

温州市普通高中 2026 届高三第二次适应性考试

技术试题卷

2026. 4

考生须知:

1. 本试卷分两部分, 第一部分信息技术, 第二部分通用技术。全卷共12页, 第一部分1至6页, 第二部分7至12页。满分100分, 考试时间90分钟。

2. 答题前, 请务必将自己的姓名、准考证号用黑色字迹的签字笔或钢笔填写在试题卷和答题卷规定的位置上。

3. 答题时, 请按照答题卷上“注意事项”的要求, 在答题卷相应的位置上规范作答, 在本试题卷上的作答一律无效。

4. 非选择题的答案须用黑色字迹的签字笔或钢笔写在答题卷上相应区域内, 作图时可先使用 2B 铅笔, 确定后须用黑色字迹的签字笔或钢笔描黑。

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、错选、多选均不得分)

阅读以下材料, 回答第 1-6 题

某云上园博系统借助摄像头采集、无人机航拍和大模型技术生成种植点位规划。管理员打开云上园博 APP, 在种植登记页面输入相关信息, 上传苗木照片至服务器端, 系统生成该苗木二维码。养护人员扫码后, 可在手机上显示苗木信息。此外, 园内配备了重载无人机用于运输, 用户可实时查询飞行轨迹等数据。

- 下列关于该系统中数据的说法, 正确的是 **C**
 - 该系统中所有数据都属于结构化数据
 - 服务器中的苗木照片是模拟信号 **数字信号**
 - 重载无人机飞行过程中会产生新数据
 - 苗木全部信息存储在二维码中
- 下列关于该系统组成的说法, 正确的是 **A**
 - 养护人员的手机属于该系统的终端设备
 - 该系统的用户为管理员 **不完整**
 - 该云上园博 APP 属于系统软件 **应用软件**
 - 摄像头的画面采集功能不需要软件支持
- 下列关于该系统功能与应用的说法, 正确的是 **C**
 - 系统无法对多棵苗木的信息进行汇总分析
 - 种植点位规划前无需查询历史苗木种植信息
 - 借助大模型技术生成种植点位规划使用了人工智能技术
 - 无人机的飞行数据仅在飞行结束后统一上传至服务器 **实时查询**
- 下列关于该系统网络技术的说法, 正确的是 **A**
 - 无人机飞行轨迹的实时传输依赖于无线网络技术
 - 养护人员仅通过移动通信网络扫码查看苗木信息
 - 摄像头传输数据时无需遵循网络协议 **TCP/IP**
 - 该系统不能通过互联网实现数据资源共享

5. 下列关于该系统安全与防护的做法, 不合理的是 **B**

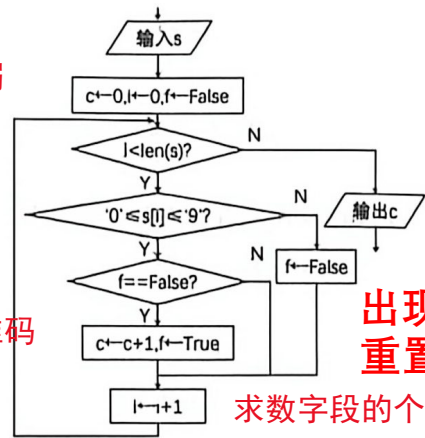
- A. 定期对服务器数据进行备份
- B. 将系统登录密码设置为手机号后六位 **密码强度太弱**
- C. 为不同角色设置不同的访问权限
- D. 对系统中的重要数据进行加密传输与存储

6. 关于系统中图像处理的做法, 合适的是 **D**

- A. 将苗木照片存储为黑白图像
- B. 生成二维码时删除原始苗木照片
- C. 通过图像数字化技术生成二维码 **编码技术生产二维码**
- D. 采用 JPEG 格式存储苗木照片

7. 某算法的部分流程图如第 7 题图所示, 若 s 的值为 “a12345bcd6e78f”, 则执行这部分流程图后, c 的值为 **A**

- A. 3
- B. 4
- C. 5
- D. 7



第 7 题图

出现字母, f 重置

求数字段的个数

8. 有两个队列 P 和 Q, 进行如下操作: 将两个队列的队首元素出队, 将较大元素入队至较小元素原来所在的队列。重复操作直到其中一个队列为空。初始时队列 P 元素为 1, 6, 2, 队列 Q 元素为 5, 3, 4, 操作结束后, 非空队列及其元素为 **B**

- A. P, 元素为 6, 4
- B. P, 元素为 4, 6
- C. Q, 元素为 6, 5, 4
- D. Q, 元素为 4, 5, 6



9. 某非完全二叉树有 6 个节点, 删除某个节点后的新二叉树为完全二叉树, 则该非完全二叉树形态个数为 **C**

- A. 3
- B. 4
- C. 5
- D. 6

修正: 叶子节点

10. 有如下 Python 程序段:

```

st = i = 0
while i < len(d) - 1:
    if d[i] >= d[i + 1]:
        if i - st + 1 > 2:
            st = i + 1
        i += 1

