

## 高三年级技术学科 试题

本试卷分两部分，第一部分信息技术，第二部分通用技术。满分 100 分，考试时间 90 分钟。

### 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）

1. 下列关于数据、信息与知识说法，正确的是
  - A. 所有的数据经过数字化后才能被存储下来
  - B. 获取数据的方式有很多种，包括人工方式、传感器获取
  - C. 信息在共享传递的过程中不会发生损耗，其价值也不变
  - D. 不同的人获取了相同的信息，就会构建相同的知识体系
2. 下列关于人工智能的说法，不正确的是
  - A. ChatGPT 是使用海量数据训练出来的实现“人机对话”系统，这属于符号主义人工智能
  - B. 在购物时智能客服和人工客服一起合作来回答顾客的问题，这是混合增强智能的一种应用
  - C. 家用扫地机器人可以自动完成清扫工作，说明人工智能可以改善人类的生活
  - D. 人工智能技术的发展提高社会生产力、促进经济发展，但不能解决所有问题

阅读下列有关材料，回答第 3 至 5 题。

某生鲜平台使用的智慧供应链系统是一个能够贯穿用户下单、基地采购、分选大仓和前置仓配送全链路的超级大脑，以高效准确的管理平台和系统取代传统的人工操作和管理。用户通过客户端 App 下单购买的每一件商品，都可以追溯到源头，从产地到运输到分选仓到前置仓，每一个环节都清清楚楚。在运输过程中，利用温度传感器等设备对不同的货物设置不同的温度控制要求，保证商品的质量。

3. 下列有关该系统组成的说法，正确的是
  - A. 该系统管理订单、用户等都可以在 App 或浏览器端实现，故无需硬件
  - B. 下订单的人员和维护系统的人员都是该系统的用户，但前置仓内的工作人员不是
  - C. 该系统中每一件产品只需要存储基本信息，无需存储其流转信息
  - D. 若该系统的网络发生故障，则部分功能不能使用，说明系统对外部环境有依赖性
4. 下列关于该系统的说法，不正确的是
  - A. 产品的溯源管理过程可以规范工作流程，提高工作效率
  - B. 系统通过大数据和算法预测，对各个前置仓进行智能分配和调拨商品，减少货物的损耗
  - C. 只要给系统配置不间断电源，系统就能正常运作不会发生错误
  - D. 可以通过安装杀毒软件或防火墙的方法来增强该系统的安全性
5. 下列有关搭建该信息系统说法，不正确的是
  - A. 该系统中的温度传感器属于硬件，用于采集当前环境的温度信息

- B. 该系统采用 C/S 架构，和 B/S 架构相比增加了系统的维护难度
- C. 完成该系统的搭建后，需要根据用户需求对系统进行测试
- D. 该系统中所采集的数据均存储在用户的手机 App 中
6. 下列关于网络技术的说法，正确的是
- A. 微信朋友圈中的消息可以设置仅部分好友可见，因此网络资源共享只能是数据资源共享
- B. 搭建网络中的数据通信系统时，只需要传输介质，包括有线传输介质和无线传输介质
- C. 若某计算机与服务器在同一个局域网中，则该计算机访问该服务器时无需通过网关
- D. 网络协议是一组标准及规则，其中 IP 协议的作用是管理被传送内容的完整性
7. 下列关于数制和编码的说法，正确的是
- A. 若十进制数  $n$  的末位为 0，则该数转为十六进制数后，末位可能为奇数
- B. 计算机本身只能对二进制进行运算，所以图形、声音都要转化成二进制才能计算处理
- C. 将模拟声音数字化需要经过采样、量化、编码三个过程，该过程为数模转换
- D. 采用 16 位色进行编码的 BMP 图像文件，其颜色数量总共有 256 种
8. 某二叉树用一维数组来表示如下表所示。该二叉树从根节点开始，按照从上到下，从左到右的顺序依次用 A-H 字母表示，该二叉树的中序遍历为

|    |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |
|----|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|
| 下标 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 元素 | A | B | C | D | E |   | F |   |   | G |    |    |    |    | H  |

- A. DBGEACFH
- B. DBGEACHF
- C. DBEGACHF
- D. ABCDEFGH
9. 假设栈 S 的最大长度为 3，其初始状态和终止状态均为空，经过一系列入栈和出栈的操作，若元素最后的出栈序列为 F, E, D, C, B, A，则可能的入栈顺序为
- A. ABCDEF
- B. ACDFEB
- C. BEFACD
- D. BFDECA

