

36. 有如下 Python 程序段:

```
for i in range(1, len(data)):
    if data[i] < data[i - 1]:
        data[i], data[i - 1] = data[i - 1], data[i]
```

运行该程序段后, 列表 data 中的数据按升序排列, 则运行前的 data 可能是

- A. [1, 3, 4, 2, 5, 6] B. [3, 2, 1, 4, 5, 6] C. [1, 5, 3, 4, 2, 6] D. [2, 1, 3, 5, 4, 6]

37. 数组元素 $a[0] \sim a[n-1]$ 中数据已按升序排列, 现将 $a[p]$ ($0 \leq p < n-1$) 的值减 2, 数组 a 仍保持有序, 实现该功能的程序如下, 方框中应填入的正确代码为

```
t = a[p] - 2
i = p
while :
    a[i] = a[i - 1]:
    i -= 1
```

$a[i] = t$

- A. $i > 0$ B. $i > 0$ and $t < a[i - 1]$
 C. $i > 0$ and $a[i] < a[i - 1]$ D. $i > 0$ or $t < a[i - 1]$

38. 有如下 Python 程序段:

```
import random
q = [0, 0, 0, 0, 0]
head = tail = 3
while head != (tail + 1) % 5:
    q[tail] = random.randint(1, 9)
    if q[tail] % 2 == 0:
        tail = (tail + 1) % 5
```

执行该程序段后, q 的值可能是

- A. [4, 4, 0, 4, 4] B. [4, 8, 2, 4, 6]
 C. [4, 0, 4, 4, 6] D. [0, 0, 0, 4, 6]

39. 定义如下函数:

```
def f(x, y):
    if x >= y:
        return 1
    print("#")
    return x * f(x, y - 1) + y * f(x + 1, y)
```

执行语句 $v = f(1, 3)$ 后, 输出 “#” 的个数为

- A. 0 B. 1 C. 3 D. 7

40. 定义如下函数:

```
def f(a, n, m):
    if n == 1:
        return a
    elif n % m == 0:
        a.append(m) # 为列表 a 追加一个元素 m
        return f(a, n // m, m)
    else:
        return f(a, n, m + 1)
```

列表 a 的初始值为 [], 执行语句 $f(a, 12, 2)$ 后, a 的值为

- A. [2, 4, 6] B. [2, 2, 3] C. [2, 3, 6] D. [2, 3, 4, 6]

41. 有如下 Python 程序段：

```
def pal(s):
    if len(s) <= 1:
        return True
    elif s[0] == s[len(s) - 1]:
        return pal(s[1:len(s) - 1])
    else:
        return False
print(pal(s))
```

若 s 为 "TTQAQTT"，运行上述程序后，下列说法正确的是

- A. 程序输出结果是 False
- B. 最后一次调用 pal 时参数 s 的值是 "A"
- C. 函数 pal 被调用了 3 次
- D. 画线处的代码若改为 "len(s)==0"，pal 函数被调用的次数不变

42. 对于任意正整数 x，甲、乙两个程序段输出的结果都相同，则乙程序段方框中正确的代码为

<pre>def f(x): if x == 1: return str(x) else: return f(x // 2) + str(x % 2) print(f(x))</pre>	<pre>s = "" while x > 0: <div style="border: 1px solid black; width: 100px; height: 40px; margin: 5px 0;"></div> print(s)</pre>
甲程序段	乙程序段

- A. ①s = s + str(x % 2)
②x = x // 2
- B. ①s = str(x % 2) + s
②x = x // 2
- C. ①x = x // 2
②s = str(x % 2) + s
- D. ①x = x // 2
②s = s + str(x % 2)

43. 定义如下函数：

```
def binSearch(data, key, i, j):
    while i <= j:
        m = (i + j) // 2
        if data[m] == key:
            return m
        elif data[m] < key:
            i = m + 1
        else:
            j = m - 1
    return -1
```

若 d 为 [6, 12, 15, 18, 22, 25, 35, 46, 58, 60]，k 为 25，分别执行下列语句，函数中画线处语句的执行次数最少的是

- A. p = binSearch (d, k, 0, 6)
- B. p = binSearch (d, k, 1, 7)
- C. p = binSearch (d, k, 1, 8)
- D. p = binSearch (d, k, 3, 9)

44. 有如下 Python 程序段：

```
# 随机产生 10 个整型元素的非降序序列存入列表 a，代码略
i, j = 0, len(a) - 1 ; key = int(input()) ; b = []
while i <= j:
    m = (i + j) // 2
    b.append(a[m])
    if a[m] > key:
        j = m - 1
```

```
else:
    i = m + 1
```

```
print(b)
```

运行该程序段后，b 的结果不可能是

- A. [14, 17, 14, 16] B. [11, 5, 9, 6] C. [13, 19, 19, 19] D. [10, 5, 7, 8]

45. 有如下 Python 程序段：

```
from random import randint
i, j = 0, len(s)
st = []
while i < j:
    if randint(0, 1) == 0:
        s += s[i]
        j += 1
    else:
        if len(st) > 0 and s[i] < st[len(st) - 1]:
            st[len(st) - 1] = s[i]
        else:
            st.append(s[i])
    i += 1
```