```

若 d = [1, 3, 5, 5, 2, 4, 4, 6], 执行该程序段后, st 的值为 **C**

- A. 3
- B. 5
- C. 6
- D. 7

11. 有如下 Python 程序段:

```

c = [0, 0]
for x in q:
    c[x] += 1
i = 0
while i < len(st) and c[st[i]] > 0:
    c[st[i]] -= 1
    i += 1
ans = len(st) - i

```

若 q = [0, 1, 1, 0, 0, 1], st = [1, 0, 1, 1, 1, 0], 则执行程序后, ans 的值为 **B**

- A. 3
- B. 2
- C. 1
- D. 0

12. 有如下 Python 程序:

输入 m, 代码略

i = m; j = 0 # j 从 0 开始遍历, i 从 m 开始向后扫描

while j < i and i < len(a): # 确保 j 在 i 左侧且索引合法。

```

    if a[j] > a[i]: # 若 a[j] > a[i]
        t = a[i]
        for k in range(i, j, -1): # 将 a[i] 插入到 j 位置
            a[k] = a[k - 1] # 右侧元素依次后移
        a[j] = t
        i += 1 # 然后 i += 1
        j += 1 # 若不满足, 仅 j += 1。

```

m=2, x=5

初始数组: [1, 5, 5, 4, 6, 7] (a[2]=5)

j=0: 1<=5 → j=1

j=1: 5<=5 → j=2

j=2: 循环结束。

最终数组 [1, 5, 5, 4, 6, 7], 不满足升序。

若 a 为 [1, 5, x, 4, 6, 7], 执行程序后 a 中数据满足 $a[0] \leq a[1] \leq \dots \leq a[n-1]$, 则 m, x 不可能为 **B**

A. 2, 4

B. 2, 5

C. 3, 6

D. 3, 10

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

13. 某物流转运中心采用大、小车转运货品, 物流转运系统根据货品到达信息提前判断派发车型, 超过 k 件派发大车, 否则派发小车。每分钟读取一次货品到达信息, 0 表示无货品到达, 1 表示有一件货品到达。每 m 分钟派发一次车辆。请回答下列问题。

(1) 系统从第 1 分钟开始运行, 前 10 分钟采集到的数据依次为 1, 0, 0, 0, 1, 0, 0, 0, 1, 1。若 k=5, m=10, 则系统 **A** (选填字母: A. 可提前 / B. 无法提前) 判断出派发车型。

(2) 编写程序模拟上述判断过程, 在划线处填上合适的代码。

输入 k, m 的值, 代码略

while True:

tot=0 ①

data = []; i = 0

while i < m - 1:

读取货品数据并加入列表 data, 代码略

tot += data[i]

r=m-i-1 ② r 是每个 m 段中剩余未判断的信息数量, 如果 tot+r <= k 那就只能派小车

if tot + r <= k or tot > k:

根据货品数量提前通知派发车型, 代码略

break

i += 1

延时 1 分钟, 代码略

if i==m-1 ③: # 如果上面的 while 没有被提前 break, 就是没有提前通知, i 会走到 m-1

未发送提前通知, 读取最后一个数据并派车, 代码略

else:

已发送提前通知, 读取后续货品数据直至派车, 代码略

定期安排人工采集植物生长数据与系统数据进行比较

提升智能终端的处理功能,对植物数据进行更精细的分类处理

优化网络,提高数据传输效率,使获得更准确数据

优化服务器相关程序的算法

优化升级服务器处理数据能力

14. 某园艺中心提供云养花服务, 实体养护由客户通过手机 App 在线认养的植物, 客户可实时查看通过系统中摄像头采集的植物画面。园艺中心还搭建了智能系统, 智能终端连接各类传感器和执行器, 采集数据后传输至服务器数据库中, 并根据植物需要进行灌溉、通风等操作。请回答下列问题。

(1) 若湿度传感器型号相同, 为提高湿度数据的精准性, 一个智能终端连接 A (单选)。
A. 多个湿度传感器 B. 一个湿度传感器

(2) 下列关于该系统的说法, 正确的有 AB (多选)。
A. 该系统实现了对植物生长环境的智能调节
B. 每个传感器编号不同以区分传送的数据
C. 智能终端与服务器之间只能进行单向数据传输
D. 传感器采集的数据都存储在对应的智能终端上

(3) 若智能终端出现故障, 则会引发的现象有 BCD (多选)。
A. 客户无法查看历史数据
B. 系统无法调节植物生长环境
C. 服务器无法获取传感器数据
D. 客户无法查看植物实时生长画面

