队，重复执行直到字符全部入队。字符全部入队后，队首元素为 **D**

- A. 丁 B. 己 C. 甲 D. 乙

9. 某二叉树有 a、b、c、d 四个节点，若中序遍历序列为 abcd，后序遍历序列为 dcba，则该二叉树的树形结构为 **C**

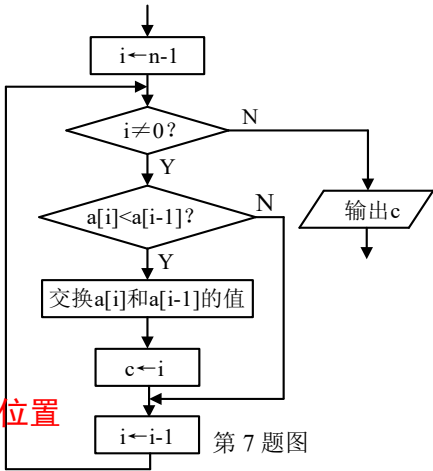
- A.

B.

C.

D.

中序遍历和后续遍历逆序



加速：可用ABCDEF代替

10. 有如下 Python 程序段: 将s字符串中非"e"字符累加, 碰到第t+1个"e" 结束

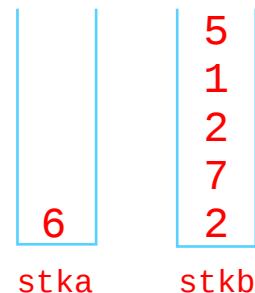
```
c, i = "", 0 s="keeper" t=2
while i < len(s) and t >= 0:
    if s[i] == "e":
        t -= 1
    else:
        c += s[i]
    i += 1
```

若 s 为"keeper", t 为 2, 执行该程序段后, c 的值为 **B**

- A. "k" B. "kp" C. "kpr" D. "kper"

11. 有如下 Python 程序段:

```
s = 0
while topa != -1: stka栈顶元素> stkb栈顶元素
    while topb != -1 and stka[topa] > stkb[topb]:
        if (stka[topa] + stkb[topb]) % 2 == 1: 栈顶元素奇偶不同
            topa += 1
            stka[topa] = stkb[topb] stkb出栈入stka
            topb -= 1
        if topb != -1: stkb出栈1次
            topb -= 1
    s += stka[topa] stka栈顶元素累加, 再出栈
    topa -= 1
```



若 stka 为[6,0,0,0,0], topa 为 0, stkb 为[2,7,2,1,5], topb 为 4, 执行该程序段后, s 的值为 **C**

- A. 13 B. 14 C. 15 D. 16 s: 2+2+5+6

12. 有如下 Python 程序段: j从1开始, 寻找小于a[0]的元素, 并与a[i]交换, 交换后i后移1个位置

```
i = 1
for j in range(1, len(a)):
    if a[j] < a[0]: a[0]:基准数
        a[i], a[j] = a[j], a[i]
        i += 1
```

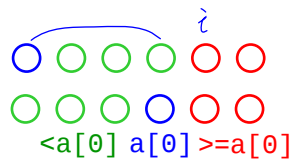
[6, 2, 9, 7, 4, 5, 3]

a[0], a[i-1] = a[i-1], a[0]

执行该程序段后, a 的值为[2, 4, 3, 5, 6, 9, 7], 则下列选项中不可能为 a[0]的初始值的是 **D**

- A. 2 原:[2, 4, 3, 5, 6, 9, 7] B. 5 原:[5, 4, 3, 2, 6, 9, 7] C. 6 原:[6, 4, 3, 5, 2, 9, 7] D. 7 i=7 a[0:i]

快速排序:
作用是选定一个基准元素, 将数组划分为
小于基准和大于等于基准的两部分。
继续... (分治算法)



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

13. 某小组模拟搭建植物大棚温度监控系统, 采用智能终端连接温度传感器、风速传感器, 每分钟采集一次温度和风速数据, 并通过无线通信方式将数据传输至服务器, 存储到数据库中。服务器处理数据后, 通过智能终端控制风扇运行, 用户可通过浏览器查看系统数据。请回答下列问题:

(1) 一个智能终端 **A** (单选)。

- A. 可以同时连接温度和风速传感器 ☒ B. 不可以同时连接温度和风速传感器

(2) 关于该系统中数据处理的说法、正确的有 **BC** (多选)。(注: 全部选对的得 2 分, 选对但不全的得 1 分, 不选或有选错的得 0 分)

- A. 系统中的所有数据均来自传感器 ☐ B. 智能终端可采用蓝牙进行数据传输 ☒
C. 服务器和智能终端均可存储数据 ☒ D. 无法在浏览器中同时呈现温度和风速数据 ☐

(3) 系统运行后, 数据库将越来越庞大, 为减轻服务器的 **存储** 压力, 下列做法有效的有 **AD** (多选)。

- (注: 全部选对的得 2 分, 选对但不全的得 1 分, 不选或有选错的得 0 分)
A. 适当增大数据的采集时间间隔 ☒ B. 增加数据采集点的数量 ☐
C. 提升服务器 CPU 的性能 ☐ D. 定期将数据库中的数据进行异地转存 ☒

能加快处理速度, 不能缓解存储压力

(4) 系统运行后, 为验证温度传感器采集的数据的准确性, 请写出 1 种合理的验证方法。

将原有传感器采集的数据与水银温度计与水银温度实测数据做对比

在同一位置增加采集点, 将两个数据做对比

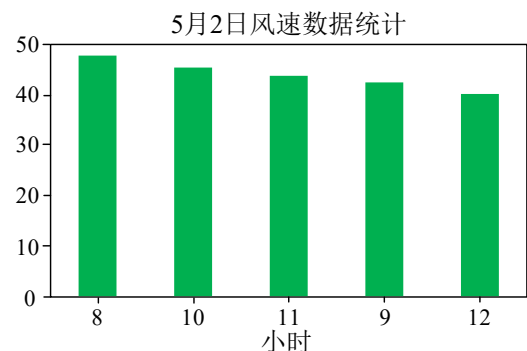
要有对象 方法

不推荐: 改变环境温度, 浏览器查看温度数据是否变化 特定温度环境(恒温箱)

(5) 将当年 5 月份的风速(单位:m/s)数据导出到文件 data.xlsx 中。部分数据如第 13 题图 a 所示。统计 5 月 2 日每小时中风速大于该日平均风速的次数, 选择次数最多的前 5 个小时的数据, 绘制如第 13 题图 b 所示的柱形图。

	A	B	C	D	E
1	月	日	小时	分钟	风速
2	5	1	0	0	1.568
3	5	1	0	1	0.257
4	5	1	0	2	0.405
5	5	1	0	3	1.074
6	5	1	0	4	0.300
7	5	1	0	5	1.184
8	5	1	0	6	1.663
9	5	1	0	7	0.195
10	5	1	0	8	1.770
11	5	1	0	9	0.497
12	5	1	0	10	0.199

第 13 题图 a



第 13 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(单选)。

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
df=pd.read_excel("data.xlsx")
```