若 s 为 "abcd"，运行该程序段后，列表 st 不可能是

- A. ["a", "c", "d"] B. ["b", "a", "d"] C. ["b", "c", "a"] D. ["c", "a", "d"]

46. 用如下 Python 程序段实现对数组元素 a [0] 到 a [9] 从小到大排序。

```
i = 0
while i < 9:
    k, i = i, 10
    for j in range(9, k, -1):
        if (1):
            a[j], a[j - 1] = a[j - 1], a[j]
            (2)
```

上述程序段中画线处的可选代码有：

- ①a [j] < a [j - 1] ②a [j] > a [j - 1] ③i = j ④i = k - 1 ⑤i = i + 1

则 (1) (2) 处代码序号依次为

- A. ①③ B. ①④ C. ②④ D. ②⑤

47. 有如下 Python 程序段：

```
import random
tag = [""] * len(data)
p = i = 0
while i < len(data):
    if tag[p] == "" and p != -1:
        tag[i] += data[p][0]
        p = data[p][1]
    else:
        tag[i] += tag[p]
        i += 1
        p = i
```

若 data 为 [{"A", 3}, {"B", -1}, {"C", 0}, {"D", 1}, {"E", 2}]，运行该程序段后，tag [4] 的值为

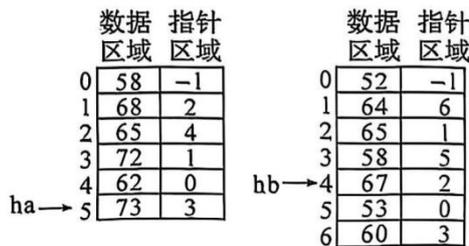
- A. "ADB" B. "CADB" C. "DB" D. "ECADB"

48. 使用列表模拟 dba 和 dbb 两个链表结构（节点数均大于 0），每个节点含数据域和指针域，两个链表的头指针分别为 ha 和 hb，链表中所有节点已按数据域降序排列，如第 48 题图所示。现要将 dbb 中数据域大于 dba 中数据域最小值的节点合并到 dba 中，并保持链表 dba 的有序性。实现该功能的程序段如下。

```

if dbb[hb][0] > dba[ha][0]:
    dba.append([dbb[hb][0], ha])
ha = len(dba) - 1
hb = dbb[hb][1]
q = ha
while hb != -1 and q != -1:
    p = dba[q][1]
    if p != -1 (1) dbb[hb][0] > dba[p][0]:
        dba.append([dbb[hb][0], p])
        dba[q][1] = len(dba) - 1
        hb = dbb[hb][1]
    (2)

```



第 48 题图

(1) 与 (2) 处可选的表达式有:①and ②or ③q = dba [q][1] ④q = p
 则 (1) 与 (2) 处的表达式序号依次为

- A. ①③ B. ①④ C. ②③ D. ②④

高三上信息技术：综合练习

49. 根据机器的负载率对工厂的 n 台机器 [编号 0~(n-1)] 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），调度负载率最高的机器休息；若有多台机器的负载率同为最高，则调度连续工作时间最长的机器中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) 若共有 10 台机器，某次采集到 0~9 号机器的负载率和连续工作的时长如第 49 题图所示，当前 7 号机器处于休息状态，则接下来需要调度休息的机器编号是 _____。

机器编号	0	1	2	3	4	5	6	7	8	9
负载率	93	85	93	93	82	85	93	0	82	93
连续工作时长/小时	5	3	6	1	4	7	2	0	2	6

第 49 题图

(2) 实现上述功能的部分 Python 程序如下，请在画线处填入合适的代码。

```

n = 10 # 共有 10 台机器
a = [0] * n # 列表 a 的长度为 n，各元素值均为 0，a[i]用于存放 i 号机器的负载率
cnt = [0] * n # cnt[i]用于存放 i 号机器的连续工作时长（单位为小时）
# 启动 1~(n-1)号机器工作，0 号机器休息，代码略
while True:
    # 延时 1 小时，采集各台机器的负载率存入 a，代码略
    mx = 0
    for i in range(n):
        if (1) _____:
            # 调度 i 号机器工作，代码略
        elif a[i] > a[mx]:
            mx = i
        elif (2) _____ and cnt[i] > cnt[mx]:
            mx = i
    for i in range(n):
        cnt[i] = cnt[i] + 1
    # 调度 mx 号机器休息，代码略
    (3) _____

```

50. 某水域水位预警系统共设立 8 个监测点，每个监测点发送水位状态数据至服务器，状态数据分为：-1（低水位），0（正常水位），1（高水位）。当超过一半的监测点发送同一高（低）水位状态数据时，则发布水位警报。编写程序，每隔 10 分钟获取各个监测点的水位状态数据，若出现高（低）水位警报，统计连续高（低）水位警报的次数并输出。

