

# 2025年1月浙江省普通高校招生选考科目考试 技 术

姓名：\_\_\_\_\_ 准考证号：\_\_\_\_\_

## 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、多选、错选均不得分）

阅读下列材料，回答第1至2题：

某市举行科普现场宣传活动，包括科学实验秀、科技故事分享等内容。该活动还通过视频直播、网站、报纸等媒介进行传播。市民可参加线下活动，也可注册后参加线上活动。

1. 下列关于该活动中数据和信息的说法，正确的是 **C**
- A. 同一活动内容的数据在不同的媒介上表现形式相同 **可以有多种表现形式**
  - B. 活动内容中图像和文本的数字化方法是完全一致的 **不一致**
  - C. 活动内容通过多种媒介传播有助于信息的共享
  - D. 同一活动信息的价值不会随着时间的推移而变化
2. 关于信息安全与信息社会责任，下列行为合适的是 **A**
- A. 视频直播时接受社会监督
  - B. 观众将科学实验秀制作成视频销售
  - C. 使用他人信息注册参加线上活动
  - D. 将注册信息发到活动用户群里

阅读下列材料，回答第3至6题：

某连锁餐厅的智能监管系统实现从食材验收区到加工区全程监管。在食材验收区，智能验收秤采集并保存食材的品种、重量等数据，同时将数据发送到服务器；在食材加工区，AI抓拍设备自动识别员工的违规行为，并通过音箱发出语音提示，如“请穿工作服”，同时将抓拍数据发送到服务器。管理员可通过安装有监管系统APP的移动终端查看各连锁餐厅的数据。

3. 下列关于该信息系统功能的说法，不正确的是 **A** **AI抓拍设备也有数据的收集功能**
- A. 系统的数据收集和输入功能由智能验收秤实现
  - B. 语音提示功能可采用语音合成技术实现
  - C. 监管系统APP可与服务器进行双向数据传输
  - D. 识别员工违规行为的功能属于人工智能技术的应用
4. 某地区有100家连锁餐厅，每家餐厅安装有5台AI抓拍设备。若使用二进制对这些设备进行编码，二进制的前几位表示餐厅号，其余位表示设备号，则所需的二进制位数最少是 **A**
- A. 10
  - B. 9
  - C. 8
  - D. 7
- 100家餐厅需要7位 ( $<2^7$ )**  
**5台设备需要3位 ( $<2^3$ )**  
**7+3=10位**
5. 下列关于该信息系统中硬件的说法，正确的是 **C**
- A. 系统的硬件不包括管理员使用的移动终端
  - B. 系统中的传感器都用于获取图像数据 **还有重量等数据**
  - C. 智能验收秤的组成部件一定有存储器
  - D. AI抓拍设备和服务器一定通过有线方式直接相连 **有线、无线都可以**
6. 下列关于该信息系统中软件的说法不正确的是 **D**
- A. 监管系统APP属于应用软件
  - B. 软件要随着系统需求的变化而不断完善
  - C. AI抓拍设备需要在软件的支持下工作
  - D. 系统中，服务器和移动终端需要使用相同的操作系统

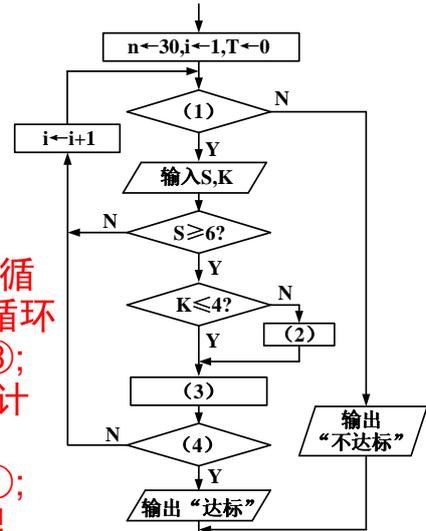
7. 某校组织学生进行阳光月跑活动, 30 天内总里程满 50km 达标。每天组织一次跑步, 平均速度(S)达到 6km/h 时方可计入总里程(T), 每天的跑步里程(K)若超过 4km 按 4km 计入, 否则按实计入。判断某学生是否达标的部分流程图如第 7 题图所示, 表达式为

