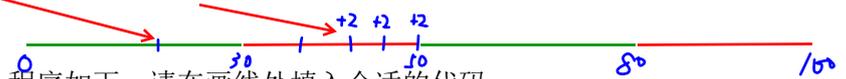


## 高三上信息专项练习

1. 一条直行车道在某路口有红绿灯指示，绿灯时长为  $m$  秒，红灯时长为  $n$  秒（例如，第  $0 \sim m-1$  秒为绿灯，第  $m \sim m+n-1$  秒为红灯），车辆到达该路口时，遇到绿灯可以直接通过，遇到红灯则需排队等待，红灯变绿后，排队车辆依次通行，不在队首的车辆比排在其前面的车辆晚 2 秒通过。假设初始时间为第 0 秒，初始状态为绿灯且路面为空，请编写程序，根据一段时间内车辆到达路口的时间，计算每辆车实际通过该路口的时间。

(1) 若该路口绿灯时长为 30 秒，红灯时长为 20 秒，初始时间为第 0 秒，初始状态为绿灯且路面为空，车辆到达时间分别为第 22, 35, 37, 38, 50 秒，则最后一辆车通过该路口的时间为第 56 秒。



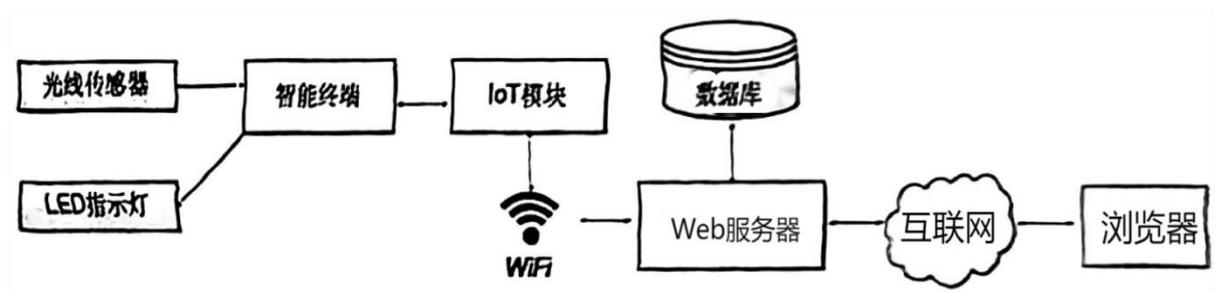
(2) 实现上述功能的部分 Python 程序如下，请在画线处填入合适的代码。

勘误

```
def gtm(t,p): # p 为单次绿灯时长和红灯时长之和
    # 函数功能：计算第 t 秒之前最后一次绿灯开始的时间
    c=t%p # p 为周期,
    return (t-c)
# 读入绿灯时长、红灯时长及所有车辆到达时间，分别存入 m, n 和列表 a
# 代码略
b = []; t = 0; p = m+n
flag = False # flag 记录车辆到达时是否需要排队
for i in a: 依次遍历车辆到达时间
    if t <= i: 算法1：总体
        t=i
    else:
        flag = True
        c = t % p
        if c >= m: 判断是否处于红灯时间
            t = gtm(t+n, p) 算下一次绿灯的时间
            flag = True
        b.append(t)
        if flag:
            t = t+2
print(b)
```

算法1:  $t$  1秒1个循环  
 算法2: 时间跳跃到每辆车的到达时间 ✓  
 $t$ : 上一辆车的实际出发时间

2. 小华同学要搭建教室环境监测系统，该系统的结构示意图如图 a 所示。该系统通过光线传感器监测教室亮度，发生异常时 LED 指示灯发出警报；师生可以通过浏览器查看实时监测结果和历史数据。请回答下列问题。