10. 有如下程序段：

```
def cal(n):
    if n <= 1:
        return 1
    if n % 2 == 0:
        return 2*cal(n-1)
    return 1+cal(n-1)
```

执行语句  $k=cal(5)$ ，则  $k$  的值为

- A. 6
- B. 7
- C. 10
- D. 11

11. 有如下程序段：

```
s = input()
head = 0; tail = 0; ans = 0; tmp = ''
q = ['']*100
flag = True
for i in range(len(s)):
```

```

if s[i] == ',':
    while head!=tail:
        tmp += q[head]
        head += 1
        if flag and head < tail:
            head += 1
        flag = not flag
    ans += int(tmp)
    tmp = ''; flag = True
elif '0' <=s[i]<='9':
    q[tail] = s[i]
    tail += 1

```

若输入 s 为 “1-500,2023900-,”，执行该程序段，变量 ans 的值为

- A. 100                      B. 22300                      C. 22351                      D. 22400

12. 有如下 Python 程序：

```

import random
s = input()
k = random.randint(1, len(s)-1)
i = 0
while k > 0 and i < len(s)-1:
    if s[i] > s[i+1]:
        k -= 1
        s = s[:i]+s[i+1:]
        if i > 0:
            i -= 1
    else:
        i += 1
if k > 0:
    s = s[:len(s)-k]

```

若输入的 s 值为” 8561324”，则执行该程序，输出 s 的值不可能为

- A. 51324                      B. 124                      C. 132                      D. 1324

二、非选择题（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. 某公路由于长期没有维修，路上出现了很多个坑。为了尽快填补好这些坑，交通管理部门决定对  $m$  处地段采取交通管制。将该公路看成一条直线，坑就是直线上的坐标点，坑所在的路段需要封控管制。例如某管制路段  $2 \sim 4$ ，需封控 2、3、4 路段。交管部门为了减少管制路段的长度，希望将这  $n$  个坑分成  $m$  段（一段可以只有一个坑），使得这  $m$  段公路的总长度最小。请你根据  $n$  个坑的位置（位置已按照从小到大进行排序），计算管制路段最小的总长度。代码运行效果如第 13 题图所示。

路段数量:4  
坑的坐标依次为:3, 4, 6, 8, 14, 15, 16, 17, 21, 25, 26, 27, 30, 31, 40, 41, 42, 43  
维修管制的路段依次为:  
3 ~ 8  
14 ~ 17  
21 ~ 31  
40 ~ 43  
管制总长度为 25

第 13 题图

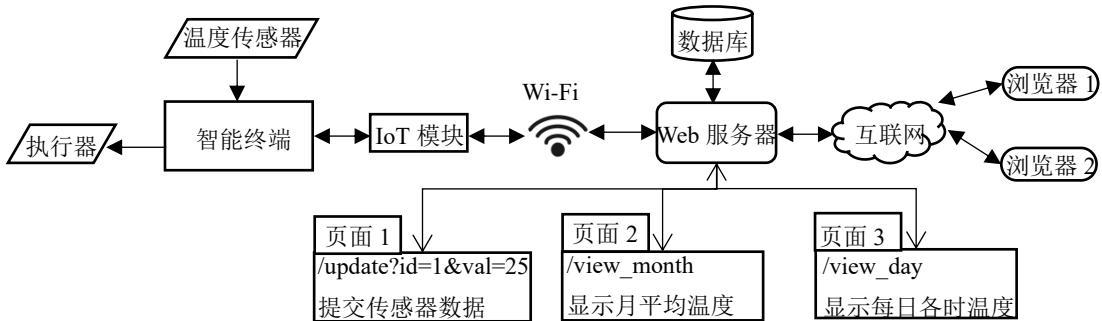
请回答下列问题：

- (1) 上图所示的例子中，若将路段数量修改为 5，则管制路段总长度为\_\_\_\_\_。
- (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。

```
m = int(input("路段数量:"))
s = input("坑的坐标依次为:").split(',')
n = len(s)
for i in range(n):
    s[i] = int(s[i])
flag = [False] * (n-1)
for i in range(1, m):
    k = -1
    for j in range(n-1):
        if _____①_____:
            if k == -1 or s[j+1]-s[j] > s[k+1]-s[k]:
                k = j
    flag[k] = True
print("维修管制的路段依次为: ")
dis, t = 0, 0
for i in range(n-1):
    if flag[i]:
        print(s[t], "~", s[i])
        dis += s[i]-s[t]+1
    _____②_____
```

```
print(s[t], "~", s[n-1])
dis = _____ ③ _____
print("管制总长度为", dis)
```

