

8.3

$60/(33*6*10^{-9})=303030303$ (个)

8.8

```
                ldi r0,address
                brn tail
                ld r1,limit
                add r1,r1,r0
                brp tail
                out
tail            halt
address        .fill x4000
limit          .fill #-127
```

8.12

假设 KBDSR 的内存地址是 xf400。

```
start          ldi r0,kbdsr
                brz start
                and r1,r1,#0
                sti r1,kbdsr
                brznp next_task
kbdsr          .fill xf400
```

8.15

a

把值 x4000 写入地址为 xfe00 的内存，显示器会持续输出字符 2。

b

在显示器上输出内存地址为 xfe02 的内容 2 次。

c

屏幕显示的情况与用户键入字符的时机有关。

如果是在指令 LD R0,B 之后、TRAP X21 之前输入的字符，那么首先跳转程序会输出两次键盘输入的值，然后回来 TRAP X21 又会输出一次该字符，总共输出三次；如果是其他情况输入的字符，那么只有跳转程序输出的两次该字符。

之后重新输出字符 2。

9.2

a

最多可以有 256 个 trap 服务程序。因为 trap 陷入矢量编号是 8-bit 的， $2^8=256$ 。

b

RET 指令可以将 R7 的内容装入 PC，而 R7 存储的是调用代码的地址，这样就可以从当前程序跳回调用程序，BR 无条件跳转无法恢复 PC 的值，一是因为 BR 跳转范围有限。二是 BR 无法根据 R7 跳转。

c

只有一次，就是把调用 TRAP 指令时的增量 PC 的值存进 R7。

9.4

a

1111 0000 0010 0001

b

x0430

c

x0437

d

HookemHorns

9.8

若 A 指向的值是质数，则 RESULT 为 1，否则为 0。

9.12

存在联系。程序执行后会对 DATA 数据进行升序排序。

9.16

.FILL x30000 数据过大，超过 LC-3 的 16bit 能表示的值，该错误是在汇编阶段被检测到的。

10.3

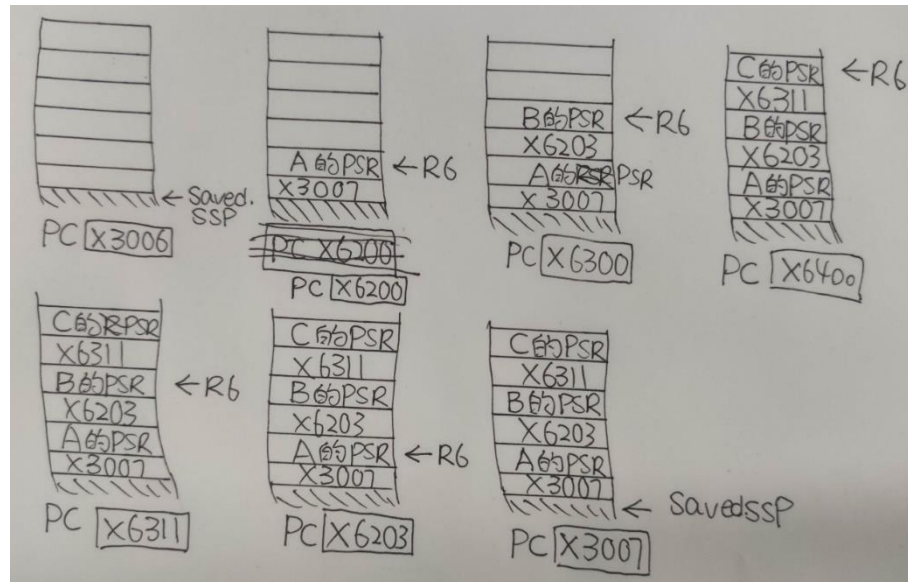
a : R1

b : R0

c : R3

d : R7

10.12



10.20

采用以时间换空间的方法，从高位到低位依次读取，每次换位都乘以 10 累加在一起。

```
.orig x3000
and r0,r0,#0
lea r6,ASCIIBUFF;
ld r5,char;
```

```
again    ld r4,count
multiply add r0,r0,r0;
          add r4,r4,#-1
          brp multiply
          ldr r2,r6,#0
          add r6,r6,#1
          add r2,r2,r5
          add r0,r0,r2
          add r1,r1,#-1
          brp again
          halt
```

```
ASCIIBUFF .blkw 10
char      .fill x-30
count     .fill #10
.end
```