

温州市普通高中2025届高三第一次适应性考试

技术试题卷 参考答案

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）

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

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 (1) ① `range(1,len(c))` 或 `range(1,3)` 或 `[1,2]` (2分)
 ② `st=c[v-1]` 或 `st=100+70*(v-1)` (2分)
 ③ `i<c[v]` and `flag[i]!=0` 或 `i<c[v]` and `flag[i]>0` 或 `i<c[2]` and `flag[i]>0` (2分)
 _____ and _____ 前后的次序不能反

(2) C (1分)

14 (1) A (1分)

(2) B (1分)

(3) ABD (2分)

(4) 聚焦于减少数据传输频次

如延长智能终端数据采集时间间隔 或 降低智能终端数据采集频率，
 或者 智能终端缓存一段时间内的数据，计算平均值、中位数等统计值后上传
 或者 各智能终端错峰错时上传数据

分散处理任务

如将部分服务器程序写到智能终端中。
 将服务器阈值判断的程序写到智能终端

优化服务器端数据处理算法

或其它合理答案
 定期清理服务器或者数据库库中数据
 将服务器/客户端架构改为C/S架构

（一种方法一分，多种方法只看前两种，硬件方面不给分（题目要求） 共2分）

(5) ①D ②B ③G ④E (每空1分)

15 (1) 3 (1分)

(2) ①6 (1分)

②i, -1, -1 或者 n-i-2, n-1 (2分)

(3) ①`cq[tail][1]=head` (2分)

②`restp += tnum` 或者 `restp += psg[1]` (2分)

③`endt = t` 或者 `endt = psg[0]` (2分)

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

温州市普通高中2026届高三第一次适应性考试

第一部分 信息技术（共 50 分）

2025. 11

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）阅读下列材料，回答第 1 至 3 题。

学生在校园智慧班牌上查看教师发布的信息，可通过人脸识别在班牌上签到，考勤信息以图文形式上传到服务器。家长可通过手机 APP 查看孩子的在校情况，并与孩子留言互动。

1. 下列关于数据的说法，正确的是 **D**
- A. 电子班牌可存储未经数字化的数据
 - B. 电子班牌服务器的数据以十六进制存储 **二进制**
 - C. 学生签到数据的价值不会随着时间的推移而变化
 - D. 家长手机 APP 中的孩子在校情况的数据来源于数据库
2. 下列做法中，不利于该信息系统中数据安全的是 **D**
- A. 定期备份系统数据✓
 - B. 定期删除或归档日志文件 **（定期： 间隔时间可能不会很短）**
 - C. 对登录管理端的教师设置访问控制权限✓
 - D. 将学生姓名和家长手机号码存储在公共终端
3. 下列关于该系统中人脸识别技术的说法，正确的是 **C**
- A. 人脸识别过程不依赖于传感器 **需要摄像头传感器**
 - B. 学生通过人脸识别签到生成的数据是结构化数据 **不一定**
 - C. 人脸识别的准确性与抓拍人脸摄像头的清晰度有关✓ **摄像头分辨率影响识别效果，显示屏分辨率不影响人脸识别**
 - D. 人脸识别技术需要事先构造知识库和推理引擎

阅读下列材料，回答第4至5题。**人脸识别一般属于联结主义，构造知识库和推理引擎属于符号主义**

某公司食堂引入自助点餐系统，方便员工在办公区使用专用平板提前点餐。员工通过手机动态验证码、刷卡等方式在平板 APP 上登录并点餐，点餐数据通过网络传输至服务器，食堂根据订单数据准备菜品。

