

机密 ★ 考试结束前

温州市普通高中 2025 届高三第一次适应性考试

技术试题卷

2024. 11

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14页，第一部分1至6页，第二部分7至14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5题：

某运动会检录系统的部分功能是：系统管理员事先将比赛项目、参赛人员、检录时间、检录地点等信息输入到服务器中，系统自动生成各赛事检录信息。在各项比赛检录前，系统将检录信息发布在赛场的大屏幕上并发送到领队的手机APP，领队提示运动员检录。运动员到赛场检录终端通过人脸识别或者使用植入电子标签的号码牌进行身份核验，完成检录。

1. 下列关于该系统中数据与信息的说法，正确的是 **D**
 - A. 检录过程没有涉及数据处理
 - B. 该系统中的数据都是结构化数据
 - C. 检录信息对所有运动员价值相同
 - D. 运动员的检录行为会产生数据
2. 下列关于编码的说法，正确的是 **B**
 - A. 服务器中数据可以是未经数字化的
 - B. 编码后的数据一般以文件方式存储于计算机中
 - C. 为节省存储空间，运动员图像应采用 BMP 格式
 - D. 服务器中所有文件编码方式相同
3. 下列关于该信息系统组成与功能的说法，正确的是 **A**
 - A. 设计该系统时要考虑可能存在的安全隐患
 - B. 该系统可以直接在裸机上运行
 - C. 该系统的管理功能主要由各移动终端完成 **服务器端**
 - D. 该系统中的终端设备不具备数据存储功能
4. 下列关于该信息系统支撑技术的说法，不正确的是 **A**
 - A. 已检录的赛事数据存储在服务器的**内存条**中 **数据库中，一般数据库文件存在服务器硬盘里**
 - B. 领队的手机和服务器的基本原理基本相同
 - C. 该系统使用了 RFID 技术
 - D. 摄像头主动采集人脸的过程应用了传感技术

A: 领队手机可以通过wifi (计算机网络) 与服务器通信; C: 网络资源包括计算机系统、数据通信系统以及网络软件和网络协议三部分构成; D: 如果在同一个局域网内, 不需要经过局域网

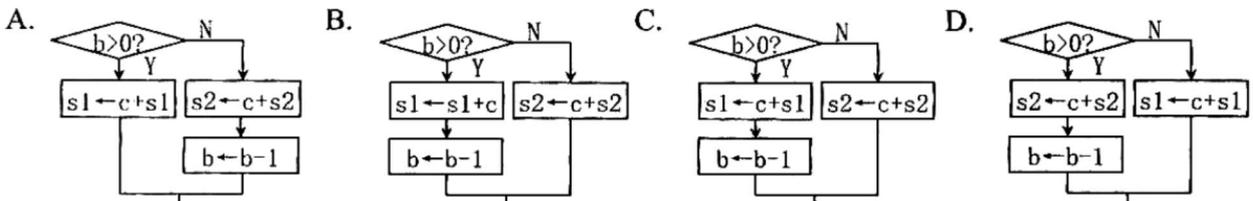
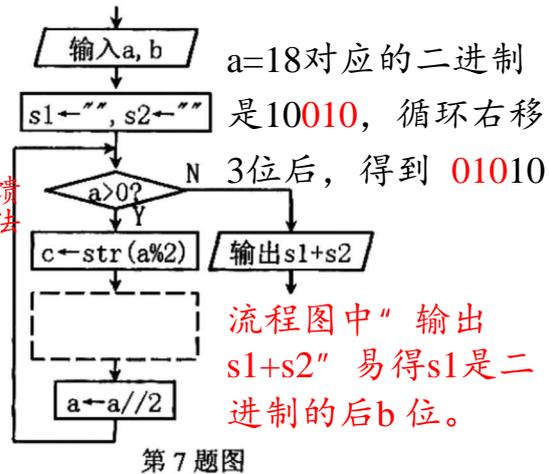
5. 下列关于网络系统的说法, 正确的是 **B**