(1) 某时刻服务器收到的水位状态信息为 [0, -1, -1, 0, 0, 1, 1, 1]，则_____（填字母： A. 有/B. 没有）发布水位警报。

(2) 实现上述功能的部分 Python 程序如下，请在画线处填入合适的代码。

```
python
t = [0, 0] # 低水位、高水位连续警报的次数
while True:
    c = [0, 0]
    # 获取 8 个监测点的水位状态数据，存入 d，代码略
    for i in range(len(d)):
        if d[i] == -1:
            c[0] += 1
            ① _____:
            c[1] += 1
        for i in range(2):
            ② _____
            if c[i] > len(d) // 2:
                flag = 1
                ③ _____
                t[1 - i] = 0
                # 根据 i 值输出低（高）水位警报及连续警报的次数，代码略
                break
            if flag == 0:
                t = [0, 0]
```

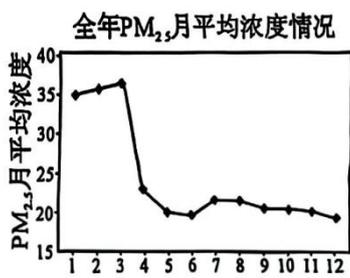
延时 10 分钟，代码略

51. 小明收集了某监测点一年的 PM2.5 数据（单位： $\mu\text{g}/\text{m}^3$ ），现要对这些监测数据进行分析，请回答下列问题：

(1) 将该监测点的数据导出，存于“data.xlsx”文件中，如第 51 题图 a 所示。现要统计出该监测点全年各月的 PM2.5 平均浓度，绘制如第 51 题图 b 所示的线形图，并统计 PM2.5 月平均浓度最高的月份中各天监测数值大于 45 的次数，选择次数最多的前 5 天由高到低输出，如第 51 题图 c 所示。实现上述功能的部分 Python 程序如下。

月	日	时	PM2.5
1	1	0	30
1	1	1	29
1	1	2	34
12	31	21	23
12	31	22	19
12	31	23	18

第 51 题图 a



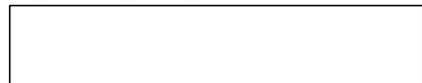
第 51 题图 b

3 月 6 日监测到 PM2.5 浓度值大于 45 的次数是：5
3 月 8 日监测到 PM2.5 浓度值大于 45 的次数是：5
3 月 9 日监测到 PM2.5 浓度值大于 45 的次数是：4
3 月 5 日监测到 PM2.5 浓度值大于 45 的次数是：3
3 月 7 日监测到 PM2.5 浓度值大于 45 的次数是：3

第 51 题图 c

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

```
df_ave = df.groupby("月", as_index=False)["PM2.5"].mean() # 分组求平均
plt.plot(①) # 设置绘图参数, 显示如第 51 题图 b 所示的线形图, 代码略
# 将 PM2.5 月平均浓度最高的月份存入 m, 代码略
df_m = df[df["月"] == m] # 筛选
df_m = df_m[df_m["PM2.5"] > 45]
```



```
# 筛选该月份中 PM2.5 大于 45 的数据
# 统计各天次数、排序并取前 5
```

删除红框内两句注释

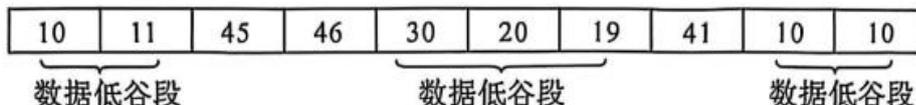
依次输出 df_res 中各天 PM2.5 浓度值大于 45 的次数, 如第 51 题图 c 所示, 代码略

①程序画线处应填入的代码_____。

②程序方框中应填入的代码是_____ (单选)

- A. df_cnt = df_m.groupby("日", as_index=False)["PM2.5"].count() # 分组计数
df_cnt = df_cnt.head(5) # 取前 5 个
df_res = df_cnt.sort_values("PM2.5", ascending=False) # 降序排序
- B. df_cnt = df_m.groupby("日", as_index=False)["PM2.5"].count()
df_res = df_cnt.sort_values("PM2.5", ascending=False).head(5)
- C. df_cnt = df_m.groupby("时", as_index=False)["PM2.5"].count()
df_res = df_cnt.sort_values("PM2.5", ascending=False).head(5)
- D. df_cnt = df_m.groupby("日", as_index=False)["PM2.5"].count()
df_res = df_m.sort_values("PM2.5", ascending=False).head(5)

(2) 一个数据低谷段有如下特点: 段内的数据均小于阈值 th, 连续小于 th 的数据同属于一个低谷段。如 th 为 35 时, 第 51 题图 d 所示的数据序列中共有 3 个数据低谷段。



第 51 题图 d

现将全年 PM2.5 日均浓度数据依次存入列表 data 中, 找出数据序列中长度大于 length 的所有低谷段, 并统计这些低谷段长度的总和。实现上述功能的部分 Python 程序如下所示, 请在画线处填入合适的代码。