4. 下列关于该信息系统组成与功能的说法，正确的是 **A**
- A. 员工勾选菜品体现了数据输入功能
 - B. 该系统的数据处理均由服务器完成 **部分数据处理可能在智能终端进行**
 - C. 点餐平板的操作系统为应用软件
 - D. 员工登录的动态验证码由点餐平板生成
5. 下列关于该系统硬件和网络的说法，不正确的是 **B** **系统内每个网络设备的IP都不能冲突**
- A. 点餐平板一定有处理器和存储器
 - B. 该系统中所有点餐平板应分配相同IP 地址
 - C. 点餐平板接收饭卡信号无需网络支持
 - D. 点餐平板可通过计算机网络访问该系统
6. 下列关于一幅 1280×1024 像素 16 位色 BMP 位图的说法，正确的是 **C** **可通过：移动通信网络、计算机网络**
- A. 放大后不会失真 **会**
 - B. 色彩越丰富，存储容量越大 **已知是16位色，图像内容、色彩与容量无关**
 - C. 某个像素的编码用十六进制表示可能是 0000H **16位色，16位二进制→4位十六进制**
 - D. 若将该图像另存为 256 级灰度 BMP 位图，则存储量变为原来的 16 倍 **16位：8位=2:1**

7. 某算法的部分流程图如第 7 题图所示，输入下列数据，输入值与其它三项不同的是 **C** **求质因数的个数**

- A. 8 ^{2*2*2} B. 18 ^{2*3*3} C. 60 ^{2*2*3*5} D. 105 ^{3*5*7}

8. 栈 s1 和 s2 从栈底到栈顶的元素分别依次为 A、B、C 和

D、E、F，约定两种操作：

①将栈 s1 中栈顶元素出栈后入至栈 s2；

②将栈 s2 中栈顶元素出栈并输出。

经过若干次操作后，栈 s1、s2 均为空且**第四个**输出

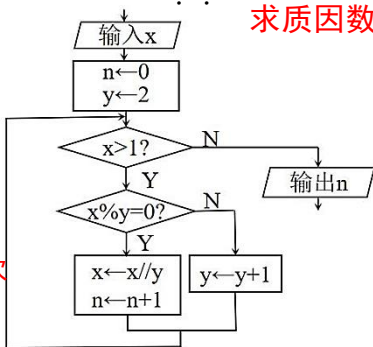
的元素为 **C**，下列说法正确的是 **C**

A. 总操作次数可能大于 9 **s1→s2 3次，s2出栈6次 总共9次**

B. 第一个输出的元素可能是 A

C. 在元素 C 前输出的序列有 4 种

D. 元素 D 可能在 E 之前输出 **D出栈必定在E之后**



第 7 题图

F出，E出，D出，C入s2，C出 FEDC

F出，E出，C入s2，B入s2，B出，C出，FEB C

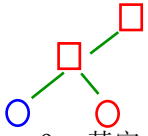
F出，C入s2，B入s2，A入s2，A出，B出，C出，FABC

F出，C入s2，B入s2，B出，A入s2，A出，C出，FBAC

C
B
A

F
E
D

□ □ □ C



方框：非叶子结点，圆形：叶子结点。次序未知
原中序：○□○□○□，去掉任意蓝色○，即为新的结果

9. 某完全二叉树包含节点 A、B、C、D、E、F，其中 B、C、E 为叶子节点，现删除一个左叶子节点，则新二叉树的中序遍历结果可能是 A

A. ABFED

B. BCADF_x

C. DFCEA_x

D. DBAFE_x

10. 使用列表lst模拟链表结构，如第 10 题图a所示，每个节点包含数据区域和指针区域，h为头指针。定义如下函数：

```
def con(p, m):
    c = lst[p][1]
```

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

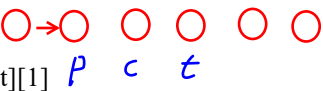
```
        t = lst[c][1]
```

```
        lst[c][1] = lst[t][1]
```

```
        lst[t][1] = lst[p][1]
```

```
        lst[p][1] = t
```

a: 6 12 26 19 18 36
b: 6 12 18 19 26 36



	数据区域	指针区域
0	19	5
1	6	3
2	26	0
3	12	2
4	36	-1
5	18	4

第10题图a

	数据区域	指针区域
0	19	2
1	6	3
2	26	4
3	12	5
4	36	-1
5	18	0

第10题图b

将链表调整后结果如第 10 题图b所示，则主程序调用的函数为 B

A. con(3,3)

B. con(3,2)

C. con(2,2)

D. con(2,3)

11. 列表a中每连续 3 个元素依次存储同一考生的考号、笔试成绩和面试成绩，即a[0]为某考生考号，a[1]为该考生笔试成绩，a[2]为该考生面试成绩，后续元素以此类推。原始数据已按考生考号升序。输入考生考号key，查找该考生的笔试、面试成绩。

