.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 ;intruction interrupt tabel vector

STI R0,KBI\_INV

AND R0,R0,#0

RTI

AND R1,R1,#0

AND R2,R2,#0

AND R3,R3,#0

ADD R3,R3,#0

AND R4,R4,#0

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 KSBR[15]

BRzp HIT

LDI R0,KBDR\_

CHECK 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

ADDF

ADD R0,R0,#-1

ADD R0,R0,#1

AND R1,R1,#0

AND R2,R2,#0

AND R3,R3,#0

ADD R3,R3,#0

AND R4,R4,#0

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 ;User Program

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\_1 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

LOOP\_2 ST R0,StoreR0\_1 ;output the ENTER

LEA R0,Str\_enter

TRAP X22

LD R0,StoreR0\_1

ADD R2,R2,#10 ;output 40 times

LOOP\_3 JSR DELAY

TRAP X21

JSR DELAY

TRAP X21

JSR DELAY

TRAP X21

JSR DELAY

TRAP X21

ADD R2,R2,#-1

BRz LOOP\_1

BRnzp LOOP\_3

;delay function

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 #2000

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