

# QG训练营嵌入式组第三周周记：

## 生活随记

### 动态规划

#### 几个简单的概念

##### 【无后效性】

一旦 $f(n)$ 确定，“我们如何凑出 $f(n)$ ”就再也用不着了。

要求出 $f(15)$ ，只需要知道 $f(14), f(10), f(4)$ 的值，而 $f(14), f(10), f(4)$ 是如何算出来的，对之后的问题没有影响。

**“未来与过去无关”，这就是无后效性。**

(严格定义：如果给定某一阶段的状态，则在这一阶段以后过程的发展不受这阶段以前各段状态的影响。)

##### 【最优子结构】

回顾我们对 $f(n)$ 的定义：我们记“凑出 $n$ 所需的**最少**钞票数量”为 $f(n)$ 。

$f(n)$ 的定义就已经蕴含了“最优”。利用 $w=14, 10, 4$ 的**最优解**，我们即可算出 $w=15$ 的**最优解**。

大问题的**最优解**可以由小问题的**最优解**推出，这个性质叫做“最优子结构性质”。

引入这两个概念之后，我们如何判断一个问题能否使用DP解决呢？

**能将大问题拆成几个小问题，且满足无后效性、最优子结构性质。**

转自<https://www.zhihu.com/question/23995189>

```
/*
 *计算n最少可由多少个1，5，11组成
 */

/*
typedef struct stytyle
{
    int times;//记录最少组成
    int ones;//记录1的个数
    int fives;//记录5的个数
    int levens;//记录11的个数
}stytyle
*/
int f[100], i, n, cost;
scanf("%d", &n);
f[0] = 0;
for(i=1; i<=n; i++)
```

```

{
    cost = INF; //INF无穷大
    if(i-1>=0) cost = min(cost, f[i-1]+1)
    if(i-5>=0) cost = min(cost, f[i-5]+1)
    if(i-11>=0) cost = min(cost, f[i-11]+1);
    f[i]=cost;
    printf("f[%d]=%d\n", i, f[i]);
}

```

## 流水灯实验

```

#include<reg52.h>
#include<intrins.h> //库中含有循环移位操作
#define uint unsigned int
#define uchar unsigned char
unsigned int a;
uchar led;
void delay()
{
    uint x,y;
    for(x=100;x>0;x--)
    {
        for(y=250;y>0;y--) //AT89C52 大约1ms时间
        {
            ;
        }
    }
}
void main()
{
    led = 0x11; //0001 0001
    while(1)
    {
        P0 = led;
        delay(); //延迟0.1s
        led = _crol_(led,1); //循环左移一位
    }
}

```

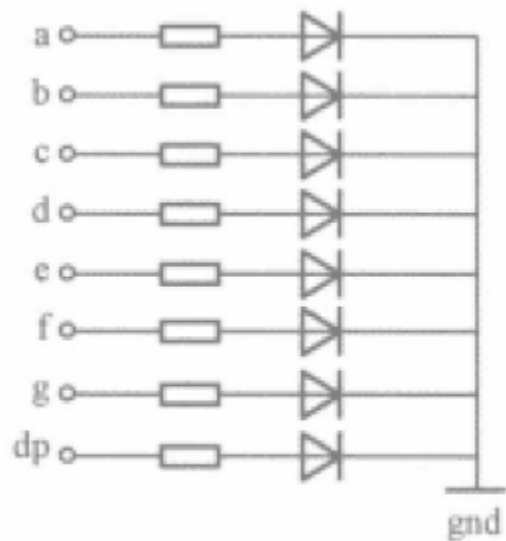
## 数码管

数码管分**共阳极**和**共阴极**两种：

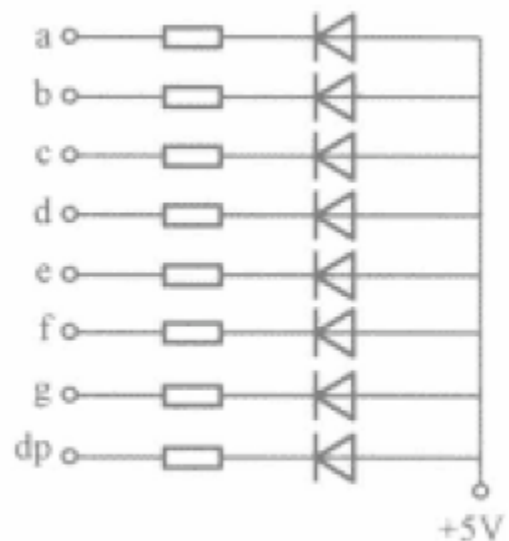
使用共阴极时，通常把阴极接地（0电位），则在使用数码管时只需要给对应发光二极管高电位即可（产生电势差）。