- A. 领队手机不可通过计算机网络与服务器进行通信
- B. 检录终端须有 IP 才能与服务器进行网络通信
- C. 该网络中的资源不包括移动终端、检录终端等硬件设备
- D. 服务器与检录终端之间的通信一定会经过网关

6. 下列关于人工智能的说法, 不正确的是 **B**

- A. 符号主义人工智能源于逻辑和推理
- B. 强化学习效果与反馈的获取难度及学习数据量无关
- C. 人机共驾是混合增强人工智能的一种应用
- D. 人脸识别可以通过基于数据驱动的人工智能实现

7. 将十进制数 a 转换为二进制并循环右移 b 位的部分流程图如第 7 题图所示, 如 a 为 18, b 为 3 时, 输出结果为“01010”。虚线框中的内容是 **C**

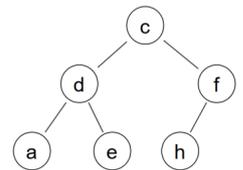


8. 栈初始为空, 经过一系列入栈、出栈操作后, 栈又为空。若元素入栈的顺序为“红”“黄”“蓝”“宝”“黑”, 其中“宝”第一个出栈, 则下列说法不正确的是 **C**

- A. “黄”一定比“红”先出栈
- B. 最后一个出栈的元素可能是“黑”
- C. 可能的出栈序列个数为 3 ⁴
- D. “蓝”可能第三个出栈

9. 某二叉树的后序遍历序列为 ae??fc, 中序遍历序列为 adechf, 则下列说法正确的是 **D**

- A. 节点 e 和节点 h 是兄弟节点
- B. 该二叉树有 2 个叶子节点
- C. 该二叉树的前序遍历序列为 cdfaeh
- D. 该二叉树是完全二叉树



10. 有如下 Python 程序段:

```
b = input(); a = "2345"; s = ""; j = -1
for i in range(len(a)):
    j = j + 1
    if j == len(b):
        j = 0
    c = 9 - int(a[i] + b[j]) % 3
    s = str(c) + s
```

s: 7897 倒序前: 7987
 9-c: 2012
 a:2345
 a[i]%3:2012
 b[j]%3:0000

运行该程序段后, 变量 s 的值为“7897”, 则变量 b 不可能的输入是 **B**

- A.3
- B.12
- C.300
- D.9306

11. 小明编写程序实现数据升序功能, 部分 Python 程序如下:

```
def bsort(arr):
    if len(arr) == 1:
        return arr
```

`pnext=a[p][1]` #用变量pnext记录p的下一个节点,因p节点要合并到hnew中,p合并后要转到这里
`a[p][1]=hnew` #p的指针域指向hnew的头节点,合并到新链表中
`hnew=p` #更新新链表的头指针指向p
`p=pnext` #让p指向下一个要处理的节点 pnext

for i in range(len(arr) - 1):

if arr[i] > arr[i + 1]:

arr[i], arr[i + 1] = arr[i + 1], arr[i]

return [arr[0]] + bsort(arr[1:])

从前往后进行的升序排序,每次先排好最后一位

提取最前面位数字连接

此程序存在问题,适合作为测试数据的是 D

A. [4,6,5,4]

B. [4,3,5,6]

C. [4,4,6,5]

D. [5,6,4,7]

12. 使用列表a和d模拟2个链表结构(节点数均大于0),头指针分别为ha和hd,每个节点包含数据区域和指针区域,如第12题图a所示。两个链表各节点均已按数据区域中数值升序排列,且 $a[ha][0] < d[hd][0]$ 。现要将d合并到a中形成降序链表结构,如第12题图b所示。实现上述功能的Python程序段如下,方框中应填入的正确代码为 A

`p = ha = 1; q = hd = 2`

`hnew = ha` #新链表的头指针

`p = a[p][1]`

`a[hnew][1] = -1`

while p != -1 and q != -1:

if a[p][0] < d[q][0]:

`a[p][1]`

else:

`pnext`

`a.append([d[q][0], hnew])`

`hnew = len(a) - 1`

`q = d[q][1]`

#继续将未并完链表的剩余节点合并进降序链表中,代码略。

A. `pnext = a[p][1]`

B. `a[p][1] = a[hnew][1]`

C. `a[p][1] = hnew`

D. `pnext = a[p][1]`

`a[p][1] = hnew`

`a[hnew][1] = p`

`hnew = p`

`hnew = p`

`hnew = p`

`p = a[p][1]`

`p = a[p][1]`

`a[p][1] = hnew`

`p = pnext`

`p = pnext`

	数据区域	指针区域
0	10	2
1	7	0
2	16	4
3	20	-1
4	17	3

	数据区域	指针区域
0	15	-1
1	11	4
2	8	3
3	9	1
4	14	0

第12题图a

	数据区域	指针区域
0	10	6
1	7	-1
2	16	9
3	20	4
4	17	2
5	8	1
6	9	5
7	11	0
8	14	7
9	15	8

第12题图b

a链节点逆序,
b链节点做插入

实现a链逆序

b节点插入

hnew p

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. 多项赛事需要电视台直播,每场比赛直播时长相同,且不超过120分钟,赛程如第13题图所示。为确保每场比赛都能被实时直播,小明编写程序,实现输入赛事直播时长,计算并输出所需的最少电视台数量。

(1) 分析第13题图所示的数据,若各赛事直播时长为90分钟,直播8月9日的赛事需要的电视台数量至少为 ▲ 3。

(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。

def trans_time(t):

#将时间t转换成分钟为单位的整数,如“01:30”转换成90,代码略

"""读取文件数据依次存入列表lst中。

lst[i]包含4个数据项,依次为日期、时间、赛事类型、赛事名称,数据已按日期为主要关键字,时间为次要关键字升序排序,代码略。"""

time = int(input("输入赛事直播的持续时长(分钟): "))

```
max_len = 0; i = 0
```

```
for j in range(1, len(lst)):
```

```
    t2 = trans_time(lst[j][1])
```

```
    while i != j:
```

```
        t1 = trans_time(lst[i][1])
```

```
        if lst[i][0] != lst[j][0]:
```

```
            t1 -= 24 * 60
```

```
        if t1 + time > t2:
```

```
            break
```

```
        i += 1
```

```
    if j - i + 1 > max_len:
```

```
        max_len = j - i + 1
```

```
print("至少需要电视台数量: ", max_len)
```

减去一天的时间，说明不是同一天

08月08日	23:15	排球, 女子铜牌赛
08月09日	01:30	花样游泳, 双人自由自选
08月09日	01:30	田径, 女子标枪/男/女子4×100米接力等决赛
08月09日	02:00	霹雳舞, 男子1/4决赛-决赛
08月09日	03:00	沙滩排球, 男子铜牌赛/决赛
08月09日	03:30	篮球, 男子决赛
08月09日	07:30	举重, 女子81公斤级决赛
08月09日	07:30	篮球, 女子铜牌赛
08月09日	09:00	排球, 女子金牌赛
08月09日	20:00	水球, 男子决赛
08月09日	21:30	篮球, 女子金牌赛
08月10日	03:00	/, 闭幕式

或 lst[i][0] < lst[j][0] 第 13 题图

14. 小张为叔叔的鲫鱼养殖鱼塘搭建了水体溶解氧含量监测系统。该系统的智能终端获取传感器数据，通过无线通信方式将数据传输到 Web 服务器，服务器将水体溶解氧含量传感器采集的数据存储到数据库，并与阈值进行判断后通过智能终端控制氧气泵。用户通过浏览器查询实时和历史数据。浙考神墙750