实现该功能的Python程序如下：

```
i=0; j=len(a)//3-1
```

```
while i <= i:
```

```
    m = (i+j)//2*3
```

```
    if a[m] == key:
```

```
        print("考号", key, "笔试成绩为", a[m+1], "面试成绩为", a[m+2]);
```

```
        break elif a[m] < key:
```

```
            i = _m//3+1
```

```
    else:
```

```
        j = _m//3-1
```

(1)(2)(3)(4)划线处可选填的有 B

①len(a) // 3 - 1 ✓

②len(a)-3

③(i+j)//2

④(i+j)//2 * 3

⑤m//3 + 1

⑥m//3 - 1

⑦m + 3

⑧ m - 3

则划线处应依次填入的是

A. ①③⑤⑥

B. ①④⑤⑥

C. ②③⑦⑧

D. ②④⑦⑧

12. 有如下 Python 程序段：

```
a = [5, 8, 6, 4, 2, 5, 4, 8, 3, 1, 6, 3]
```

```
b = [0, 2, 3, 6, 7, 8, 9, 12]
```

```
k = 0; m = 0
```

```
for i in range(b[len(b) - 2]):
```

```
    if i == b[k]:
```

```
        k += 2
```

```
    j = b[k]
```

```
    while j <= b[k+1]:
```

```
        if a[i] + a[j] < m or m == 0:
```

```
            m = a[i] + a[j]
```

```
        j += 1
```

运行程序段后，m 的值是 B

A. 4

B. 6

C. 7

D. 8

- 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. 某博物馆门口寄存柜有小、中、大三种规格，柜子从 0 开始依次编号，小柜编号<中柜编号<大柜编号。存件时，用户输入柜子规格，柜机搜索相应规格的柜子，有空柜则打开编号最小的空柜并输出取件码，否则提示用户选存其它规格的柜子。取件时，用户仅需输入正确的取件码即可取件。请回答下列问题：

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

漏写range

(1, len(c))

generate def genpas():

生成当前未被使用的四位数字取件码并返回, 代码略
c = [100, 70, 50] # 该柜机小、中、大三种柜子的数量
for i in range(1, len(c)) 或 range(1, 3)

柜子编号从0开始

c[i] = c[i] + c[i - 1] 累加和
flag = [0] * c[2] 桶标记
while True:

c	100		70		50	
c	100		170		220	
	0	99	100	169	170	219

operation op = input("存件请输入 p, 取件请输入 g")

if op == "p":

v = int(input("输入需寄存的柜子规格。小柜输入 0, 中柜输入 1, 大柜输入 2"))

if v == 0:

st = 0 小柜从0开始检索

else:

st = c[v-1] 或 st = 100 + 70 * (v-1)

从c[v-1]开始检索有无可用柜子

"柜号在范围内"和

"柜子为满"两个条件

while i < c[v] and flag[i] != 0 或 i < c[v] and flag[i] > 0 或 i < c[2] and flag[i] > 0

i += 1

if i < c[v]:

flag[i] = genpas()

打开 i 柜并打印取件码小票, 代码略

else:

print("此规格的柜子暂时没有空柜, 请选择其它规格的柜子")

else:

取件代码略

(2) 结合题意和上述程序, 用户 A 选择中柜存入后未取件, 一段时间后, 用户 B 选择存入中柜, 则下列说法正确的是 ▲ C (单选)。

A. 用户 A 的柜号一定小于用户 B 的柜号 期间可能有其他人取件, 空闲柜子编号可能小于用户 A 的编号

B. 用户 A 和用户 B 的取件码可能相同

C. 用户 A 和用户 B 的柜号范围为 100 ~ 169

c	100		70		50	
c	100		170		220	
	0	99	100	169	170	219

14. 小张为自家中餐馆搭建了环境监测系统, 在 20 个包厢各设置了 1 个监测点。每个智能终端连接温度传感器、CO₂ 传感器, 新风和温控设备, 智能终端将智能终端编号和采集的数据通过网络传输到服务器。服务器根据阈值判断出异常情况时, 通过智能终端控制执行器。餐馆工作人员通过浏览器访问系统数据。请回答下列问题:

(1) 若服务器网络断开, 则下列功能不受影响的是 ▲ A (单选)。

A. 传感器采集环境数据 B. 智能终端传输数据到服务器 C. 浏览器访问实时数据

(2) 该餐馆增设包厢, 小张在每个新包厢各设置一个监测点。智能终端、传感器、执行器都采用与原来相同品牌相同型号的设备。搭建好硬件和网络后, 必须 ▲ B (单选)。

A. 修改服务器程序

B. 编写新增智能终端的程序

C. 编写客户端程序

修改并烧录

例: <http://192.168.1:8000/input/id=08&val1=xx%val2=xxx>

(3) 下列关于该系统的说法, 正确的有 ▲ ABD (多选)。

A. 可以为不同包厢设置不同的数据采集时间间隔

B. 服务器既可以处理来源于智能终端的数据, 也可以处理来源于客户端的数据 ✓

C. 该系统的数据和程序都存储数据库中

D. 系统搭建完毕后需进行硬件、软件、网络等测试 ✓

(4) 不改变服务器硬件, 不减少监测点的前提下, 写出两种减轻服务器处理数据负荷的方法。

聚焦于减少数据传输频次

如延长智能终端数据采集时间间隔 或 降低智能终端数据采集频率, 或者 智能终端缓存一段时间内的数据, 计算平均值、中位数等统计值后上传 或者 各智能终端错峰错时上传数据

分散处理任务

如将部分服务器程序写到智能终端中。将服务器阈值判断的程序写到智能终端

优化服务器端数据处理算法

定期清理服务器或者数据库库中数据, 将服务器/客户端架构改为 C/S 架构

超出阈值不发、相同值不发?

- (5) 将系统中 2025 年 7 月数据导出到文件 data.xlsx 中，部分数据如第 14 题图 a 所示。输入某一监测点智能终端编号及待查环境因素编号，统计 7 月该监测点待查环境因素值超过阈值记录数最多的 5 日，相同异常记录数的日期一并输出。如：输入终端编号 1，待查环境因素编号 1，则输出结果如第 14 题图 b 所示。

实现上述功能的部分 Python 程序如下，请选择合适的代码填

```
import pandas as pd
df=pd.read_excel("data.xlsx")
```

id=input("输入待查的终端编号：") 字符串类型

xm=int(input("输入待查的环境因素编号(0 表示温度，1 表示二氧化碳)：")) 数字类型

yz=[["温度",28],["二氧化碳",1000]] #环境因素名称和阈值

df1= D ① df[df["终端编号"]==int(id)]

x= B ② yz[xm][0]

df1=df1[df1[x]>yz[xm][1]]

df1= G ③ df1.groupby("日期",as_index=False)[x].count()

df1= E ④ df1.sort_values(x,ascending=False)

df1=df1[df1[x]>=df1.values[4][1]]

①②③④处可选代码有：

A. yz[xm]

B. yz[xm][0]

C. df[int(df["终端编号"])==id] X

D. df[df["终端编号"]==int(id)]

E. df1.sort_values(x,ascending=False)

F. df1.sort_values(x)

G. df1.groupby("日期",as_index=False)[x].count()

日期	时间	终端编号	温度	二氧化碳
07月01日	10:00	1	31	962
07月01日	10:00	2	29	728
07月01日	10:00	3	28	955
07月01日	10:00	4	30	779
07月01日	10:00	5	32	703

第 14 题图 a

	日期	二氧化碳
11	07月15日	5
15	07月19日	4
9	07月13日	4
0	07月01日	3
3	07月05日	3
21	07月26日	3

第 14 题图 b

15. 景区根据旅行团预约的到达时间顺序安排接驳车，接驳车座位数为 m ，每辆车往返需 d 分钟，返回后可重复使用。最早到达的乘客最多等 w 分钟，期间满员即发车，否则按期间最后一批到达乘客的到达时间发车。编写程序，计算景区当天需要安排的车辆数。请回答下列问题。