(1) 该系统中的执行器设备是 LED指示灯。

(2) 在该系统进行软件开发时，小华 B (单选，填字母：A. 必须 / B. 无需) 编写客户端程序。 **B/S模式**

(3) 小华基于 Flask Web 框架编写服务器端的程序，部分代码如下。访问该系统网站首页时，执行的视图函数是 index()。

# 导入 Python 框架模块及其他相关模块

```
app = Flask(__name__)
```

```
@app.route("/")
```

```
def index():
```

```
    # 从数据库中查询所有记录，生成返回页面，代码略
```

```
if __name__ == "__main__":
```

```
    app.run(host="192.168.1.58", port=8080)
```

服务器IP地址，不是智能终端IP地址

(4) 小华将系统中某天 6 时~22 时的亮度数据导出，部分数据如图 b 所示 (时间格式 “时:分:秒”)。分析每小时的最小亮度值，线形图如图 c 所示，部分 Python 程序如下:

时间,亮度  
06:00:01,434  
06:00:02,434  
06:00:03,433  
06:00:04,434  
06:00:05,434  
06:00:06,434  
06:00:07,434

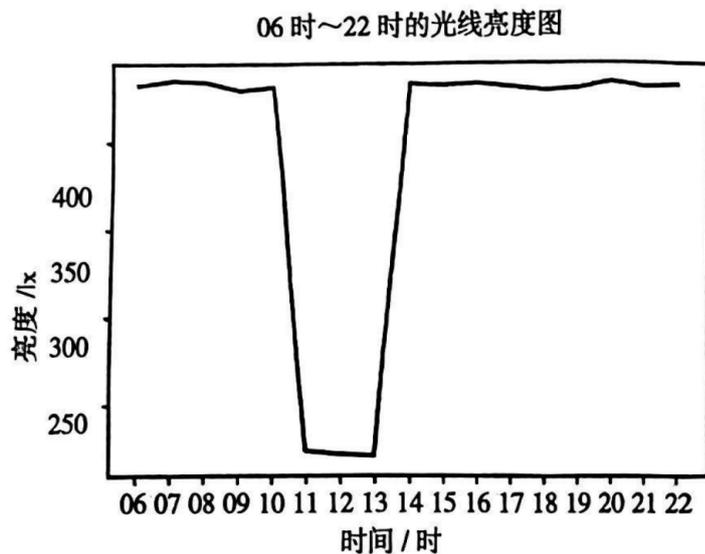


图 b

图 c

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("data.csv") # 读取数据文件
df["小时"] = ""
for i in df.index:
    # 根据行和列标签选取单个值
    df.at[i, "小时"] = df.at[i, "时间"][0:2]
df = df.groupby("小时", as_index=False).min() # 按小时分组
plt.plot(_____, df["亮度"]) df.小时 / df["小时"]
plt.show() # 设置绘图参数，显示如第 26 题图 c 所示线形图，代码略
```

①请在画线处填入合适的代码。

②现要对光照亮度值小于 val 的数据行进行删除处理，编写如下 Python 函数：