- ① $T \leftarrow T+K$     ② $T \geq 50?$     ③ $i \leq n$     ④ $K \leftarrow 4$

则 (1) ~ (4) 处表达式序号依次为 **B**

- A. ③①④②  
B. ③④①②  
C. ②①④③  
D. ②④①③

(1)处的下面为输入框, 同时 $i=i+1$ 的循环返回到(1)处的上方, 所以该处为循环判断该学生的30天运动记录, 填入③;  
(2)处当次跑步里程 $K > 4$ 是,  $K$ 按照 4 计算, 填入④;  
(3)处统计学生跑步总里程  $T$ , 填入①;  
(4)处判断学生总里程 $T$ 是否满50公里, 填入②;



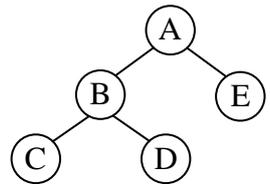
第 7 题图

8. 有后缀表达式 “1 3 + 2 \* 3 + 2 \*”, 现利用栈计算该表达式: 从左向右扫描, 遇到数字时, 数字入栈; 遇到运算符时, 两个元素出栈, 用运算符计算, 所得结果入栈, 如此反复操作, 直到扫描结束, 栈顶元素是 **B**     $[(1+3) * 2 + 3] * 2 = 22$

- A. 21    B. 22    C. 23    D. 24

9. 某二叉树如第 9 题图所示, 若其中的一个叶子节点增加右子树(仅包含节点 N), 则新二叉树的中序遍历结果不可能是 **D**

- A. CNBDAE    B. CBDNAE  
C. CBDAEN    D. NCBDAE



未增加前的中序遍历: CBDAE  
增加的右子树N只能出现在CDE右边

10. 数组元素  $a[0] \sim a[n-1]$  已按升序排列, 现要将  $a[pos]$  ( $0 \leq pos \leq n-1$ ) 的值加 1, 并保持数组的有序性不变, 实现该功能的程序段如下, 方框中应填入的正确代码为 **B**

```
t = a[pos]+1    出列
i = pos
while [ ]:    边比边移动
    a[i] = a[i+1]    前移
    i += 1
a[i] = t    归队
```

插入排序

数值变大, 在升序序列中只会调整到后面, 前面的不变  
数值位置不动

- A.  $i < n-1$     B.  $i < n-1$  and  $t > a[i+1]$   
C.  $i < n-1$  and  $a[i] > a[i+1]$     D.  $i <= n-1$  or  $t > a[i]$

11. 对于任意非空字符串 s, 甲、乙程序段输出结果相同, 则乙程序段加框处的正确代码为 **D**

|  |  |
|--|--|
| <pre>def f(s,t):     if t &gt;= len(s)-2:         return s[t]     return f(s,t+2) + s[t] print(f(s,0))</pre> <p style="text-align: center;">甲程序段</p> | <pre>r = "" n = len(s) for i in range(0,n,2):     [ ] print(r)</pre> <p style="text-align: center;">乙程序段</p> |
|--|--|

- A.  $r = s[n-i]+r$     B.  $r = r+s[n-i-1]$   
C.  $r = r+s[i]$     D.  $r = s[i]+r$

甲程序段, 从t位置出发, 间隔提取, 并倒序连, 例如“abc123”, 输出值为“2ca”;乙程序完成类似功能(以s=“abc123”为例)

- A.  $r=s[n-i]+r$ ,  $i=0$  时, 下标越界, 错误;  
B.  $r=r+s[n-i-1]$ , 输出结果为:“31b”, 错误;  
C.  $r=r+s[i]$ , 输出结果为:“ac2”, 错误;  
D.  $r=s[i]+r$ , 正确。