- (1) 景区某一天申请数据如下，若 $m=20$ ， $w=10$ ， $d=30$ ，则需要安排的车辆数为 3。

团号	1	2	3	4
到达时间	08:38	08:00	08:15	08:05
人数	20	5	35	10

2	4	3	1
8:00	8:05	8:15	8:38
5	10	35	20

8:05 bus+2
8:05 bus+1
8:35

- (2) 自定义函数 bsort，实现对列表 a 按元素的第一项进行升序排序。

def bsort(a): 冒泡全排序

n=len(a)

for i in range(n-1):

for j in range(0,n-i-1):

if a[j][0]>a[j+1][0]:

a[j],a[j+1]=a[j+1],a[j]

return a

def bsort2(a):

n=len(a)

for i in range(n-1):

for j in range(): i, -1, -1 或 n-i-2, n-1

if a[j][0]>a[j+1][0]:

a[j],a[j+1]=a[j+1],a[j]

else:

break

return a

什么排序?

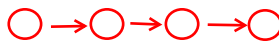
- ①若 a 的值为[[498, 10],[480, 25], [515, 30],[522,35],[498, 35],[530, 15],[500, 5]]，则程序运行后，方框内的语句执行的次数为 6 (填数字)。升序，逆序对个数 6

- ②bsort2 函数实现与 bsort 函数相同的功能，请在划线处填入合适的代码。

题干分析：

1. 满员发车，车辆不足必须增派车辆
2. 超时必发车，最多延后 w 时间，必须发车
3. 车辆 d 时间后可再次使用
4. 滞留的乘客数不会超过 m (座位数)

8:05



(3) 实现计算最少数量的接驳车功能的部分 Python 程序如下，请在划线处填入合适的代码。

def add(time): **cp列表：接驳车池**

global head, tail, d

if head != -1 and cq[head][0] <= time: **队首车辆可用** **操作1：出队，队尾append()**

cq[tail][1]=head **会增加cp列表长度**

tail = head

head = cq[head][1]

cq[tail] = [time + d, -1]

操作2：不增加列表长度，队头元素称为新增的队尾，就是循环队列

且此循环队列中可能会在尾部添加元素

else: **先阅else部分**

cq.append([time + **d**, -1])

在列表 cq 末尾添加元素[time + d, -1]

if head == -1:

head = 0 **如果是空列表，成为链头**

time+d后的时间会是判断车辆是否可用的关键

else:

cq[tail][1] = len(cq) - 1 **返回的车成为队列尾**

tail = len(cq) - 1

"""读取 m、w、d；读取旅行团申请的数据表存入 psgs 列表，每个元素包含到达时间（用分钟表示，如 08:00 表示为480）和人数两个数据项，代码略"""

passenger psgs = bsort(psgs) **乘客按照到达时间排序，利于从首开始遍历**

cq = []

restp = 0; head = tail = -1

for psg in psgs:

t, tnum = psg[0], psg[1] **时间，人数**

rest passenger if restp == 0: **剩余乘客数为0**

firt = t

elif t - firt > w: **有上次剩余乘客且等待超时，先送走**

本批乘客开始处理前处理超时滞留乘客

add(endt)

restp = 0 **派车 endt 是上一团到达时间，也作为发车时间**

firt = t **当前团成为新的等待起点**

restp += tnum 未超时滞留乘客+当前乘客=当前要处理的乘客数

if restp >= m:

for i in range(restp // m):

add(t) **派车**

restp -= m **减人**

firt = t **记录下这批乘客的到达时间，用于计算滞留时间**

endt=t 记录当前发车批次中，本团最后到达时间（可能的下一次发车时间）

totc = len(cq)

if restp > 0 and (head == -1 or cq[head][0] > endt):

totc += 1

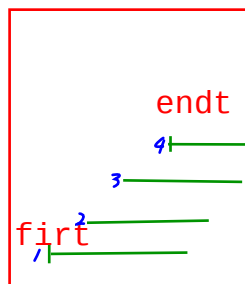
剩余人数处理

print("需要安排的车辆数为", totc)

错误

endt=t+w?

endt=firt+w



end

4

3

2

1

firt

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209