读入全年 PM2.5 日均浓度数据, 存入列表 data 中; 读入阈值 th 和长度 length, 代码略

```
① _____
left = -1
for i in range(len(data)):
    if left == -1:
        if data[i] < th:
            ② _____
        elif data[i] >= th:
            n = i - left
            left = -1
            if n > length:
                cnt += n
if ③ _____ and i - left + 1 > length:
    cnt += i - left + 1
# 输出 cnt, 代码略
```

52.某区各学校五年级学生 1 分钟仰卧起坐测试成绩(个数,最高为 60 个)保存在文件 "data.xlsx" 中部分数据如第 52 题图 a 所示。现要分析各校的 A 等成绩(32 个及以上)和 C 等成绩(24 个及以下)的学生分别在本校所占的比例。请回答下列问题。

学校	班级	姓名	性别	仰卧起坐
A 校	1	郭 * 凯	女	34
A 校	1	吴 * 荣	女	30
A 校	1	张 * 志	男	30
A 校	1	王 * 雄	男	27
H 校	4	叶 * 博	男	25
H 校	4	吴 * 游	女	24
H 校	4	张 * 瑜	女	23
H 校	4	刘 * 英	女	23

第 52 题图 a

(1) 定义如下 calc(n,a,rank_A,rank_C) 函数,参数 n 表示某校的总人数;参数 a 存储的数据如第 52 题图 b 所示,例如, a[22] 值为 3,表示该校仰卧起坐成绩是 22 个的共有 3 人;参数 rank_A 存储 A 等成绩的下限,rank_C 存储 C 等成绩的上限。利用函数功能计算并返回该校 A 等成绩和 C 等成绩的学生占比。

实现该功能的部分 Python 程序如下,请在画线处填入合适的代码。

i	0	1	...	22	23	24	25	...	30	31	32	33	...	60
a[i]	0	0	...	3	7	6	11	...	9	12	8	10	...	0

第 52 题图 b

def calc(n, a, rank_A, rank_C):

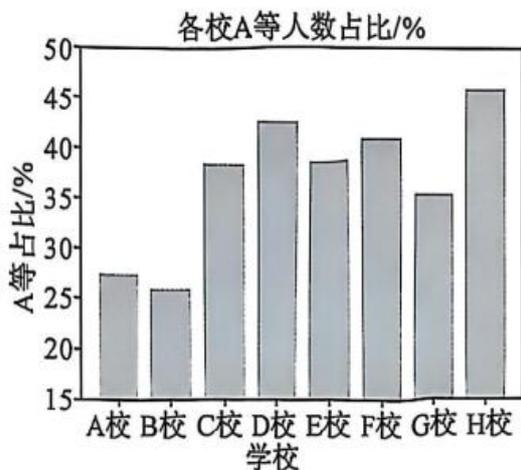
```

    ①
    j = 0
    while j < len(a):
        s += a[j]
        if j == rank_C:
            rt_C = s / n * 100
            s = 0
            j = ②
        j += 1
    rt_A = s / n * 100
    return (rt_A, rt_C) # 返回成绩为 A 等、C 等的学生占比

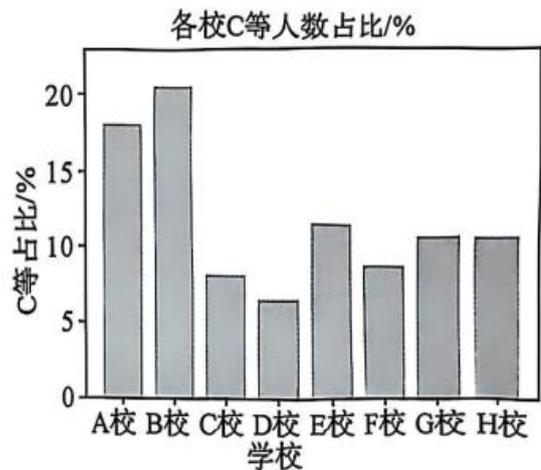
```

删除冒号

(2) 实现分析各学校 A 等成绩和 C 等成绩的学生占比的部分 Python 程序如下所示,请在画线处填入合适的代码。



第 52 题图 c



第 52 题图 d

```

df = pd.read_excel("data.xlsx")
school = ["A 校", "B 校", "C 校", "D 校", "E 校", "F 校", "G 校", "H 校"]
m = len(school) # 学校数量

```

```

b1, b2 = [], [] # b1 存储各校 A 等占比, b2 存储各校 C 等占比
for i in range(m):
    df1 = df[df["学校"] == _____ ①] # 筛选当前学校数据
    # 统计生成如第 52 题图 b 所示列表 a 的数据, 代码略
    # 调用 calc(len(df1), a, 32, 24)函数并将返回值分别添加到列表 b1、b2, 代码略
x = _____ ② # 设置 x 轴数据 (学校名称)
plt.bar(x, b1) # 如第 52 题图 c 所示图
# 设置绘图参数并显示图表, 代码略
plt.bar(x, b2) # 如第 52 题图 d 所示图
# 设置绘图参数并显示图表, 代码略
    
```