本题要注意:甲程序段中,  $t \geq \text{len}(s) - 2$  条件, 若条件更新为:  $t \geq \text{len}(s) - 1$ , 字符串长度奇数与偶数输出值将不同。

12. 列表 c 长度为 100, 如第 12 题图所示, 其中 c[10]~c[89]各元素的值均为 10 以内的随机正整数。执行如下程序段, 输出的最后一行是 **B**

|      |   |   |   |   |   |   |   |   |   |   |       |    |    |    |    |    |    |    |    |    |    |
|------|---|---|---|---|---|---|---|---|---|---|-------|----|----|----|----|----|----|----|----|----|----|
| i    | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ..... | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |
| c[i] | 1 | 1 | 1 | 1 | 2 | 3 | 2 | 3 | 2 | 3 | ..... | 2  | 3  | 2  | 2  | 2  | 2  | 1  | 1  | 1  | 1  |

第 12 题图

qa = [0,0,0,0,0]

qb = [0,0,0,0,0]

h,t = 0,4

temp = 0

for k in range(100):

qa[t] = c[k]

qb[t] = temp+qa[t]-qa[h]

print(qb[h], qb[t])

temp = qb[t]

t = (t+1) % 5

h = (h+1) % 5

进一个新的, 出一个旧的, 始终保持4个值的求和

qb[t]=temp+c[k]-c[k-4]

qb[t]=c[k]+c[k-1]+c[k-2]+c[k-3]

循环队列一共5个位置, tail占一个空位, 实际队列一共4个值

A. 5 4

B. 8 4

C. 9 5

D. 9 9

## 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. 根据机器的负载率对工厂的 6 台机器 (编号 0~5) 进行监控和调度。调度规则是: 每隔 1 小时采集 1 次各台机器的负载率 (负载率用百分制表示, 例如, 负载率 95% 表示为 95, 机器休息时的负载率为 0), 负载率超过 90 的机器都调度休息; 如果所有机器负载率都不超过 90, 则负载率最高的机器休息, 若多台机器负载率同为最高, 则编号最小的机器休息。休息的机器在休息 1 小时后再次工作。

请回答下列问题:

(1) 若某次采集到 0~5 号机器的负载率依次为 75、85、88、0、88、87, 当前处于休息状态的机器编号是 3, 接下来休息的机器编号是 ▲ 2

(2) 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。

n = 6

a = [0] \* n #列表 a 长度为 n, 各元素值均为 0

#启动 0~4 号机器工作, 5 号机器休息, 代码略

While True:

#延时 1 小时, 再采集各机器负载率存入 a, a[i]存放 i 号机器的负载率, 代码略

t = 0

cnt=0

for i in range(n):

if a[i]>90: ①

超过90的都休息

#调度 i 号机器休息, 代码略

cnt += 1

if a[i] > a[t]:

t=i ②

不管是不是有超过90的, 都找最大值, 相等时要编号小的, t的编号本身小于等于i,不需要再写条件

elif a[i] == 0:

#调度 i 号机器工作, 代码略

if cnt==0: ③

判断有没有超过90的机器

#调度 t 号机器休息, 代码略

14. 某研究小组搭建了室外温度检测系统，在 4 所学校各设置了 1 个监测点。智能终端连接传感器，每隔 3 小时采集 1 次温度数据通过网络将温度数据传输到服务器。服务器根据数据判断出异常情况时，通过智能终端控制执行器发出预警信号。请回答下列问题：

传感器不能离智能终端很远

- (1) 在搭建该监测系统时，温度传感器与智能终端的配备总数量合理的是 **A▲** (单选，填字母：A. 4 个温度传感器和 4 个智能终端 / B. 4 个温度传感器和 1 个智能终端)。
- (2) 系统数据采集的时间间隔为 3 小时，可用于控制采集时间的设备是 **B▲** (单选，填字母：A. 传感器 / B. 智能终端)。
- (3) 编写智能终端程序时，不需要知道 **BD** (多选，填字母)。(注：全部选对的得 2 分，选对但不全的得 1 分，不选或有错的得 0 分)