(1) 用户通过浏览器查询数据时，访问了该系统的 B (单选，填字母：A. 智能终端 / B. 数据库 / C. 氧气泵控制程序)。

(2) 下列关于该系统中服务器和客户端的说法，正确的是 B (单选，填字母)
A. 搭建该系统需要编写服务器和客户端程序

B. 数据在服务器与客户端之间双向传输

C. 不能在服务器端访问该系统

(3) 小张发现数据库中近期存在大量连续低于阈值下限值的水体溶解氧含量数据记录，不利于鲫鱼生存，可能的原因是 AD (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. 氧气泵制氧能力不足

B. 客户端数据处理异常

C. 智能终端与服务器通信中断

D. 水体溶解氧含量传感器故障

(4) 请写出两种可提升鱼塘水体溶解氧含量检测精准性的方法。选择精度高的溶解氧传感器、增加监测点、提高检测频率、合理放置传感器位置

(5) 小张导出 2024 年 7 月氧含量检测数据，部分数据如第 14 题图 a 所示(氧含量单位为“mg/L”)，氧含量值低于 5 则为异常。现要统计该月异常次数多于 2 次的日期中异常次数最多的 3 日，若存在符合条件的日子，则建立如第 14 题图 b 所示的图表，否则输出“无异常数多于 2 次的日子”。实现该功能的部分 Python 程序如下：

日期	时间	氧含量
2024-07-01	00:00:00	6.7
2024-07-01	01:00:00	6.6
2024-07-01	02:00:00	6.8
2024-07-01	03:00:00	6.7
2024-07-01	04:00:00	6.7
2024-07-01	05:00:00	6.2
2024-07-01	06:00:00	6.6
2024-07-01	07:00:00	6.6
2024-07-01	08:00:00	6.8
2024-07-01	09:00:00	6.5

第 14 题图 a



第 14 题图 b

dfg=df[df['氧含量']<5]

	日期	时间	氧含量
0	2024-07-02	14:00:00	4.3
1	2024-07-02	17:00:00	3.7
2	2024-07-03	03:00:00	3.7
		
n	2024-07-31	00:00:00	3.8

dfg=dfg.groupby("日期",as_index=False)["氧含量"].count()

	日期	氧含量
0	2024-07-02	2
1	2024-07-03	13
2	2024-07-04	15
	
n	2024-07-31	1

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

dfg=dfg[dfg["氧含量"]>2]



if len(dfg) > 0:

```
plt.bar(dfg["日期"],dfg["氧含量"] 或 dfg.日期,dfg.氧含量)
plt.title("2024年7月氧含量异常数最多的3日")
plt.show()
```

	日期	氧含量
1	2024-07-03	13
2	2024-07-04	15
3	2024-07-05	5
	

else:

```
print("无异常数多于2次的日子")
```

BADEC 或BAEDC 或 BAECD

①方框中应填入的语句依次为 ▲ (注: 全选、填字母序列, 次序错不得分)。

	日期	氧含量
1	2024-07-04	15
2	2024-07-03	13
3	2024-07-05	5

- A. dfg = dfg.groupby("日期", as_index=False)["氧含量"].count()
- B. dfg = df[df["氧含量"]<5]
- C. dfg = dfg.head(3)
- D. dfg = dfg[dfg["氧含量"]>2]
- E. dfg = dfg.sort_values("氧含量", ascending=False)

②请在程序中划线处填入合适的代码。

15. 某大项目由 m 个小项目 (编号为 1~m) 构成, n 个员工 (编号为 1~n) 参与该大项目。大项目完成后员工需汇报, 汇报员工参与的项目须涵盖所有小项目, 每个员工想知道自己至少要和其他哪些员工合作才可以完成大项目汇报。每个员工参与小项目情况用 [a,b,c] 表示, 若 a≤b, 则表示 c 员工参与了 [a,b] 区间的小项目, 若 a>b, 则表示 c 员工参与了 [a,m] 和 [1,b] 区间的小项目 (一个员工参与的项目区间不会被另一员工参与的项目区间包含)。如 4 人参与 8 个小项目的情况为 [3, 5, 1], [8, 3, 2], [7, 1, 3], [5, 7, 4], 则 1 号员工至少需要和 2 号、4 号员工合作完成大项目汇报, 3 号员工则需和所有员工合作才能完成大项目的汇报。编写程序, 求每个员工为完成大项目汇报所需的最少员工编号组合。