14. 小明为家庭小菜园搭建了一个简易的智能控温系统。系统通过传感器获取大棚内的温度，实现如下功能：若温度不在标准值范围内，自动打开温控系统；若温度达到了标准值，则关闭温控系统。系统的架构示意图如第 14 题图 a 所示。系统中的智能终端通过接口获取温度传感器的数据，经过 IoT 模块发送到 Web 服务器并保存到数据库中，执行器用于实现温度的控制。Web 服务器端程序采用 Flask Web 框架开发，其 IP 地址为 192.168.12.136，端口号为 8080。



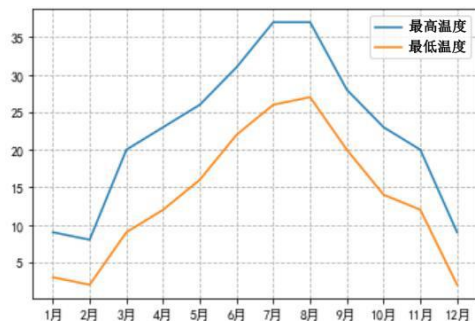
第 14 题图 a

请回答下列问题：

- (1) 该系统网络应用软件的实现架构是 B/S 架构，确定该架构方式属于信息系统前期准备中的 \_\_\_\_\_ 过程。（单选，填字母：A. 需求分析/B. 开发模式的选择/C. 概要设计/D. 详细设计）
- (2) 若一个月内开启控温系统的次数超过阈值，则系统会打开 LED 灯进行提醒，LED 灯属于第 14 题图 a 中的 \_\_\_\_\_。（单选，填字母：A. 服务器 /B. 传感器 /C. 执行器）
- (3) 若要访问大棚内每个月的平均气温，则访问的 URL 为 `http://_____`。
- (4) 系统正常工作 5 分钟后，小明发现访问页面的温度数据不再变化，刷新后仍不变。结合第 14 题图 a，小明应该检查该系统的硬件有（ ）（多选，A. 数据库/B. IoT/C. 浏览器/D. 执行器/E. 智能终端）。（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）
- (5) 将系统中 2022 年每个月的温度数据导出到文件中，例如“202201.xlsx”表示为 2022 年 1 月的温度数据，表格中部分数据如第 14 题图 b 所示。统计每月日最高气温平均值和日最低气温平均值并绘制线型图，部分 Python 程序代码如下，请在划线处填入合适的代码。

|   | A    | B | C | D     | E  |
|---|------|---|---|-------|----|
| 1 | 年    | 月 | 日 | 时间    | 气温 |
| 2 | 2022 | 1 | 1 | 9:00  | 5  |
| 3 | 2022 | 1 | 1 | 10:00 | 6  |
| 4 | 2022 | 1 | 1 | 11:00 | 8  |
| 5 | 2022 | 1 | 1 | 12:00 | 9  |
| 6 | 2022 | 1 | 1 | 13:00 | 9  |
| 7 | 2022 | 1 | 1 | 14:00 | 10 |

第 14 题图 b



第 14 题图 c

```

import pandas as pd
x, y_max, y_min = [''] * 12, [0] * 12, [0] * 12
for i in range(12):
    file = "2022"
    if i < 9 :
        file += "0"
    file += str(i+1) + ".xlsx"
    df = pd.read_excel(file)
    df1 = df._____①_____("日", as_index = False)
    ma = df1.max().气温 #分组后求最大值
    mi = df1.min().气温 #分组后求最小值
    x[i] = _____②_____ + "月"
    y_max[i] = ma.mean() #求平均
    y_min[i] = mi.mean()
plt.plot(x, y_max, label='最高温度') #绘制折线图
plt.plot(x, y_min, label='最低温度')
#设置绘图参数, 显示如第 14 题图 c 所示线型图, 代码略