(3) 分析图 c 和图 d, 发现 A 等学生占比最低和 C 等学生占比最高的为同一学校, 该校是 _____。

53. 某研究小组要搭建档案馆环境监测与控制系统。该系统的温度、湿度传感器每隔一定时间采集档案馆内的温度、湿度数据, 智能终端通过无线通信方式将传感器数据传输至服务器, 服务器将数据存储到数据库中, 根据预设的阈值进行判断, 并通过智能终端控制执行器实现温度和湿度控制。用户可通过浏览器查询实时和历史数据。请回答下列问题:

(1) 下列功能由智能终端实现的是 _____

- A. 读取温度传感器数据
- B. 将温度数据写入数据库

(2) 该系统用 Flask Web 框架编写的程序 _____

- A. 全部部署在服务器端
- B. 全部部署在智能终端
- C. 部分部署在服务器端, 部分部署在智能终端
- D. 不需要部署, 直接运行

(3) 要统计出相邻两次采集的温度数据均大于 26°C 的次数, 部分流程图如第 53 题图所示, 图中①②③处可选表达式有:

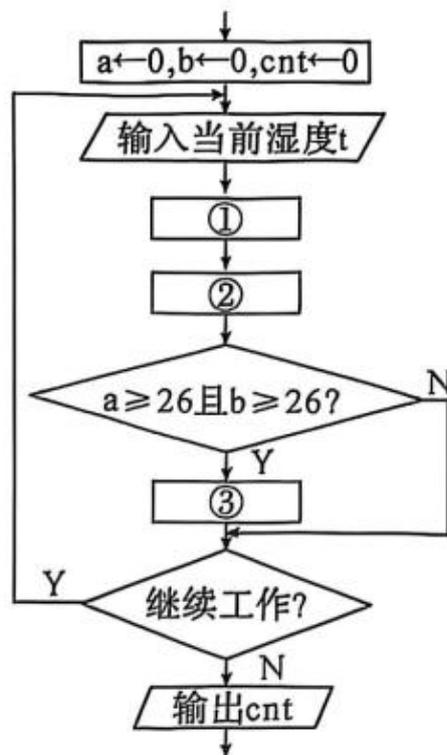
- A. $a \leftarrow b$
- B. $a \leftarrow t$
- C. $cnt \leftarrow cnt + 1$
- D. $b \leftarrow t$

则 ① ② ③ 处应填入的表达式序号依次为 _____ (选填字母组合序列)

(4) (多选) 下列关于该系统的设计, 合理的有 _____

- A. 为每个传感器分别配备智能终端
- B. 为智能终端增加响应用户浏览历史数据请求的功能
- C. 不同档案室设置不同的温度阈值范围
- D. 通过声光、短信等多种预警形式通知管理人员

(5) 系统运行一段时间后, 发现执行器频繁启停, 请从数据或软件的角度, 分析可能的原因: _____。

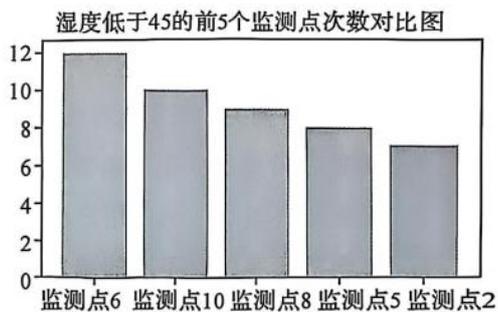


54.某小组搭建温、湿度监测系统，模拟蔬菜大棚的温、湿度监测。各监测点每 10 分钟采集 1 次数据，智能终端通过 IoT 模块将数据传输到服务器。服务器监测到某监测点数据异常时，自动向管理员发送预警信息，并通过智能终端控制相应监测点的执行器工作。管理员通过浏览器查看各监测点的数据，并通过服务器向智能终端发送执行器的控制指令。请回答下列问题：

- (1) 关于该系统中智能终端与传感器连接的说法，正确的是_____。
- A. 不同类型的传感器必须连接不同的智能终端
 - B. 智能终端可以同时连接多个不同类型的传感器
 - C. 一个智能终端只能连接一个传感器
 - D. 传感器不能直接连接智能终端，需通过服务器中转
- (2) 采集传感器数据的程序在_____中执行（单选：A. 服务器 B. 智能终端 C. 浏览器 D. 执行器）
- (3)（多选）下列关于系统故障的说法，正确的有_____
- A. 若服务器故障，则智能终端无法继续读取传感器数据
 - B. 若智能终端故障，则管理员无法通过浏览器查看实时数据
 - C. 若 IoT 模块故障，则管理员无法远程控制执行器
 - D. 若执行器故障，则传感器将无法采集温、湿度数据
- (4) 当发现服务器收到的某监测点的温度数据与实际温度不符时，可能的原因是_____。
- (5) 将系统中某天记录的温度和湿度数据导出并存于文件 "data.xlsx" 中，部分数据如第 54 题图 a 所示。已知该类大棚的正常湿度为 45~65，现要统计各监测点湿度低于 45 的次数，并绘制次数最多的 5 个监测点的对比柱形图，如第 54 题图 b 所示。