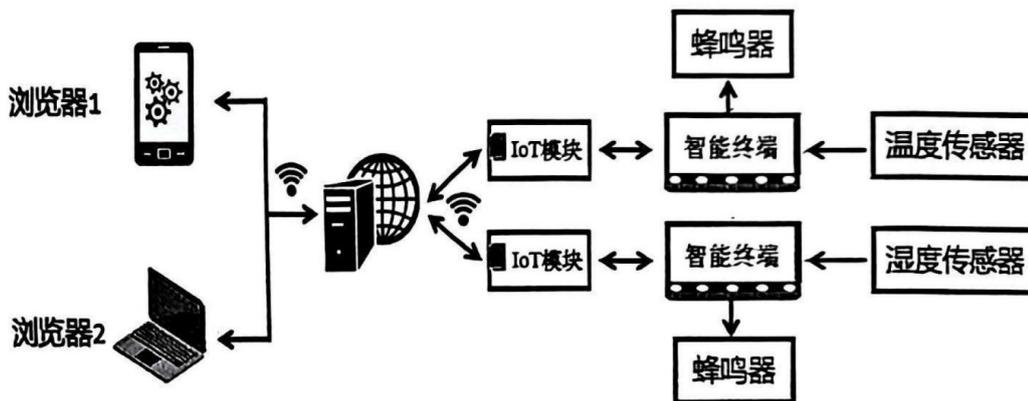
```
def del_data(df, val):
    (1) C. df=df.sort_values("亮度", ascending=False) 排序
    (2) A. n=len(df[df["亮度"] >= val])                取前n个数据
    (3) B. df=df.head(n)
    return df
```

上述程序段中(1) (2) (3)方框处可选语句为：

- A. n=len(df[df["亮度"] >= val])
- B. df=df.head(n)
- C. df=df.sort\_values("亮度", ascending=False)

则(1)至(3)的语句依次为 **CAB 或ACB** (填字母 A 、 B 、 C 的组合序列)

3. 小李为了监测图书馆内的湿度和噪声数据，模拟搭建了一个图书馆环境监测系统，系统结构示意图如图所示。该系统主要功能是：通过湿度传感器和声音传感器分别采集图书馆的湿度和噪声，再通过 IoT 模块将采集的数据实时传送到 Web 服务器中，服务器端实现对湿度和噪声数据的接收、存储、处理、统计等信息管理。客户端只能通过浏览器访问服务器。执行器实现湿度或噪声信号示警。



(1) “图书馆环境监测系统” 中的声音传感器属于 **A** (单选，填字母：A. 输入设备 / B. 输出设备)。

(2) 如果该系统的无线网络连接不成功，结果会 **C** (单选，填字母：A. 系统内的所有模块之间都不能传递数据 / B. 传感器无法采集数据 / C. 服务器无法控制执行器)。

(3) 使用基于 Python 的 Flask Web 框架编写 Web 应用后，启动 Web 应用的代码为：

```
library.run(host="192.168.0.1", port=5000, debug=False)
```

则该 Web 服务器的 IP 地址为 **192.168.0.1**。

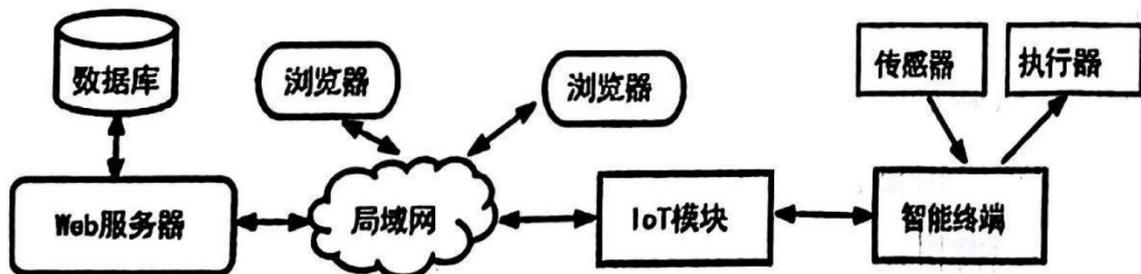
(4) 当图书馆内的湿度超过一定阈值时，会影响书籍的保质；当噪声高于某一阈值时，会影响学习者的学习效果和身心健康。为此可以设计预警机制，在客户端呈现示警信息，并通过智能终端控制执行器。实现上述预警机制的算法伪代码描述如下：