```
df1=df[df["日"]==2]
```

```
ave=df1["风速"].mean()
```

```
df1 = ① B.df1[df1["风速"] > ave]
```

```
df2 = ② E.df1.groupby("小时",as_index=False).count() 按照小时统计每个小时的次数
```

```
# 重命名 df2 中"风速"列名称为"次数", 代码略
```

```
df2 = ③ D.df2.sort_values("次数",ascending=False) 按照次数降序排序
```

```
df3 = df2.head(5)
```

```
# 获取前 5 条数据
```

```
#设置绘图参数, 选取 df3 中的数据创建图表, 显示如图 b 所示的柱形图, 代码略
```

```
程序中①②③处可选的代码有:
```

A df[df["风速"] > ave]

B df1[df1["风速"] > ave]

C df2.sort_values("次数",ascending=True) # 升序排序

D df2.sort_values("次数",ascending=False)

E df1.groupby("小时",as_index=False).count() # 分组计数

F df1.groupby("风速",as_index=False).count()

先D后E?

1. 重命名后才有次数列

2. 排序后做groupby, 会重新按小时排序

14 服务器根据温度数据生成状态码, 满足一定条件时发送给智能终端以控制风扇运行。生成和发送状态码的规则如下:

①温度 $\leq 15^{\circ}\text{C}$ 时, 状态码为 0; 温度 $\geq 35^{\circ}\text{C}$ 时, 状态码为 2; 其他温度范围, 状态码为 1。

②当状态码连续 k ($k>1$) 个相同, 且与最近已发送状态码不同时, 则发送该状态码。

请回答下列问题:

(1) 若 k 为 3, 最近已发送状态码为 1, 随后生成的状态码序列为“0, 0, 0, 2, 0, 0, 1, 1, 1, 0”, 则由该序列触发的状态码发送次数为 2 次。

实际场景: 波动的模拟信号如何判断超过阈值

(2) 实现上述功能的部分 Python 程序如下, 请在划线处填入合适代码。

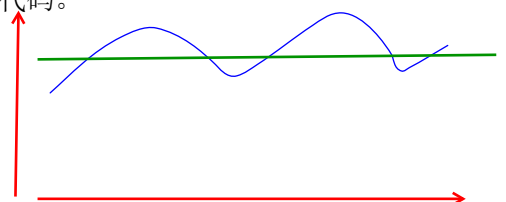
```
sent = last = -1
```

```
# 获取 k 值, 代码略
```

```
while True:
```

```
# 获取温度数据保存到变量 t 中, 代码略
```

```
①
```



```

sent = last = -1
# 获取 k 值，代码略
while True:
    # 获取温度数据保存到变量 t 中，代码略
    ① code=0
    if t >= 35:
        code = 2
    elif t > 15:
        code = 1
    if code != sent:
        if code != last:
            cnt = 1
        else:
            cnt += 1
        if cnt == k:
            # 发送状态码 code，代码略
            sent = code
    last = code
    # 延时 1 分钟，代码略

```

sent: 最后1次发送的状态值

last: 上一次状态值

code: 当前状态值

如果本次代码与sent一样，就不用理会，关注的是当前代码与发送代码不同的情况

cnt重置

cnt+=1

if cnt == k:

发送状态码 code，代码略

sent = code

last = code

延时 1 分钟，代码略

15. 某中转站负责转运物品，每个物品规格相同且有唯一的正整数编号，物品送达中转站的时间先后与编号大小无关，每次运走的物品编号均需连续。

将中转站中的现有物品编号划分为若干序列，连续的编号按升序置于同一序列，不连续的编号置于不同序列。当需要运走 x 个物品时，检查当前时间现有序列中是否存在长度 $\geq x$ 的序列，若存在，则：

① 选择长度最短的序列；若最短序列有多个，则选择起始编号最小的序列。

② 选定序列后，运走从其起始编号开始的 x 个连续编号的物品。

若不存在符合条件的序列，则本次需求取消。请回答下列问题：

- (1) 若物品按第 15 题图所示的时间送达中转站，第一次需要运走的物品个数为 3，时间为 09:15，则该次运走物品的起始编号为 206

时间	08:15	08:20	08:30	08:55	09:00	09:05	09:10	09:25
编号	202	206	203	201	208	207	204	205

第 15 题图

- (2) 定义如下函数 `check(reqs,m)`，用于判断列表 `reqs` 中是否存在下列情况：任意连续 60 分钟内，需要运走的物品数量总和超过 m 。列表 `reqs` 中的每个元素包含两个数据项，依次为需要运走的时间（已转换为分钟数）和物品数量，已按时间升序排列。

def check(reqs,m): 滑窗算法

i = j = 0

s = 0

while j < len(reqs):

if reqs[j][0]-reqs[i][0] < 60:

s += reqs[j][1]

if s > m:

return True

j += 1

else:

s -= reqs[i][1]

i += 1

return False

调用该函数，请回答①和②两个问题。

① 若 `reqs` 为 `[[6,1], [19,13], [35,3], [70,2], [75,7]]`， m 为 20，则返回值为 True

② 若 `reqs` 的长度为 100，函数返回值为 False，则 `while` 语句中循环体的执行次数最多是 199。

- (3) 中转站需逐条处理需求信息，根据物品的送达时间和编号，以及需要运走的时间和数量，记录成功运走的物品编号。模拟上述过程的 Python 程序如下，请在划线处填入合适的代码。



小细节：这个函数中有60，后面代码以及总要求中没有60的体现，说明此函数与后面代码无关

序列1 序列2
201 206
202 207
203 208
204
共3个
共4个

循环执行次数最多：每个时间点差距 ≥ 60 以3个为例：

$m=13+3+2+7=25 > 20$

