```
val equ 99h
scl equ P2.0
sda equ P2.1
org 0000h


main:     ;subrutina main
          acall lcd_on ;apelam subrutina care porneste ecranul
          loop:
          lcall initializarei2c
          lcall starti2c
          mov a, #80h
          lcall writei2c
          jc loop
          mov a, #0F5h
          lcall writei2c
          jc loop
          notdone:
          lcall rstarti2c
          mov a, #81h
          lcall writei2c
          jc notdone
          lcall acki2c
          lcall readi2c
          lcall acki2c
          lcall readi2c
          lcall acki2c
          lcall readi2c
          lcall nacki2c
          lcall stopi2c

          acall but1  ;apelam subrutina care verifica daca este apasat butonul 1
          acall but2  ;apelam subrutina care verifica daca este apasat butonul 1
          acall but3  ;apelam subrutina care verifica daca este apasat butonul 1
          ljmp loop
ret

lcd_on:        ;subrutina care porneste ecranul LCD- ului
          mov A, #38h      ;mutam in A valoarea 38h, comanda care ii spune ecranului ca folosim 2 linii
          acall command  ;apelam subrutina command, subrutina care ii transmite ecranului o comanda
          mov A, #0Eh                ;mutam in A valoarea 0Eh, aceasta fiind comanda pentru cursor on
          acall command
```

```
        mov A, #80h             ;mutam in A valoarea 80h,comanda pentru pozitionarea cursorului in
stanga sus a ecranului(lini 1, coloana 1)
        acall command
ret


command:                ;subrutina care transmite ecranului comanda
        mov P1, A       ;mutam in portul de date valoarea din A
        clr P0.4 ;setam portul P0.4(RS) pe 0, indicand faptul ca ii dam o comanda
        setb P0.3       ;setam portul P0.3(E) pe 1, adica permitem scrierea/citirea de date
        acall delay     ;apelam subrutina delay, subrutina care realizeaza o intarziere de 1ms
        clr P0.3 ;trecem enable pe 0, adica oprim scrierea/citirea de date
ret

data1:                  ;subrutina care transmite ecranului date
        mov P1, A       ;mutam in portul de date valoarea din A
        setb P0.4       ;setam portul P0.4(RS) pe 1, indicand faptul ca ii transmitem date
        setb P0.3       ;setam portul P0.3(E) pe 1, adica permitem scrierea/citirea de date
        acall delay     ;apelam subrutina delay, subrutina care realizeaza o intarziere de 1ms
        clr P0.3 ;trecem enable pe 0, adica oprim scrierea/citirea de date
ret

but1:
        jb P0.0, exit1  ;verificam daca butonul este apasat
        mov A, #01h             ;mutam in A valoarea 01h, comanda de clear la ecran
        acall command
        mov A, #80h             ;mutam in A valoarea 80h,comanda pentru pozitionarea cursorului in
stanga sus a ecranului(lini 1, coloana 1)
        acall command
        mov A, #'T'             ;Scriem litera cu litera cuvantul "Temperatura"
        acall data1
        mov A, #'e'
        acall data1
        mov A, #'m'
        acall data1
        mov A, #'p'
        acall data1
        mov A, #'e'
        acall data1
        mov A, #'r'
        acall data1
        mov A, #'a'
```

```
	acall data1
	mov A, #'t'
	acall data1
	mov A, #'u'
	acall data1
	mov A, #'r'
	acall data1
	mov A, #'a'
	acall data1
	lcall temperatura
	exit1:
ret

but2:
	jb P0.1, exit2	;verificam daca butonul este apasat
	mov A, #01h		;mutam in A valoarea 01h, comanda de clear la ecran
	acall command
	mov A, #80h		;mutam in A valoarea 80h,comanda pentru pozitionarea cursorului in
stanga sus a ecranului(lini 1, coloana 1)
	acall command
	mov A, #'U'		;Scriem litera cu litera cuvantul "Prev"
	acall data1
	mov A, #'m'
	acall data1
	mov A, #'i'
	acall data1
	mov A, #'d'
	acall data1
	mov A, #'i'