```
# 设示警信息用 2 位数字表示，第 1 位表示噪声，第 2 位表示湿度
# 设湿度的最高阈值为 maxh，噪声的最高阈值为 maxn
```

```
# 设某一时刻采集到图书馆内的湿度为 h，噪声为 n
if h < maxh and n < maxn:
    示警信息="00"
elif h > maxh and n < maxn:
    示警信息="01"
    蜂鸣器响 1 次
elif h < maxh and n > maxn:
    示警信息="10"
    蜂鸣器响 2 次
else:
    示警信息="11"
    蜂鸣器响 3 次
```

**勘误** 若某一段时间内收到的示警信息为 “101010101010”，则表示 B（单选，填字母：A. 湿度超出阈值，蜂鸣器响了 6 次 / B. 噪声超出阈值，蜂鸣器响了 12 次）。

4. 小华搭建温控系统模拟游泳馆恒温水池温控管理，实现以下功能：系统自动采集水温数据并存入数据库，系统根据水温设定阈值自动调节控制；游泳馆工作人员可利用手机、计算机等设备使用浏览器通过局域网登录管理系统，进行查看当前和历史水温、设置水温阈值等操作。系统结构如图所示，请回答下列问题。



(1) 实现 “查看当前和历史水温、设置水温阈值等功能” 的程序，属于该系统的 B（单选，填字母：A. 智能终端程序 / B. 服务器端程序）。

(2) 下列关于该系统传感和控制的说法，正确的是 B（单选，填字母）。

- A. 游泳馆员工使用的手机是该系统的执行器
- B. 可通过多个传感器采集水温数据来提高水温准确度
- C. 智能终端无需将采集到的数据传输给 Web 服务器

(3) 下列关于该系统的说法，正确的是 BD（多选，填字母）。（注：全部选对得 2 分，选对但不全的得 1 分，不选或选错的得 0 分）

- A. 工作人员无法使用平板电脑登录该系统
- B. 无需自行开发支撑该系统运行的所有软件
- C. 数据库中所有数据都需要存储在同一数据表中 **存同一个数据库中，但可以存在不同的表中**
- D. 存储采集的水温数值时，可以同步存储该水温数值的采集时间

(4) 系统按如下方式自动控制水温：系统每分钟采集并存储一次水温，若连续 10 次采集的水温均低于设定值，则启动加热器；加热过程中，若连续 10 次采集到的温度值均高于设定值，则停止加热。实现该功能的部分代码如下所示，请在画线处填入合适的代码。

```

# 导入相关库；获取系统设置的温度阈值存入 h(若获取失败，h 赋值为 28)，代码略
d = [0]*10
s, k = 0, 0
while True:
    # 读取当前传感器采集到的温度值 t，将 t 传输并存储到服务器，代码略
    k = (k+1)%10
    d[k] = 0
    if t < h:
        ① d[k]=1
    he = sum(d) # 计算列表 d 中数据之和
    if he == 10 and s == 0:
        s = 1 温度低为 1
        # 启动升温，代码略
    if he==0 and s==1
        s = 0
        # 停止升温，代码略
    sleep(1000*60) # 等待 1 分钟
    
