实验报告：lab03
金泽文 PB15111604

一、需求分析

输入： 两组(从小到大排好序的）链表 p（表头在 x4000）,m（表头在 x4001）

输出： 三组（从小到大排好序的）链表 p （表头在 x4000） ，m （表头在 x4001） ，pm （表头在 x4002）

要求： 不能使用额外内存

每个节点占用 2 个地址（一个保存下一个地址，一个保存 license）

只改变地址，不改变 license

每个链表以 x0000 结尾

二、算法设计

1.initial

read address

read license

2.loop

compare license: cmp pn, mn:

if same （z）:

add &pn to pm list:

set &pmn.addr = &pn

set &pmn = &pn

set &pn_1.addr = &pn.addr

set &mn_1.addr = &mn.addr

set &pmn.addr = x0000

&pn_1 = &pn

&pn = &pn.addr

if &pn == x0000 exit

pn = &pn.data

&mn_1 = &mn

&mn = &mn.addr

if &mn == x0000 exit

mn = &mn.data

else if less （n）

&pn_1 = &pn

&pn = &pn.addr

if &pn == x0000 exit

```
            pn = &pn.data
        else greater p
            &mn_1 = &mn
            &mn = &mn.addr
             if &mn == x0000 exit
            mn = &mn.data
```

寄存器分配

0 store current node address &pn
1 store current node address &mn
2 store last node address &pn_1
3 store last node address &mn_1
4 store current node license pn
5 store current node license mn
6 store current node address pmn
7 store temp values

具体算法请看代码旁的注释

三、实验总结：

本次实验吸取了上次的教训，在开始写代码之前认真、仔细、冷静的分析了需求，设计了算法并检查了算法。所用时间比上次短了很多，而且思路清晰了很多，体验痛快了很多。

但是中间还是遇到了一些 bug，比如在实现&pn_1 = &pn 的时候，一开始错用了 STR 指令。后来很快清除。

所以说：要想不花费大量的时间在初写代码时纠结，不花费更多时间调试，必须在开始敲代码之前冷静分析需求，设计算法。