- A. 与传感器连接的智能终端引脚                      B. 服务器的存储容量  
C. 服务器的地址及端口                                      D. 数据库的文件名

(4) 当服务器判定有异常情况时，除可通过智能终端控制蜂鸣器发出预警声音之外，请写出其他两种预警的具体方式。**通过智能控制终端控制LED灯闪烁;通过智能控制终端控制舵机转动;服务器发送预警邮件**

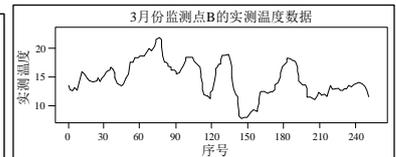
(5) 将系统中某年的数据导出到文件 data.xlsx 中，部分数据如第 14 题图 a 所示。其中“预报温度”列是指学校所在地的天气预报温度数据，“差值”列是指实测温度与预报温度相减的绝对值。现要由高到低输出 3 月份各监测点差值的平均值(如图 b 所示)，再用差值平均值最高的监测点的 3 月份实测温度数据绘制线形图(如图 c 所示)。

| 监测点 | 年    | 月 | 日 | 时间点 | 实测温度 | 预报温度 | 差值     |
|-----|------|---|---|-----|------|------|--------|
| A   | 2023 | 2 | 9 | 2   | 9.9  | 9.1  | 0.8    |
| B   | 2023 | 2 | 9 | 2   | 10.2 | 9.0  | 1.20.8 |
| C   | 2023 | 2 | 9 | 2   | 10.1 | 9.3  | 0.8    |
| D   | 2023 | 2 | 9 | 2   | 9.9  | 9.2  | 0.7    |
| A   | 2023 | 2 | 9 | 5   | 10.1 | 9.4  | 0.7    |

第 14 题图 a

监测点：B 差值平均值：1.5  
监测点：A 差值平均值：1.3  
监测点：D 差值平均值：1.2  
监测点：C 差值平均值：1.1

第 14 题图 b



第 14 题图 c

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("data.xlsx")
```

```
df1 = _____ ① C
df2 = _____ ② E
df2 = _____ ③ G
```

筛选3月份数据  
按监测点分类汇总算平均值  
从高到低按平均值排序

```
#依次输出 df2 中各监测点编号及其差值的平均值，如图 b 所示，代码略
#将 df2 中首行的监测点编号存入 uid，代码略 平均值最高的监测点uid
df2 = _____ ④ F
```