```

(5) 若系统运行一段时间后，IoT 模块突发故障导致无法工作，请结合第 4 题图所示的系统结构和上述代码，描述对系统工作造成的一种影响。

5. 小华收集了各高校在某省招生报考选考要求的部分数据，数据已经按“地区”列进行排序，如图 a 所示。其中“选考要求”列指报考该校某专业要求的选考科目，用逗号隔开的多个科目表示需同时要求，空白代表没有任何选考要求。请回答下列问题。

地区	代码	学校	专业	选考要求
0 北京	10003	清华大学	计算机科学与技术	物理，化学
1 北京	10003	清华大学	信息管理	物理
2 北京	10003	清华大学	遗传工程	物理，化学，生物
3 北京	10003	清华大学	考古	历史，地理
4 南京	10288	南京理工大学	化学工艺	化学
.....				
50235 云南	11393	昆明学院	园林	化学，生物
50236 云南	10678	昆明医科大学	助产学	生物
50237 云南	10678	昆明医科大学	物流管理	
50238 云南	10673	云南大学	护理学	生物
50239 云南	10673	云南大学	矿产探测	物理，地理

图 a

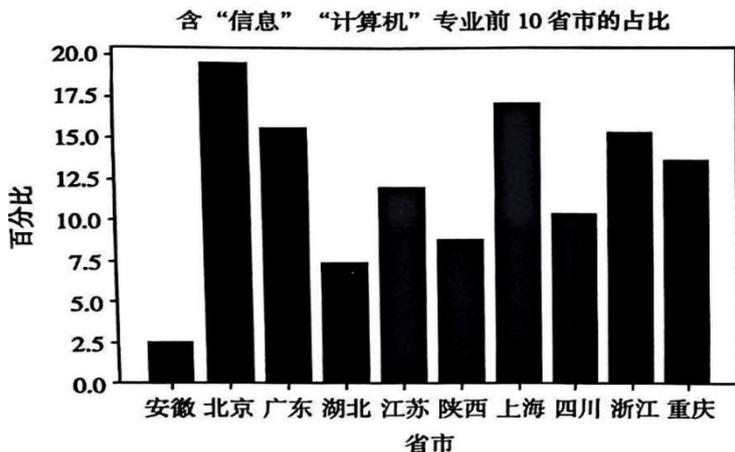


图 b

(1) 小华初步设想选考物理和化学，在没有确定第三科的情况下，为了统计已确定能报考的大学专业总数，需要基于如图 a 所示的 DataFrame 对象 df 进行逐行遍历，如果该行“选考要求”列的值满足条件，则大学专业的总数增加 1，下列描述中，能作为判断条件的是 A (单选，填字母)。

- ① “选考要求” 列的值为空字符串
  - ② “选考要求” 列的值为 “物理” 或 “化学” 其中一个学科名称
  - ③ “选考要求” 列的值仅同时包含 “物理” 和 “化学” 两个学科名称
  - ④ “选考要求” 列的值包含三个学科名称，其中含有 “物理” 和 “化学”
- A. ①或②或③      B. ①或②或④      C. ①或③或④      D. ②或③或④

(2) 小华想统计不同地区专业名称中含 “信息” “计算机” 的专业百分比。部分 Python 程序如下，请在画线处填入合适的代码。

```
import pandas as pd
import matplotlib.pyplot as plt
# 读取相关数据，保存在如图 a 所示的 DataFrame 对象 df 中，代码略
n, i, j = len(df), 0, 0
num = {} # 用字典存储各地区的占比
while i < n:
    s = df.at[i, "地区"]
    cnt = 0
    while j < n and ① df.at[j, "地区"] == s:
        m = df.at[j, "专业"]
        if ("信息" in m) or ("计算机" in m):
            cnt = cnt + 1
        j = j + 1
    total = ② j-i # j到下一个地区
    num[s] = round(cnt/total*100, 2)
    i = j
```

前提：数据已按地区排序

地区	代码	学校	专业	选考要求
0 北京	10003	清华大学	计算机科学与技术	物理, 化学
1 北京	10003	清华大学	信息管理	物理
2 北京	10003	清华大学	遗传工程	物理, 化学, 生物
3 北京	10003	清华大学	考古	历史, 地理
4 南京	10288	南京理工大学	化学工艺	化学
.....				
50235 云南	11393	昆明学院	园林	化学, 生物
50236 云南	10678	昆明医科大学	助产学	生物
50237 云南	10678	昆明医科大学	物流管理	
50238 云南	10673	云南大学	护理学	生物
50239 云南	10673	云南大学	矿产探测	物理, 地理