使用共阳极时，数码管内部发光二极管点亮时需要5mA以上的电流，而且电流过大时会烧坏二极管，由于单片机I/O口无法输出如此大的电流，所以数码管与单片机连接时需要加**驱动电路**，可以用**上拉电阻**的方法或使用专门的**数码管驱动芯片**。

数码管有**单位一体**和**多位一体**，多位一体时，内部公共端独立，而负责显示数字的线段连接在一起，独立的公共端可以显示那一位显亮，而连接在一起的线段可以控制点亮数码管的数字，一般把公共端叫做“**位选线**”，连接在一起的线段叫做“**段选线**”。



(b) 共阴极



(c) 共阳极

数字	16进制
0	0x3f
1	0x06
2	0x5b
3	0x4f
4	0x66
5	0x6d
6	0x7d
7	0x07
8	0x7f
9	0x6f
A	0x77
b	0x7c
C	0x39
d	0x5c
E	0x79
F	0x71

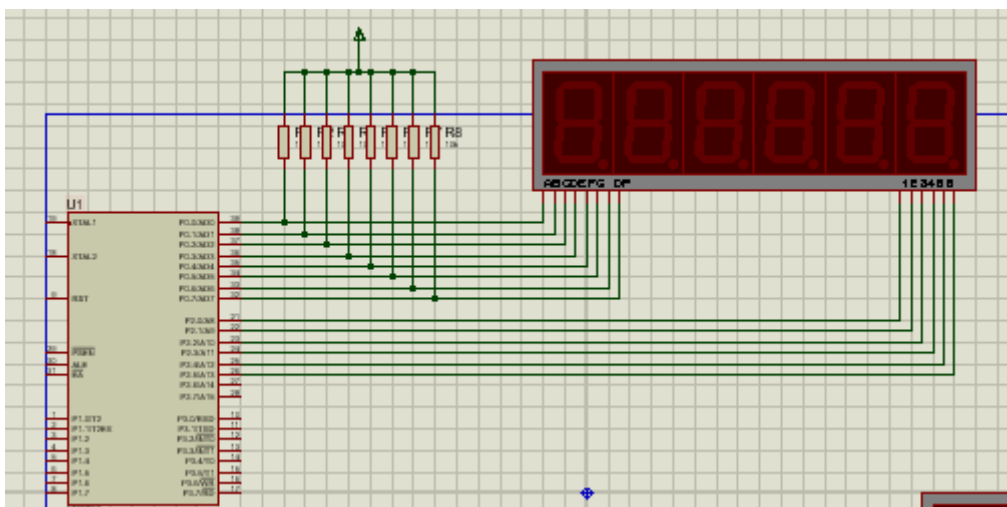
```
#include<reg52.h>
#include<intrins.h> //使用库函数内的循环移位
#define uint unsigned int
#define uchar unsigned char //宏定义方便后续函数的定义
void delay()
{
```

```

uint x,y;
for(x=100;x>0;x--)
    for(y=250;y>0;y--)    ;
} //延迟函数 AT89C52 大约100ms时间
void main()
{
    uint temp,i,count;
    /*code是编码的意思*/
    uchar code num[10] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};

    /*
    *6位数数码管 P0端接段选线 P2端接位选线
    *以下电路图未用73HC573为例
    若使用73HC573则需要将锁存器的锁存端与P2.6,P2.7相连
    通过对P2.6,P2.7的控制实现位选端和段选端的打开与关闭以
    实现对应位数显示对应数字的功能
    */
    temp = 0xdf;
    count = 0;
    while(1)
    {
        for(i=0;i<10;i++)
        {
            P0 = num[i];
            P2 = temp;
            count++;
            if(count>=6)
            {
                count = 0;
                temp = 0xdf;
            }
            else temp = _crol_(temp,-1); //循环右移一位
            delay();
        }
    }
}

```



## 一周总结

·C语言中可以实现空指针实现泛型。

·在使用空指针时应先开辟动态内存空间或在必要时用其他类型指针进行赋值（需记录此时数据类型以确定输出格式）

·空指针在函数中直接赋值存在问题：

- 1.利用函数中变量直接赋值时，退出该函数时变量可能被释放；
- 2.直接使用\*（char\*）进行赋值只能获取首元素。

·数码管可在一次循环中反复选择位选线和段选线实现多位显示补码数字的操作

·单片机中代码的实现应用循环操作（未规定单片机反复循环主函数）

## 存在问题

---

·本周对单片机的学习以及工作室的考核花费比较多的时间，线上上课注意力也不够集中，对自己的课程未能及时复习，周末复习花费了较多的时间。

## 下周规划

---

·学习数码管的动态显示以及矩阵键盘的使用