```
#创建 x 列表，长度为 248，元素依次为 0~247，表示 3 月份采集时间点的序号，代码略
plt.plot(x,df2['实测温度']) #绘制线形图
#设置绘图参数，并显示如图 c 所示的线形图，代码略
```

①②③④处可选代码有：

- A. df.groupby("月",as\_index=False).差值.mean() #分组求平均  
B. df[df["监测点"]==uid] #筛选  
C. df[df["月"]==3]  
D. df1.sort\_values("差值",ascending=False) #降序排序  
E. df1.groupby("监测点",as\_index=False).差值.mean()  
F. df1[df1["监测点"]==uid] **注意：筛选的数据是uid监测点3月份的数据**  
G. df2.sort\_values("差值",ascending=False)

15. 某市举行体育赛事活动,  $n$  所学校的选手已完成预赛, 现计划根据预赛的成绩挑选  $s$  名选手参加市决赛。成绩位列所在学校前  $w$  名次的选手直接入选, 剩余名额按成绩由高到低依次挑选, 成绩相同的选手一并入选, 选中的选手数一旦达到或超过  $s$  名, 挑选结束。

现给定所有选手预赛的成绩数据表, 每位选手的数据包含学校编号 ( $0 \sim n-1$ )、选手编号、成绩, 成绩数据表已按成绩由高到低排列。编写程序, 计算各选手的校内名次, 再按上述规则挑选决赛选手, 按成绩数据表中的顺序输出选手编号, 同时提供查询功能。选手校内名次的计算方法是: 若选手所在学校有  $m$  人成绩高于该选手, 则该选手的名次为  $m+1$ 。

在第 15 题图所示的样例中,  $n$ 、 $s$ 、 $w$  分别为 3、8、2, 根据图中前 3 行数据计算出了每位选手的校内名次, 进而选出实际入选的 9 名选手。

|      |      |      |      |      |      |      |      |       |      |      |       |       |
|------|------|------|------|------|------|------|------|-------|------|------|-------|-------|
| 学校编号 | 0    | 2    | 2    | 0    | 0    | 2    | 2    | 0     | 1    | 1    | 1     | 1     |
| 选手编号 | 0002 | 2027 | 2002 | 0072 | 0182 | 2071 | 2128 | 0012  | 1081 | 1002 | 1008  | 1208  |
| 成绩   | 198  | 185  | 183  | 182  | 182  | 177  | 177  | 176   | 175  | 163  | 161   | 161   |
| 校内名次 | 1    | 1    | 2    | 2    | 2    | 3    | 3    | 4     | 1    | 2    | 3     | 3     |
| 是否入选 | True | False | True | True | False | False |

第 15 题图

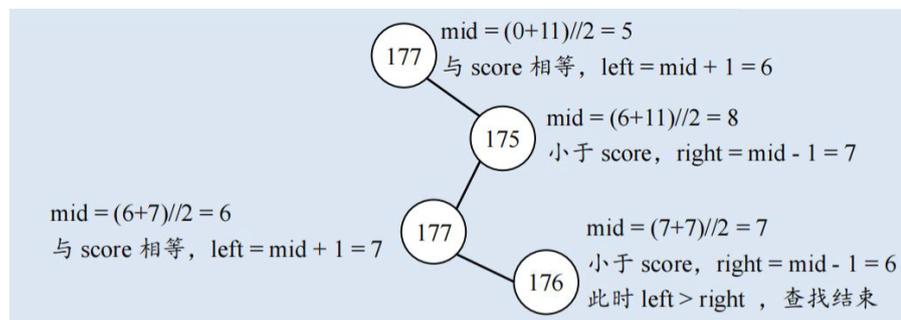
请回答下列问题:

- 对于第 15 题图所示前 4 行数据, 若  $s$ 、 $w$  分别为 5 和 1, 则 0 号学校入选人数是 ▲3。
- 定义如下 `search(data,sid,score)` 函数, `data` 列表每个元素的前 5 个数据项依次为学校编号、选手编号、成绩、校内名次、是否入选, 列表已按成绩由高到低排列。函数功能是查找选手编号为 `sid`、成绩为 `score` 的元素, 返回其下标, 若未找到则返回 -1。