```

15. 某工厂每天会收到多个订单，有  $n$  台机器对零件进行加工。为减少机器的损耗，需要在满足所有订单加工的情况下（订单即到即加工），机器开启数量尽量少。若开启  $n$  台机器不能满足订单即到即加工，则计算所有订单最少的平均等待时间。若给定某天内所有的订单信息，请计算需要开启的机器数量以及订单平均等待时间，代码运行效果图如第 15 题图所示（注意：若上一个订单结束时间为 9:00，下一个订单开启时间最早为 9:00）。

订单信息如下：(批次, 到达时间, 加工时间min)  
(A1, 9:00, 30) (A2, 11:30, 50) (A3, 10:40, 50) (A4, 10:00, 60) (A5, 9:20, 40)  
(A6, 11:00, 20) (A7, 10:20, 40) (A8, 9:30, 20)  
机器数量：2  
2台机器全部开启，订单平均等待2.5min  
第1台机器：  
A1:09:00~09:30, A8:09:30~09:50, A4:10:00~11:00, A3:11:00~11:50  
第2台机器：  
A5:09:20~10:00, A7:10:20~11:00, A6:11:00~11:20, A2:11:30~12:20

第 15 题图

请回答下列问题：

- (1) 上图所示的例子中，若机器有 10 台，则只需要开启\_\_\_\_\_台机器。
- (2) 定义如下 `data_sort(a)` 函数，参数 `a` 为列表，列表中每个元素包含三个数据项，依次分别对应订单批次、到达时间、加工时间（时间均转为分钟）。该函数实现将列表 `a` 按照订单到达时间升序排序。

```
def data_sort(a):
```

```

for i in range(len(a)):
    for j in range(len(a)-i-1):
        if _____:
            a[j], a[j+1] = a[j+1], a[j]

```

①划线处填入的语句为\_\_\_\_\_，可实现上述功能。

②若将加框处语句写错为 range(i, len(a)-1)，则下列 4 组数据中，若列表 a 的值为\_\_\_\_\_（单选，填字母）不能测试出问题。

- A. [['A1', 100, 30], ['A2', 120, 30], ['A3', 110, 30], ['A4', 140, 30], ['A5', 130, 30]]
- B. [['A1', 120, 30], ['A2', 110, 30], ['A3', 100, 30], ['A4', 130, 30], ['A5', 140, 30]]
- C. [['A1', 110, 30], ['A2', 140, 30], ['A3', 130, 30], ['A4', 100, 30], ['A5', 120, 30]]
- D. [['A1', 110, 30], ['A2', 120, 30], ['A3', 130, 30], ['A4', 140, 30], ['A5', 100, 30]]

(3) 实现计算开启机器数量的部分 Python 程序如下，请在划线处填入合适的代码。

```

def huan(n):
    #将分钟转换为时间 AA:BB 格式，返回值为字符串，代码略
    #读取文件中的信息，并存储在列表 order 中，代码略
    data_sort(order)
    n=int(input("机器数量: "))
    for i in range(len(order)):
        order[i].append(-1)                #order[i]追加一个元素-1
    mach = [-1] * n
    num, wait = 0, 0
    for i in range(len(order)):
        k = -1
        time = -1
        for j in _____①_____:
            t1 = mach[j]
            if k == -1:
                k = j
                time = order[t1][1]+order[t1][2]
            else:
                t2 = mach[k]
                if order[t1][1]+order[t1][2]<order[t2][1]+order[t2][2]:
                    k = j
                    time = order[t1][1]+order[t1][2]
    if k == -1 or num < n and _____②_____:
        mach[num] = i
        num += 1
    else:

```

```

    order[i][3] = mach[k]
    mach[k] = i
    if time > order[i][1]:
        wait += time-order[i][1]
        order[i][1] = time
if num < n:
    print("只需开启"+str(num)+"台机器")
else:
    print(str(n)+"台机器全部开启, 订单平均等待"+str(round(wait/len(order), 2))+"min")
for i in range(num):
    print('第'+str(i+1)+'台机器: ')
    p = mach[i]
    ans = ''
    while p!=-1:
        ans = order[p][0]+' :'+huan(order[p][1])+' ~'+huan(order[p][1]+order[p][2])+' , ' + ans
        p = _____③_____
    print(ans[:-1])

```