



## 区块练 1 算法程序实现的综合应用

(核心考向:编程思维、控制结构、解析算法、枚举算法的综合应用)

1. **真题** 某监控设备可定时采集红绿灯信号灯状态数据,数据格式记为[a,b],其中 a、b 分别为红灯和绿灯的状态值,0 表示灯灭,1 表示灯亮,如[0,1]表示红灯灭、绿灯亮。

现要编写程序,每隔 1 秒采集并检测信号灯是否存在如下异常状态:第一类,红绿灯同亮或同灭;第二类,红灯或绿灯超时,即保持同一状态时长大于上限值(如 300 秒)。检测到异常状态就发送相应信息。请回答下列问题:

- (1)若检测到“红绿灯同亮”异常,则采集到的数据是\_\_\_\_\_ (单选,填字母)。

A.[0,0] B.[0,1] C.[1,0] D.[1,1]

- (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。

```
#设置信号灯保持同一状态时长上限值
tlimit=300
pre=[-1,-1]
"""t[0]、t[1]分别记录红灯、绿灯保持同一状态的时长"""
t=[0,0]
while True:
    · "接收一次采集到的状态数据,存入 d,代码略"
    · if _____ ①:
        · · if d[0]==1:
            · · · #发送“红绿灯同亮”信息,代码略
        · · else:
            · · · #发送“红绿灯同灭”信息,代码略
    · for i in _____ ②:
        · · if d[i]==pre[i]:
```

```
        · · · t[i]+=1
        · · · if _____ ③:
            · · · · if i==0:
                · · · · · "发送“红灯超时”信息,
                    代码略"
            · · · · else:
                · · · · · "发送“绿灯超时”信息,
                    代码略"
        · · else:
            · · · t[i]=1
    · pre=d
    · #延时 1 秒,代码略
```

2. **真题** 服务器根据温度数据生成状态码,满足一定条件时发送给智能终端以控制风扇运行。生成和发送状态码的规则如下:

- ①温度 $\leq 15^{\circ}\text{C}$ 时,状态码为 0;温度 $\geq 35^{\circ}\text{C}$ 时,状态码为 2;其他温度范围,状态码为 1。  
②当状态码连续  $k(k>1)$  个相同,且与最近已发送状态码不同时,则发送该状态码。

请回答下列问题:

- (1)若  $k$  为 3,最近已发送状态码为 1,随后生成的状态码序列为“0,0,0,2,0,0,1,1,1,0”,则由该序列触发的状态码发送次数为\_\_\_\_\_次。

- (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。

```
sent=last=-1
#获取 k 值,代码略
while True:
    · #获取温度数据保存到变量 t 中,代码略
    · _____ ①
```

```

    • if t>=35:
    •   • code=2
    • elif t>15:
    •   • code=1
    • if _____ ②:
    •   • if code!=last:
    •     • cnt=1
    •   • else:
    •     • _____ ③
    •     • if cnt==k
    •       • #发送状态码 code,代码略
    •       • sent=code
    • last=code
    • #延时 1 分钟,代码略

```

3.[2026 台州模拟]有一条从西向东延伸、长度为  $m$  公里的直线型公路,沿线有  $n$  个适合建设基站的点位,点位的位置以其距离公路最西端的距离表示。每个基站的信号覆盖半径为  $r$  公里,即选择位置为  $x$  的点位建设基站,信号可覆盖  $x-r$  到  $x+r$  范围的公路。现要选择部分点位建设基站,使整段公路(从 0 到  $m$ )被信号覆盖。编写程序,根据点位位置,输出最少需要建设的基站数量;若无法实现整段公路的信号覆盖,则输出  $-1$ 。请回答下列问题:

(1)若公路长度为 10 公里,每个基站的信号覆盖半径为 3 公里,有 3 个适合建设基站的点位,位置分别为 2、7、9,则最少需要安装的基站数量为\_\_\_\_\_。

(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。

"""输入公路长度  $m$ 、基站信号覆盖半径  $r$ 、点位数量  $n$ ,代码略"""

"""读取  $n$  个点位的位置,升序排序后存入列表  $a$ ,代码略"""

ans,i=0,0

\_\_\_\_\_ ①

```

while pos<m:
    • d=-1
    • while i<n and a[i]-r<=pos:
    •   • _____ ②
    •   • i=i+1
    • if d== -1:
    •   • break
    • pos=d+r
    • ans=ans+1
if _____ ③:
    • print(ans)
else:
    • print("-1")

```

4.[2026 宁波模拟]空调作为现代家庭的必备家电,从部件加工到组装完成需经过多道工序。每个工序可能有多个不重复的前置工序,且必须完成其所有前置工序后才能开始当前工序,若无前置工序则可直接开始,同一时间只能执行一个工序。现在已知每个工序完成所需的时间(单位:分钟,简称工时),需编写程序模拟这一流程,根据输入的工序信息(包含工序编号、工时及其前置工序列表),计算完成指定编号为  $m$  的工序所需的总工时(含所有前置工序用时及当前工序用时)。