监测点	日期	时间	类型	数值
监测点1	20250501	00:00	温度	16
监测点1	20250501	00:00	湿度	58
监测点2	20250501	00:00	温度	14
监测点2	20250501	00:00	湿度	36
监测点3	20250501	00:00	温度	16
监测点9	20250501	23:50	湿度	60
监测点10	20250501	23:50	温度	20
监测点10	20250501	23:50	湿度	57

第 54 题图 a

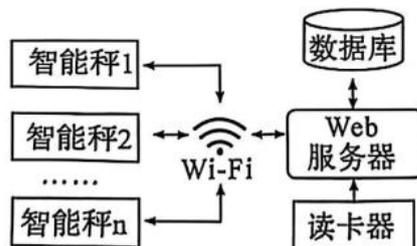


第 54 题图 b

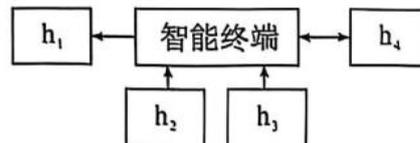
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel("data.xlsx")
df1 = _____ ①
df1 = _____ ②
print(len(df1)) # 输出记录中湿度低于 45 的总次数
# 统计各监测点的湿度异常数据次数
df1 = _____ ③
df1 = _____ ④
plt.bar(df1["监测点"], df1["数值"])
# 设置绘图参数，显示第 54 题图 b 所示的柱形图，代码略
程序中①②③④处可选的代码有：
A. df[df["数值"]<45]
B. df[df["类型"]=="湿度"]
C. df1[df1["类型"]=="湿度"]
D. df1.sort_values("湿度", ascending=False)
E. df1.groupby("监测点", as_index=False).数值.count()
F. df1.sort_values("数值", ascending=False)
G. df1.groupby("监测点", as_index=False).湿度.count()
```

55. 某校食堂自助就餐系统的结构示意图如第 55 题图 a 所示。点菜过程为：

- ①就餐者在读卡器上刷校园卡，领取餐盘；
- ②就餐者将餐盘放在某智能秤的餐盘感应区上，智能秤获取该餐盘编号并将编号发送给服务器；
- ③就餐者从该智能秤上面的菜品容器中取适量菜品，智能秤计算本菜品金额，实时输出到显示屏，同时将相关数据传输、存储到服务器数据库中，并完成结算；
- ④管理人员通过浏览器查询系统相关数据。



第 55 题图 a



第 55 题图 b

(1) 本系统中智能秤的主要硬件构成除了智能终端，还有①IoT 模块；②显示屏；③压力传感器；④餐盘感应器。第 55 题图 b 中 h₁、h₂、h₃、h₄ 处的硬件序号依次为 ()

(单选，填字母：A. ②③④①/B. ①③④②/C. ③①②④/D. ③②①④)

(2) (多选) 下列关于该系统的软件开发的说法，正确的有 ()

- A. 实现从传感器获取重量数据并计算金额的程序存储在智能秤中
- B. 智能秤与服务器之间的数据传输是单向传输
- C. 就餐者的历史消费数据应存储在系统的数据库中
- D. 浏览器查询数据库中数据的功能与服务器端程序无关

(3) 系统正常运行一段时间，发现某个菜品被取用后，智能秤显示屏的数据无变化，请写出两种可能造成上述问题的原因：_____。

(4) 将系统中某月的学生消费数据导出并保存到文件“stu.xlsx”中，数据包含了每一次就餐的学生 ID、结算时间和消费金额。现要找出文件中该月消费总金额最低的 10 位学生，部分 Python 程序如下：

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

输出 df1 中该月消费总金额最低的 10 位学生，代码略

方框中要填入的正确代码依次为_____ (选两项，填序号)

- ① df1=df.sort_values("消费金额",ascending=True)
- ② df1=df1.sort_values("消费金额",ascending=True)
- ③ df1=df.groupby("学生 ID",as_index=False).sum()
- ④ df1=df1.groupby("学生 ID",as_index=False).sum()

56. 某景点提供自行车租借服务 (总数为 m 辆，先到先得)。若干游客团队将陆续到达景点，若到达时景点剩余自行车的数量能够满足该团队的需求，则该团队租用自行车并在离开时归还，否则放弃租用。请回答下列问题：

(1) 若该景点共有 30 辆自行车，某天到达该景点的各团队情况如下图所示，则成功租借自行车的团队数量为_____。

团队编号	A001	A002	A003	A004	A005
到达时间	9:00	9:30	10:00	10:30	10:45
离开时间	10:30	11:00	12:00	12:00	12:20
需自行车数	15	12	8	20	18

(2) 列表 `data` 存储了各团队的到达情况,每个元素的前两个数据项分别表示团队编号和到达时间。定义函数实现功能:根据各团队的到达情况,按到达时间升序排列,同一时间到达的,团队编号较小的排在前面。小明编写了如下所示函数。