请回答下列问题:

(1) 若 5 人参与完成 10 个小项目的情况为 [2, 4, 1], [6, 10, 2], [5, 9, 3], [8, 3, 4], [3, 7, 5], 则 2 号员工至少需要和 4,5 (填员工编号) 合作才能完成大项目的汇报。

(2) 定义如下 GetNext 函数, 参数列表 x 由 n 个元素组成, 满足 x[i][0] ≤ x[i][1] (0 ≤ i < n) 且 x[i][0] < x[i+1][0] 且 x[i][1] < x[i+1][1] (0 ≤ i < n-1)。

def GetNext(n, x):

```
    j = 0; gn = []
```

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

```
        while j < n and x[i][1] + 1 >= x[j][0]:
```

#遍历i为起点满足交叉或连续区间的最后一个区间的索引

```
            j = j + 1
```

```
        gn.append(j - 1) #记录以区间i为起点能连续重叠区间的最后一个区间的索引
```

```
    return gn
```

若 x 为 [[2,4,1],[3,7,5],[4,8,2], [6,9,3], [8,13,6], [10,15,4]], n 为 6, 调用 GetNext(n, x) 后, 则 gn[1] 的值为 4。

gn=[2,4,4,5,5,5]

(3) 实现功能的部分 Python 程序如下, 请在划线处填入合适的代码。

```

def bubble_sort(a):
    n = len(a)
    loc = [-1] * (n + 1)
    for i in range(n - 1):          #以项目区间左端点值为关键词升序排序
        for j in range(n - 1, i, -1):
            if a[j][0] < a[j - 1][0]:
                a[j], a[j - 1] = a[j - 1], a[j]
        loc[a[i][2]] = i          #以员工编号为下标在 loc 数组中记录排序后的项目索引
    return loc

def group(m, n, a):
    loc = bubble_sort(a)
    for i in range(n):
        if a[i][1] < a[i][0]:
            a[i][1] += m
        a.append([a[i][0] + m, a[i][1] + m, a[i][2]])
    s = ① GetNext(2*n,a)          #注意，上述循环中的处理完成后，a 列表中元素个数为原来的两倍
    for i in range(2 * n):        #在a 的每个数据元素中添加当前元素重叠或相连区间的最后一个区间的索引，以完成后续区间的选用和跳转。
        a[i].append(s[i])
    res = [[] for i in range(n)]
    for i in range(n):
        tmp = i
        while ② a[tmp][1] < a[i][0] + m - 1:          #选用当前元素重叠或相连区间的最后一个区间，直至覆盖满 m 个项目
            tmp = a[tmp][3]
        res[i].append(tmp)
        # tmp 从当前节点往下找，当前节点的左端点 a[i][0]和tmp 节点的右端点a[tmp][1]之间相差值 (a[tmp][1]- a[i][0]+1) 未超过m，则继续找，即 tmp = a[tmp][3]，并将该节点加入 res 中
    for i in range(1, n + 1):
        ans = []
        for j in ③ res[loc[i]]:          #遍历当前员工i 所对应的在数组a 中下标值 loc[i]在res 中的值，这些值即为loc[i]下标节点需完成项目还需要的其他节点在数组a 中的下标值
            ans.append(a[j][2])          #将这些下标对应的员工编号,加入ans 结果数组中
        print(i, "号员工至少和", ans, "合作才能完成大项目的汇报")
'''

```

读取小项目数、员工数、员工数据分别存入m、n和列表a中。a[0]~a[n-1]分别存储 1~n号员工数据。a[i]包含 3 个数据项，依次为项目区间左端点值、右端点值、员工编号，代码略。

'''

group(m,n,a)