D选项有争议
按题干中的说明, 摄像头其实不属于智能终端, 在智能终端故障的时候, 其实不影响摄像头。但如果把摄像头作为传感器包括在智能终端, 那么在智能终端故障时, 无法看摄像头

(4) 为了获取更真实可靠的植物生长数据, 请从数据采集、数据传输、数据处理三个角度中任选两个角度, 分别提出一条对应的改进建议。

(5) 将基地 1-4 月份不同花卉的土壤湿度数据导出到文件 data.xlsx 中, 部分数据如图 a 所示。统计各花卉 4 月份平均土壤湿度值, 选择平均土壤湿度值最高的前 5 种花卉的数据, 绘制如图 b 所示的柱形图。

	A	B	C	D
1	花卉名称	月	日	土壤湿度
2	绿萝	1	1	68.2
3	仙人掌	1	1	24.7
4	月季	1	1	56.8
5	多肉植物	1	1	31.5
6	龟背竹	1	1	72.9
7	茉莉花	1	1	52.3
8	蝴蝶兰	1	1	78.4
9	虎皮兰	1	1	29.1
10	杜鹃花	1	1	82.6

第 14 题图 a

角度	常见关键词
数据采集	①部署多传感器 ②选择精度更高的传感器 ③传感器定期校准 ④提高采样频率计算均值
数据传输	①校验 ②断点续传 ③有线/无线网络选择
数据处理	①异常数据检测 ②数据清洗 ③对多次采集数据求平均或中位数 ④滤波算法 ⑤与历史数据对比判断是否合理

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)。

导入相关库, 代码略

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

df1 = ① A 筛选 4 月的数据

df1 = ② D 求各种花卉的平均土壤湿度

df1 = ③ E 排序, 求平均土壤湿度最高的前 5 种

df2 = df1.head(5)

设置绘图参数, 选取数据, 绘制如图 b 所示的柱形图, 代码略

A. df[df["月"] == 4]

B. df1[df1.月 == 4]

C. df1.groupby("土壤湿度", as_index = False).mean()

D. df1.groupby("花卉名称", as_index = False).mean()

E. df1.sort_values("土壤湿度", ascending = False)

F. df1.sort_values("花卉名称", ascending = True)

程序中①②③处可选的代码有：

- A. `df[df["月"] == 4]`
- B. `df1[df1.月 == 4]`
- C. `df1.groupby("土壤湿度", as_index = False).mean()`
- D. `df1.groupby("花卉名称", as_index = False).mean()`
- E. `df1.sort_values("土壤湿度", ascending = False)`
- F. `df1.sort_values("花卉名称", ascending = True)`

15. 某车间有多类设备，设备用于承接并完成订单对应的各类生产任务。为减少设备启停带来的损耗，若某订单的完成时刻与后续订单的开始时刻相同，且所需的设备类型一致，则合并订单，安排该设备不间断运行。车间根据合并后的订单，优先分配完成时刻最早的该类设备。每台设备一次只能处理一个订单。请回答下列问题。

(1) 某段时间内的订单数据如图 a 所示，设备数据如图 b 所示。则分配给 104 订单的设备编号为 。按代码的运行，应该是 1，如果考虑设备不间断，选 0

本题有争议

订单编号	设备类型	开始时刻	运行时长
101	A	2	3
102	A	5	6
103	C	7	5
104	A	8	4

第 15 题图 a

设备编号	设备类型	完成时刻
0	A	0
1	A	2
2	A	4
3	C	5

第 15 题图 b

(2) 定义函数 merge，用于处理订单合并。

```
def merge(orders):
```

```
    res = []
```

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

```
        tp, st, t = orders[i][1], orders[i][2], orders[i][3]
```

```
        k = -1
```

```
        for j in range(len(res) - 1, -1, -1):
```

```
            if res[j][0] == tp:
```

```
                k = j
```

```
                break
```

```
            if k != -1 and res[k][1] + res[k][2] == st:
```

```
                res[k][2] += t
```

```
            else:
```

```
                res.append([tp, st, t])
```

```
    return res
```

将 orders[i] 与 res 中的数据比对类型

如果同类型，且开始时间 st 恰好等于 res[k] 中运行结束的时间，则合并

orders[[编号, 类型, 开始时间, 运行时长]...]

若 orders 初始为 [["101", "A", 1, 1], ["102", "C", 1, 3], ["103", "B", 2, 4], ["104", "C", 4, 3], ["105", "B", 7, 2], ["106", "C", 7, 2]]，执行 merge(orders) 后，

① 返回值中订单总数为 。合并后：res = ["A", 1, 1],

② 变量 k 的最终值为 。 ["C", 1, 8], ["B", 2, 4], ["B", 7, 2]

最后一次 106 并入下标 1 所在的 102