(1)若工序信息如图所示,完成工序 2 需先完成前置工序 0 和 1,总工时为  $(3+7+5=15)$ ,则完成工序 5 的总工时为\_\_\_\_\_。

工序编号	工时	前置工序
0	3	1
1	7	无
2	5	0
3	4	无
4	2	3
5	5	2、4
6	8	无
7	6	6
8	5	5、7

(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 cal(hours,pre,m):
    · need=[m]
    · i=0
    · while i<len(need):
        · · if need[i] in pre:
            · · · for j in pre[need[i]]:
                · · · · need.append(j)
            · · · _____ ①
        · total=0
        · for i in need:
            · · _____ ②
        · return total
```

#hours[i]存储工序 i 的工时

hours=[3,7,5,4,2,5,8,6,5]

"""键值对 2:[0,1]表示工序 2 的前置工序为  
工序 0 和工序 1"""

pre={0:[1],2:[0],4:[3],5:[2,4],7:[6],8:[5,7]}

m=int(input("请输入工序编号:"))

total\_time=\_\_\_\_\_ ③

print("完成该工序及其全部前置工序所需  
总工时为:",total\_time)

**5. 现真题** 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)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 #机器台数
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 _____ ①:
            · · · #调度 i 号机器休息,代码略
            · · · cnt+=1
        · · if a[i]>a[t]:
            · · · _____ ②
        · · elif a[i]==0:
            · · · #调度 i 号机器工作,代码略
    · if _____ ③:
        · · #调度 t 号机器休息,代码略
```

**6.**小陈用多个共阴极 LED 数码管实现天数倒计时效果。单个数码管的七个引脚分别标记为 a、b、c、d、e、f、g(如图 a 所示),显示的字符和十六进制字形码如图 b 所示。如显示字符“3”,需将 a、b、c、d、g 对应的数码管引脚设置为高电平(用数字“1”表示),其他引脚设置为低电平(用数字“0”表示),效果如图 c 所示。

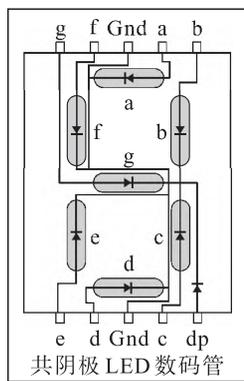


图 a

共阴极 LED 数码管										
dp	Gnd	g	f	e	d	c	b	a	字形码	字符
0	0	0	1	1	1	1	1	1	3FH	0
0	0	0	0	0	0	1	1	0	06H	1
0	0	1	0	1	1	0	1	1	5BH	2
0	0	1	0	0	1	1	1	1	4FH	3
0	0	1	1	0	0	1	1	0	66H	4
0	0	1	1	0	1	1	0	1	6DH	5
0	0	1	1	1	1	1	0	1	7DH	6
0	0	0	0	0	0	1	1	1	07H	7
0	0	1	1	1	1	1	1	1	7FH	8
0	0	1	1	0	1	1	1	1	6FH	9

图 b

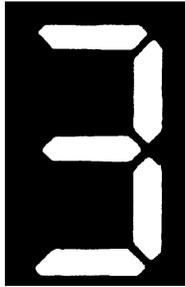


图 c

请回答下列问题：

(1) 若显示数字“25”，则对应的十六进制字形码为\_\_\_\_\_。

(2) 如下的 days 函数用于计算 y 年 m 月 d 日到元年元旦的总天数，即当年天数与过去所有平年和闰年的天数之和。已知闰年的年份是 4 的倍数且不是 100 的倍数，或者是 400 的倍数，且闰年有 366 天，平年有 365 天。请在划线处填入合适的代码。

```
def days(y,m,d):
    • mdays=[0, 31, 59, 90, 120, 151, 181, 212,
              243,273,304,334]
    • ans=y*365+y//4-y//100+y//400
    • ans=ans+mdays[m-1]+d
    • if (y%4==0 and y%100!=0 or y%400==0)
        and m>=3:
    • • _____
    • return ans
```

(3) 如下的 single 函数用于求单个七段数码管各引脚的电平值，请在划线处填入合适的代码。

def single(num):

```
• code={0:"3F",1:"06",2:"5B",3:"4F",4:
        "66",5:"6D",6:"7D",7:"07",8:
        "7F",9:"6F"}
• b=""
• for i in range(1,-1,-1):
• • if "0"<=code[num][i]<="9":
• • • c=int(code[num][i])
• • • else:
• • • c=_____
• • • for i in range(4):
• • • b=str(c%2)+b
• • • c//=2
• return b[1:]
```

(4) 如下的主程序用于实现天数倒计时效果，请在划线处填入合适的代码。

```
import datetime
import time
while True:
    • "获取开幕之前日期的年月日并将其
      转成 8 位字符串形式，如"20240528"
    • n=datetime.datetime.now().strftime("%
      Y%m%d")
    • y=int(n[:4])
    • m=int(n[4:6])
    • d=int(n[6:])
    • total=days(2024,7,26)-days(y,m,d)
    • ans=""
    • if total==0:
    • • break
    • while _____:
    • • num=total%10
    • • ans=single(num)+ans
    • • total=total//10
    • print(ans) #输出各引脚的电平值
    • #硬件相关的通信部分程序，代码略
    • time.sleep(1)
```