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，28=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

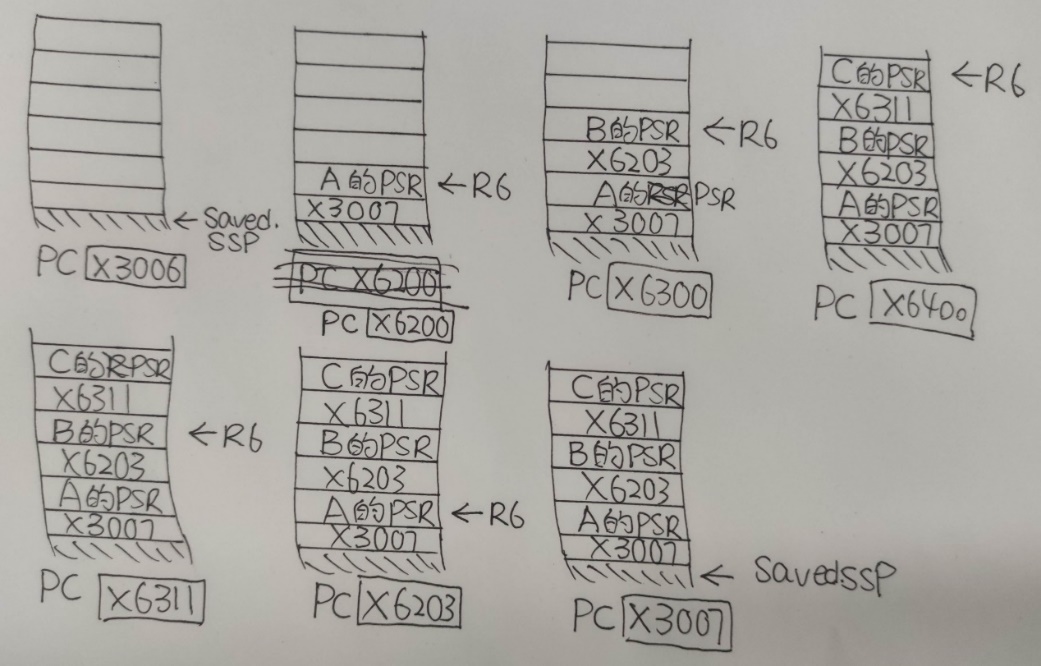
a : R1

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