```
def search(data,sid, score):
    left,right = 0, len(data)-1
    f = -1
    while left <= right:
        mid = (left+right)//2
        if score == data[mid][2]:
            f = mid
            left = mid+1
        elif score < data[mid][2]:
            left = mid+1
        else:
            right = mid-1
    if f == -1:
        return -1
    for i in range(f, len(data)):
        if data[i][2] != score:
            return -1
        elif data[i][1] == sid:
            return i
```

|      |      |      |      |      |      |       |       |       |      |       |       |       |
|------|------|------|------|------|------|-------|-------|-------|------|-------|-------|-------|
| 学校编号 | 0    | 2    | 2    | 0    | 0    | 2     | 2     | 0     | 1    | 1     | 1     | 1     |
| 选手编号 | 0002 | 2027 | 2002 | 0072 | 0182 | 2071  | 2128  | 0012  | 1081 | 1002  | 1008  | 1208  |
| 成绩   | 198  | 185  | 183  | 182  | 182  | 177   | 177   | 176   | 175  | 163   | 161   | 161   |
| 校内名次 | 1    | 1    | 2    | 2    | 2    | 3     | 3     | 4     | 1    | 2     | 3     | 3     |
| 是否入选 | True | True | True | True | True | False | False | False | True | False | False | False |

相等时往后找,  $f$  停在最后一次  $score$  的位置上



- 调用 `search` 函数, 若 `data` 列表长度为 12, `data[0][2]`, `data[1][2]`, ..., `data[11][2]` 的值依次为: 198, 185, 183, 182, 182, 177, 177, 176, 175, 163, 161, 161,  $score$  值为 177, 则 `while` 语句中循环体的执行次数是 ▲4。
- 程序中加框处代码有错, 请改正。

(3) 实现根据选手成绩（成绩不超过 200）计算校内名次，以及挑选决赛选手功能的 Python 程序如下，请在划线处填入合适的代码。

def proc(data,n,s,w): **说明r一定是关于学校的变量**

#创建 r 列表，共 n 个元素，每个元素的值均为[0,0,201]，代码略

heads = [-1,-1]

tails = [-1,-1]

cnt = 0

r中每个元素内容[校内名次, 当前总人数, 当前成绩]

for i in range(len(data)):

k=data[i][0]

r[学校编号][1]=人数

r[k][1] += 1

if data[i][2] < r[k][2]:

r[k][2] = data[i][2]

当前学生成绩低于前一个同学成绩，则更新r中的成绩记录，同时更新校内名次，即与总人数相同

r[k][0]=r[k][1]

如果当前学生成绩等于前一学生成绩，校内名次不更新

data[i][3] = r[k][0]

data[i].append(-1) #为 data[i]追加一个元素-1 为变成链表做准备

v = 1

if data[i][3] <= w: **还在校内前w名，直接入选**

data[i][4] = True

cnt += 1 **入选总人数增加**

v = 0 **v的值0、1两种，分别代表校内直接入选还是剩余人数中挑选**

if heads[v] == -1:

heads[v] = i

链队

else:

data[tails[v]][5] = i

tails[v] = i

p,q = heads[0],heads[1]

res = [] #res 列表用于存放入选决赛的选手编号，顺序与 data 列表保持一致

while cnt < s and q != -1: **将head[1]链中的数据挑选一部分入选**

tmp = data[q][2]

while q != -1 and data[q][2] == tmp: **同分的情况下，都可以入选**

while p != -1 and p < q

res.append(data[p][1])

在将q加入到res之前，head[0]中在此成绩之前的人都要加入到res,p指针遍历

p = data[p][5]

res.append(data[q][1])

data[q][4] = True

cnt += 1

q = data[q][5]

|      | p    |      |      |      |      | q    |      |       |      |      |       |
|------|------|------|------|------|------|------|------|-------|------|------|-------|
| 学校编号 | 0    | 2    | 2    | 0    | 0    | 2    | 2    | 0     | 1    | 1    | 1     |
| 选手编号 | 0002 | 2027 | 2002 | 0072 | 0182 | 2071 | 2128 | 0012  | 1081 | 1002 | 1008  |
| 成绩   | 198  | 185  | 183  | 182  | 182  | 177  | 177  | 176   | 175  | 163  | 161   |
| 校内名次 | 1    | 1    | 2    | 2    | 2    | 3    | 3    | 4     | 1    | 2    | 3     |
| 是否入选 | True | False | True | True | False |

while p != -1:

res.append(data[p][1])

在q之后的head[0]中的未加入到res中继续加入

p = data[p][5]

return res

"""读取 n、s、w；读取选手成绩数据表存入 data 列表，每个元素包含学校编号、选手编号、成绩、校内名次（初值为 0）、是否入选（初值为 False）5 个数据项，代码略

res = proc(data,n,s,w)

#输出 res 列表中的入选选手编号，代码略

#读取待查询的选手编号与成绩，调用 search 函数，根据返回值输出查询结果，代码略