```
def dsort(lst):
    for i in range(len(lst)-1):
        k = i
        for j in range(i+1, len(lst)):
            if lst[k][1] > lst[j][1] :
                k = j
        if k > i:
            lst[i], lst[k] = lst[k], lst[i]
```

①若调用 `dsort (data)` 能够实现上述功能,则排序前 `data` 一定符合的特征是:同时到达的团队,

- A. 排在前面的编号大,且在连续相邻的位置 B. 排在前面的编号大,可在不相邻的位置
C. 排在前面的编号小,且在连续相邻的位置 D. 排在前面的编号小,可在不相邻的位置

②若排序前 `data` 不符合上述特征,仍需使 `dsort (data)` 实现上述排序功能,则可将画线处的代码修改为 `lst [k][1] > lst [j][1] or`_____。

(3) 根据各团队停留时间及其所需自行车的数量,判断其是否能够成功租借自行车。实现该功能的函数如下所示,请在画线处填入合适的代码。

参数 `data` 列表存放已按到达时间升序排序的团队数据,其中每个元素 `data[i]` 包含 5 个数据项,`data[i][0]~data[i][4]` 依次为:团队编号、到达时间、离开时间、所需自行车数、能否成功,到达、离开时间都用 8 位数字字符串表示,如“07280830”表示 7 月 28 日 8 点 30 分,“能否成功”数据项用于存放各团队能否成功租用自行车。`m` 存放用于租借的自行车总数。

```
def proc(data, m):
    b = [];t = 0
    ① _____
    for i in range(len(data)):
        n = t + data[i][3]
        pt = p
        while pt != -1 and b[pt][0] <= data[i][1]:
            ② _____
            pt = b[pt][2]
        if n > m:
            data[i][4] = False
        else:
            p = pt
            prev = -1
            while pt != -1 and b[pt][0] < data[i][2]:
                prev = pt
                pt = b[pt][2]
            b.append([data[i][2], data[i][3], pt]) # 列表 b 末尾添加元素
            if prev == -1:
                p = len(b) - 1
            else:
                b[prev][2] = len(b) - 1
            ③ _____
            data[i][4] = True
```

57. 某市举办科技竞赛，已有预赛成绩，计划从 n 所学校选拔 tot 名选手参加决赛。选拔规则是：各校排名前 st 名的选手直接入选；各校排名 $st+1$ 至 ed 名的选手进入备选池，剩余名额按预赛成绩由高到低从备选池中挑选，成绩相同的选手一并入选，一旦人数达到或超过 tot ，选拔结束。现给定所有选手的预赛成绩，每名选手的数据包括学校编号 $[0 \sim (n-1)]$ 、选手编号、成绩，计算各选手的校内排名。将所有学校的数据合并到一起并按成绩由高到低排序，生成汇总表，如图所示。再按上述规则选拔，按汇总表的顺序输出入选选手编号。

(1) 如下图所示，若 n, tot, st, ed 的值依次为 3, 8, 1, 4，则实际入选决赛的选手数是_____。

学校编号	1	1	1	0	0	1	1	0	0	2	2
选手编号	1052	1073	1049	45	75	1079	1029	6	60	2028	2022
成绩	98	93	92	91	91	90	90	90	86	86	85
校内排名	1	2	3	1	1	4	4	3	4	1	2

(2) 校内排名的计算方法是：若选手所在学校有 m 个选手成绩高于该选手，则该选手的校内排名为 $m+1$ 。实现校内排名的函数如下所示。

```
def ranking(sdata):
    a = [0] * 101 # 满分 100 分，统计各分数人数
    b = [0] * 101 # 存储各分数对应的排名
    for i in range(len(sdata)):
        score = sdata[i][2]
        a[score] += 1
    mc = 1
    for i in range(100, -1, -1):
        if a[i] != 0:
            b[i], mc = mc, mc + a[i]
    for i in range(len(sdata)):
        sdata[i][3] = b[sdata[i][2]]
```

①若将方框处代码改为语句： $b[i], mc = mc, mc+a[i]$ ，则排名结果将_____（填字母：A. 不变/B. 改变）

②若将画线处代码改为： $range(len(sdata)-1, -1, -1)$ ，则排名结果将_____（填字母 A. 不变/B. 改变）

(3) 实现选拔的部分 Python 程序如下所示，请在画线处填入合适的代码。

```
def proc(data, tot, st, ed):
    hs=[-1, -1]
    ts=[-1, -1]
    cnt = 0
    for i in range(len(data)):
        if data[i][3] <= ed:
            k = 1
            ① _____
            cnt += 1
            k = 0
            if hs[k] == -1:
                hs[k] = i
            else:
                ② _____
            ts[k] = i
    p, q = hs[0], hs[1]
```