```
df1 = pd.DataFrame({"省份":num.keys(),"占比":num.values()})
plt.bar(df1["省份"], df1["占比"]) # 设置绘图参数，显示如图 b 所示，代码略
```

6. 某图书馆有  $m$  个借阅窗口, 编号分别用  $1 \sim m$  表示, 各窗口处理一本图书所需的时长分别用  $v[0] \sim v[m-1]$  表示(时间单位均为秒)。现有  $n$  位读者借书, 先到的先处理, 用列表 `data` 中的元素按时间先后顺序依次记录每个人到达的时间和借书的数量。轮到某位读者借书时, 他会选择当前能最快借到书且处理一本书的速度最快的窗口。编写程序模拟上述过程, 并统计每个窗口处理的读者位数和图书数目。请回答下列问题。

(1) 若只有 2 个窗口, 1 号窗口每处理一本书所需时长为 50, 2 号窗口每处理一本书所需时长为 30, 则下表中读者完成借阅处理后, 1 号窗口处理的图书数目为 3。

到达时间	借阅图书数目
20	3
70	2
135	2
150	1
210	3

(2) 定义如下 `wosrt(v)` 函数, 参数  $v$  为按编号次序各窗口处理一本书的所需时长, 函数按所需时长对各窗口升序排列, 返回对应的窗口编号和所需时长。

```
def wosrt(v):
    m = len(v)
    print(m)
    vm = []
    for i in range(0, m):
        vm.append([i+1, v[i]])
    for i in range(1, m): 冒泡排序 按vm[1]升序
        for j in range(m-2, i-2, -1):
            if vm[j][1] > vm[j+1][1]:
                vm[j], vm[j+1] = vm[j+1], vm[j]
    return vm
```

勘误

① 若  $v$  为  $[20, 50, 30, 35, 5]$ , 则返回值 `vm` 中第一个元素为  $[5, 15]$ 。

② 若函数中 `i` 语句的条件 “`vm[j][1]>vm[j+1][1]`” 误写为 “`vm[j][0]>vm[j+1][0]`”,  $v$  为  $[20, 50, 30, 35, 15]$  时, 执行 `print(wosrt(v))` 函数后, 输出的是 A (单选, 填字母)。

- A.  $[[1, 20], [2, 50], [3, 30], [4, 35], [5, 15]]$
- B.  $[1, 15], [2, 20], [3, 30], [4, 35], [5, 50]]$
- C.  $[5, 20], [1, 50], [3, 30], [4, 35], [2, 15]]$
- D.  $[5, 15], [1, 20], [3, 30], [4, 35], [2, 50]]$

(3) 实现模拟过程并进行计算的部分 Python 程序如下, 请在画线处填入合适的代码。

```

def proc(data, vm):
    n = len(data) # 人数
    m = len(vm) # 窗口数
    wind = [0]*m # 窗口变为空闲时的时间
    servnum = []
    for i in range(m):
        servnum.append([0, 0, 0])
    i = 0
    for i in range(n):
        t = data[i][0]
        k = 0
        while k < m and wind[vm[k][0]-1] > t:
            k += 1
        if k == m:
            wt = wind[vm[0][0]-1]
            k = 0
            for j in range(1, m):
                if wind[vm[j][0]-1] < wt:
                    wt = wind[vm[j][0]-1]
                    k = j
            t = wt
        wn = vm[k][0]
        if servnum[wn-1][0] == 0:
            servnum[wn-1][0] = wn
        servnum[wn-1][1] += 1
        servnum[wn-1][2] += data[i][1]
        wind[wn-1] = t + data[i][1] * vm[k][1]
    return servnum

# 读取读者数据存入列表 data, 每个元素包含各读者到达时间和借阅图书数目两个数据项
# 读取每个窗口处理一本书所需的时长数据, 存入列表 v 中
# 代码略
vm = wsort(v)
print(proc(data, vm)) # 输出每个元素包含的窗口编号、处理的读者位数和借阅图书数目三个数据项
    
```

主程序

vm = wsort(v), vm中存[[窗口编号, 处理时长],

从排序最前面的窗口开始找

找空闲窗口

找不到空闲窗口

找最早空闲的窗口

对应窗口学生数+1

servnum[wn-1][2] += data[i][1] 对应窗口借阅数+i号学生借阅数