```
def addpid(pid, lst):
```

```
"""
```

该函数根据新加入的物品编号 pid 更新链表。

列表 lst 模拟链表结构，链表节点的前两个数据项依次为连续编号序列的起始和终止编号，第三个数据项为指针。lst[0]为链表的头节点（前两个数据项不存储有效数据），其余节点在链表中按起始编号升序排列，连续编号序列存于同一节点。

例如：若 pid 为 19，lst 为[[-1,-1,2],[15,18,3],[11,13,1],[20,22,-1]]，调用 addpid(pid,lst) 后，lst 为[[-1,-1,2],[15,22,-1],[11,13,1],[20,22,-1]]

```
"""
```

```
# 代码略
```

```
def proc(reqs,items):
```

```
res = []
```

```
lst = [[-1,-1,-1]]
```

初始化为仅包含头节点的链表

```
idx = 0 idx (index索引)
```

```
for i in range(len(reqs)): 遍历需求列表reqs, 可视为reqs只有1个, 减少循环
```

```
cur,need = reqs[i][0], reqs[i][1] 需要运走的时间 数量
```

```
while idx<len(items) and items[idx][0]<cur: 物品在规定时间内
```

```
pid = items[idx][1] 物品编号
```

```
addpid(pid,lst) 添加物品, 并维护lst # 根据新加入的物品编号 pid 更新链表
```

```
① idx+=1
```

```
bestnum = len(items) + 1 假定一个最大值
```

```
bestpre,pre = -1,0 与lst有关
```

idx

```
p = lst[0][2]
```

lst 为[[-1,-1,2],[15,18,3],[11,13,1],[20,22,-1]]

```
while p != -1 and need != bestnum:
```

bestpre pre p

```
num = lst[p][1] - lst[p][0] + 1 p节点的物品总数
```

```
if ② need<=num<bestnum 或need<=num and num<bestnum
```

```
bestnum = num 更新
```

错误: and bestpre<pre?
物品编号是乱序

```
bestpre = pre bestpre就是最终确定的序列的前一个
```

```
pre = p
```

```
p = lst[p][2]
```

```
if bestpre != -1: bestpre 不是-1, 说明已经找到合适的序列
```

1. 物品总数要满足need
2. 物品总数要小于bestnum
(找最短序列)

```
p = lst[bestpre][2] p再次定位到目标节点
```

```
res.append([i,lst[p][0],lst[p][0] + need-1]) # 为 res 追加一个元素
```

```
if bestnum == need: 该序列物品全要-删除节点
```

```
lst[bestpre][2] = lst[p][2]
```

lst 为[[-1,-1,2],[15,18,3],[11,13,1],[20,22,-1]]

```
else: 部分要, 更新lst[p]中起始序号
```

```
③ lst[p][0]+=need
```

p bestpre

```
else:
```

```
# 取消本次需求, 代码略
```

```
return res
```

```
"""
```

将物品信息存入 items 列表，每个元素包含两个数据项，依次为物品送达时间和编号；

将需求信息存入 reqs 列表，每个元素包含两个数据项，依次为需要运走的时间和数量；

两个列表均已按时间**升序**排列。**相同条件下最先找到的肯定是起始编号较小的序列**

代码略

没有按编号升序 ①选择**长度最短**的序列；若最短序列有多个，则选择起始编号最小的序列。

```
"""
```

```
res = proc(reqs,items)
```

```
# 输出 res 列表中的处理结果, 代码略
```

思考：

1. 题干：取的物品需要连续编号 → 数据结构为[开始编号, 结束编号, 指针域]...

2. 不能一次性处理所有的items，物品陆续添加+部分取特点，数据结构变化较大

3. 两个时间线： 1. reqs 取物品时间线 2. items 逐步到达

4. 谁做外循环更优？