	acall data1
	mov A, #'t'
	acall data1
	mov A, #'a'
	acall data1
	mov A, #'t'
	acall data1
	mov A, #'e'
	acall data1
	mov A, #'a'
	acall data1
	exit2:
ret
```

```
but3:
	jb P0.2, exit3	;verificam daca butonul este apasat
	mov A, #01h		;mutam in A valoarea 01h, comanda de clear la ecran
	acall command
	mov A, #80h		;mutam in A valoarea 80h,comanda pentru pozitionarea cursorului in
stanga sus a ecranului(lini 1, coloana 1)
	acall command
	mov A, #'S'		;Scriem litera cu litera cuvantul "Setup"
	acall data1
	mov A, #'e'
	acall data1
	mov A, #'l'
	acall data1
	mov A, #'e'
	acall data1
	mov A, #'c'
	acall data1
	mov A, #'t'
	acall data1
	exit3:
ret

delay:				;subrutina pentru o intarziere de 1ms
mov r7, #val
timer:
	nop
	nop
	nop
	nop
	djnz r7, timer
	nop
ret

initializarei2c:
	setb sda
	setb scl
ret

starti2c:
	setb scl
	setb sda
```

```
        clr sda
ret

stopi2c:
        clr scl
        clr sda
        setb scl
        setb sda
ret

writei2c:
        mov R1, #8
datasend:
        clr scl
        rlc A
        mov sda, c
        setb scl
        djnz r1, datasend
        clr scl
        setb sda
        setb scl
        mov c, sda
ret

acki2c:
        clr sda
        setb scl
        clr scl
        setb sda
ret

nacki2c:
        setb sda
        setb scl
        clr scl
        setb scl
ret

readi2c:
        mov r1, #8
dataread:
        clr scl
```

```
          setb scl
          mov c, sda
          rlc A
          djnz r1, dataread
          clr scl
          setb sda
ret

rstarti2c:
          clr scl
          setb sda
          setb scl
          clr sda
ret



temperatura:
mov A, #8Ch
lcall command
lcall delay
mov A, P3
cjne A, #0A6h, t16
mov A, #'1'
acall data1
mov A, #'5'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4

t16:
mov A, P3
cjne A, #66h, t17
mov A, #'1'
acall data1
mov A, #'6'
acall data1
mov A, #'C'
acall data1
acall delay
```

```
jmp exit4


t17:
mov A, P3
cjne A, #16h, t18
mov A, #'1'
acall data1
mov A, #'7'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4


t18:
mov A, P3
cjne A, #0D6h, t19
mov A, #'1'
acall data1
mov A, #'8'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4


t19:
mov A, P3
cjne A, #76h, t20
mov A, #'1'
acall data1
mov A, #'9'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4

t20:
```

```
mov A, P3
cjne A, #8Eh, t21
mov A, #'2'
acall data1
mov A, #'0'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4


t21:
mov A, P3
cjne A, #2Eh, t22
mov A, #'2'
acall data1
mov A, #'1'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4

t22:
mov A, P3
cjne A, #0EEh, t23
mov A, #'2'
acall data1
mov A, #'2'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4


t23:
mov A, P3
cjne A, #5Eh, t24
mov A, #'2'
acall data1
```

```
mov A, #'3'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4


t24:
mov A, P3
cjne A, #0BEh, t25
mov A, #'2'
acall data1
mov A, #'4'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4

t25:
mov A, P3
cjne A, #1h, t26
mov A, #'2'
acall data1
mov A, #'5'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4


t26:
mov A, P3
cjne A, #0C1h, t27
mov A, #'2'
acall data1
mov A, #'6'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4


t27:
mov A, P3
cjne A, #61h, t28
mov A, #'2'
acall data1
mov A, #'7'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4


t28:
mov A, P3
cjne A, #91h, t29
mov A, #'2'
acall data1
mov A, #'8'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4

t29:
mov A, P3
cjne A, #31h, t30
mov A, #'2'
acall data1
mov A, #'9'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4
```

```
t30:
mov A, P3
cjne A, #0F1h, exit4
mov A, #'3'
acall data1
mov A, #'0'
acall data1
mov A, #'C'
acall data1
acall delay
jmp exit4

exit4:
ret

end
```