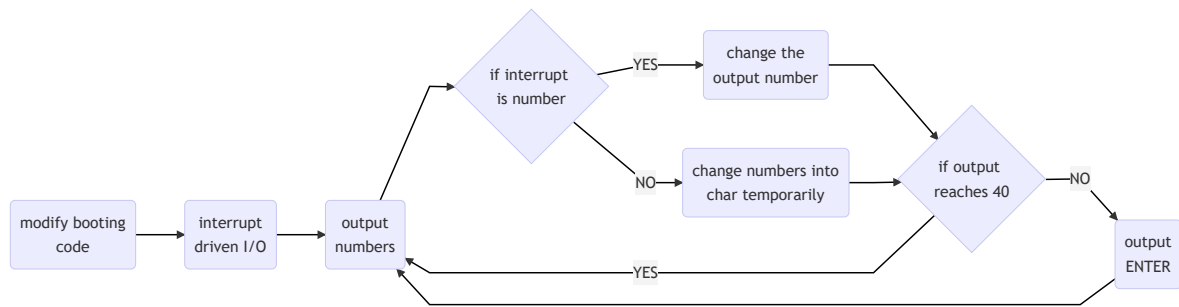


Lab3 Report:

Mermaid:



Code:

```
.ORIG x0200                ;system booting code
LD  R6,OS_SP
LD  R0,USER_PSR            ;push USER_PSR
ADD R6,R6,#-1
STR R0,R6,#0
LD  R0,USER_PC            ;push USER_PC
ADD R6,R6,#-1
STR R0,R6,#0
LD  R0,KBSR_IE            ;make KBSR[14] equal to 1
STI R0,KBSR
LD  R0,KBI_ADDR           ;intruption interrupt tabel vector
STI R0,KBI_INV
AND R0,R0,#0
RTI

OS_SP      .FILL  x3000
USER_PSR   .FILL  x8002
USER_PC    .FILL  x3000
KBSR       .FILL  xFE00
KBSR_IE    .FILL  x4000
KBDR       .FILL  xFE02
KBI_ADDR   .FILL  x0800
KBI_INV    .FILL  x0180
.END
```

```
.ORIG x0800                ;interrupt service routine
```

```

        ST    R0,SaveR0
        ST    R1,SaveR1

HIT     LDI    R0,KBSR_      ;check KBSR[15]
        BRzp  HIT
        LDI    R0,KBDR_

        LD     R1,ENTER      ;check whether R0 equals to x000A, if so , output
number -1
        ADD    R1,R1,R0      ;when r0 is 0 , then we needn't subtract 1
        BRnp  #6
        LD     R0,SaveR0
        ADD    R1,R0,#-16
        ADD    R1,R1,#-16
        ADD    R1,R1,#-16
        BRZ    #1
        ADD    R0,R0,#-1

DISP    LDI    R1,DSR_
        BRzp  DISP
        STI    R0,DDR_

        AND    R2,R2,#0      ;restart the output to make sure that there are 40
output
        ADD    R2,R2,#10

        ST     R0,SaveR0      ;output the ENTER
        LD     R0,StrEnter
        trap  x21
        LD     R0,SaveR0

        LD     R1,SaveR1
        RTI

SaveR0   .FILL  x0000
SaveR1   .FILL  x0000
KBSR_    .FILL  XFE00
KBDR_    .FILL  XFE02
DSR_     .FILL  XFE04
DDR_     .FILL  XFE06
ENTER    .FILL  XFFF6
StrEnter .FILL  x000A
        .END

```

```

        .ORIG x3000
        LD     R0,Ini_R0      ;Initial register
        AND    R1,R1,#0
        AND    R2,R2,#0

        AND    R3,R3,#0
        ADD    R3,R3,#2

        AND    R4,R4,#0

LOOP     JSR    JUDGE_NUMBER
        ADD    R3,R3,#0
        BRp    #1
        LD     R0,StoreR0_0    ;reload the number

```

```

        JSR JUDGE_NUMBER
        ADD R3,R3,#0
        BRnz #1
        ST R0,StoreR0_0    ;if interrupt is alphabet ,store the number
temporarily

        ST R0,StoreR0_1    ;output the ENTER
        LEA R0,Str_enter
        TRAP X22
        LD R0,StoreR0_1

        ADD R2,R2,#10      ;output 40 times

NUM40   JSR DELAY
        TRAP X21

        JSR DELAY
        TRAP X21

        JSR DELAY
        TRAP X21

        JSR DELAY
        TRAP X21

        ADD R2,R2,#-1
        BRZ LOOP
        BRnzp NUM40

DELAY   ST R1, DELAY_R1
        LD R1, DELAY_COUNT
DELAY_LOOP ADD R1, R1, #-1
        BRnp DELAY_LOOP
        LD R1, DELAY_R1
        RET

JUDGE_NUMBER AND R3,R3,#0    ;if R0 is number ,then R3 is 1,else is 0
        ST R0,StoreR0_3
        LD R4,ZERO
        ADD R4,R4,R0
        BRn #4
        LD R4,NINE
        ADD R4,R4,R0
        BRp #1
        ADD R3,R3,#1
        LD R0,StoreR0_3
        RET

DELAY_COUNT .FILL #256
DELAY_R1 .BLKW #1
StoreR0_0 .BLKW #1
StoreR0_1 .BLKW #1
StoreR0_3 .BLKW #1
Ini_R0 .FILL x0037
ZERO .FILL xFFD0
NINE .FILL xFFC7

```

```
Str_enter .STRINGZ "\\n"
```

```
.END
```

Algorithm:

1. in booting code, make KBSR_EI equal to 1. and make address x0180 equal to x0800. then push USER_PSR ,then push USER_PC
2. in interrupt section , we judge whether interrupt is ENTER ,if so ,we make number subtract 1. and if number is negative , we set R0 equal to 0. and at last ,we output an ENTER and reset the counter
3. in user mode, we use loop to output R0. and we should judge whether R0 is char or number, if r0 is char ,we should replace it with number in next loop and if r0 is number ,we could do nothing.

Check:

- TA: How do you write you booting code?
I: Make KBSR[14] equals to 1 and make contents in x0180 equal to x0800. Then push PSR and PC.