```

while cnt < tot and q != -1:
    cj = data[q][2]
    while q != -1 and cj == data[q][2]:
        while p != -1 and p < q:
            pre = p
            p = data[p][4]
        cnt += 1
        tmp = data[q][4]
        data[q][4] = p
        ③
        pre = q
        q = tmp
p = hs[0]
while p != -1: # 输出入选选手编号
    print(data[p][1], end='')
    p = data[p][4]
data = []
# 读取 n, tot, st, ed 的值, 代码略
for i in range(n):
    ''' 读取 i 学校的选手成绩数据, 存入 sdata, 每个元素形如 [2, '2050', 84, 5, -1], 前四项依次为
    学校编号、选手编号、成绩、校内名次, 代码略'''
    ← ranking(sdata)
    ← # 将 sdata 合并到 data, 代码略
    # 将 data 中数据按照成绩由高到低排序, 代码略
proc(data, tot, st, ed)
    
```

缩进调整

58. 某数据序列 data 中的元素均为小于 127 的正整数。现要对 data 进行加密处理，过程分为“变换”和“重排”两步。

“变换”处理方法是使用指定的 n 组序列 R_0, R_1, \dots, R_{n-1} 依次对 data 进行变换得到“变换后序列”。利用 R_i 对 data 进行变换的过程是：在 data 中查找所有与 R_i 相同的子序列，将找到的每个子序列中的元素值加上 R_i 的长度值 L_i ，并在各子序列前插入一个标记元素（值为 $127+L_i$ ），这些子序列及标记元素不再参与后续的变换。

对“变换后序列”进行“重排”的方法为：先以 m (分组长度, $2 \leq m \leq 6$) 个元素为 1 组，将 data 中的元素从前往后分成若干组，不足 m 个元素的组末尾用 0 补足，再将各组元素进行组内逆序排列，得到“重排后序列”，即为密文。

例如用于变换的两组序列为 [5, 1]、[3, 8, 7]，m 为 4，对“原始序列”进行变换与重排的结果如下图所示。

原始序列	3	5	1	6	3	8	7	5	1	8	7					
变换后序列	3	129	7	3	6	130	6	11	10	129	7	3	8	7		
重排后序列	3	7	129	3	11	6	130	6	3	129	10	7	8	7	0	0

现要对密文进行解密，过程是：先将“重排后序列”恢复为“变换后序列”，再将“变换后序列”恢复为“原始序列”。请回答下列问题。

- 若变换后序列为 [4, 129, 10, 11, 14, 129, 7, 11, 11, 130, 12, 13, 11, 12, 129, 7, 11, 3, 10, 130, 12, 13, 11, 5]，则用于变换处理的序列组数为_____。
- 将“重排后序列”恢复为“变换后序列”的函数如下所示，代码有误，应改正为_____

59. 某学校棋友社组织五子棋在线对弈活动。活动期间，对弈平台自动为“等待”状态的会员匹配对手，匹配规则：按胜率从高到低两两一组安排对弈（若胜率相同，则按报名号从小到大安排对弈）。请回答下列问题：

(1) 某时刻在线会员信息如下图所示，按上述规则匹配对手，与会员“清风”对弈的会员是_____

会员名	月光	墨韵	烟雨	落樱	星辰	明月	竹影	清风
报名号	101	82	232	10	54	88	23	21
胜率 /%	50	60.5	65	48	50	55	60	50
状态	等待	等待	等待	等待	对弈	对弈	等待	等待

(2) 有如下函数，用于对“等待”状态的会员仅按胜率从高到低进行排序。参数 data 列表用于存储“等待”状态的会员信息，其中每个元素包含 3 个数据项，依次为会员名、报名号、胜率。

```
def match(data):
    i, m = 0, len(data)
    flag = True
    while i < m and flag:
        flag = False
        for j in range(m-1, i, -1):
            if data[j-1][2] < data[j][2]:
                data[j], data[j-1] = data[j-1], data[j]
                flag = True
        i += 1
    return data
```

① 若列表 data 的值为[["墨韵", 82, 60.5], ["清风", 21, 50], ["竹影", 23, 60], ["月光", 101, 50], ["落樱", 10, 48], ["烟雨", 232, 65]], 则语句“data[j], data[j-1] = data[j-1], data[j]”的执行次数为_____。

② 现要将该函数的排序规则改为“按胜率从高到低排序;若胜率相同,则按报名号从小到大排序”,那么程序中加框处的条件表达式应改为 data[j-1][2] < data[j][2] or _____。

(3) 定义如下函数,功能是判断玩家落子后是否赢得棋局,玩家落子后,在“水平,垂直、左上到右下的对角线,右上到左下的对角线”任意方向五子连线即为胜利,返回“True”,否则返回“False”。参数 board 列表存储棋盘状态(0 表示未落子,1 表示黑子,2 表示白子),player 表示玩家(1 表示黑子玩家,2 表示白子玩家),x 和 y 表示当前玩家落子的位置。

```
def check_win(board, player, x, y):  
    directions = [[0,1], [1,0], [1,1], [1,-1]]  
    for item in directions:  
        ① _____  
        for k in ② _____ :  
            step = 1  
            count = 0  
            while True:  
                nx = x + item[0] * step * k  
                ny = y + item[1] * step * k  
                if 0 <= nx < len(board) and 0 <= ny < len(board) and ③ _____:  
                    count += 1  
                    step += 1  
                else:  
                    break  
            if count >= 5:  
                return True  
    return False
```