



Re: 8031 question

ORG 32 ; INT RAM 32+96 BY  
FLAG: DS 1  
HANDS BIT FLAG.0 ; 20 ms HANDSHAKING  
PISMO BIT FLAG.1 ; =0: LATIN =1: CYRILIC  
OVERWR BIT FLAG.2 ; =0: PUSHWRITE =1: OVERWRITE  
PISMO2 BIT FLAG.3 ; SAME AS LETTER, FOR DEMO TXT AND SERIAL  
NEWSEC BIT FLAG.4 ; FULL SEC HANDSHAKING  
DCL BIT FLAG.5 ; DATE CYRILIC (1) – LATIN (0)  
DCH BIT FLAG.6 ;  
INHIBIT BIT FLAG.7 ; PREVENT WRITE FROM RTC 58321 IN (RTC)

KEYFLAG: DS 1  
SHIFLAG BIT KEYFLAG.0 ; SHIFT PRESS  
CAPSFLAG BIT KEYFLAG.1 ; CAPS PRESS  
CTRLFLAG BIT KEYFLAG.2 ; CTRL PRESS  
ALTFLAG BIT KEYFLAG.3 ; ALT PRESS  
NUMFLAG BIT KEYFLAG.4 ; NUM LOCK PRESS  
SCRFLAG BIT KEYFLAG.5 ; SCROLL LOCK PRESS

KEYON: DS 1  
CAPSON BIT KEYON.0 ; CAPS LOCK ACTIVE (LED ON)  
NUMON BIT KEYON.1 ; NUM LOCK ACTIVE (LED ON)  
SCRON BIT KEYON.2 ; SCROLL LOCK ACTIVE (LED ON)  
INSON BIT KEYON.3 ; INSERT ACTIVE (OVERWRITE MODE)

FREERUN: DS 1 ; FREERUNING COUNTER FOR BLINK

; FOLLOW REGISTERS NON BIT-ADRESIBILE

GLBRO: DS 1 ; MAIL COUNTER FOR LINES 0–6  
JUST: DS 2 ; JUST PRESS KEY, 2–WIDE FIFO, MASTER +0  
EPTR: DS 2 ; MEM PTR LEFT CHAR ON SCREEN IN EDIT MODE  
RPTR: DS 2 ; MEM PTR IN RUN MODE  
CURSOR: DS 2 ; MEM PTR CURSOR ON SCREEN  
KEYCOUNT: DS 1 ; COUNTDOWN FOR CTRL,ALT,SHIFT OFF  
REFRENO: DS 2 ; ADRESS REFREN (O)  
REFRENX: DS 2 ; ADRESS REFREN <X>  
RETADRO: DS 2 ; ADRESS FOR RETURN (O)  
RETADRX: DS 2 ; ADRESS FOR RETURN (X)

RTCCOU: DS 1 ; RTC REGISTER COUNTER, 0–12  
RTC: DS 13 ; +0: 1 S +4: 1 H +7: 1 Day +11: 1 Y  
; +1:10 S +5:10 H +8:10 Day +12:10 Y  
; +2: 1 Min +6:DayWeek +9: 1 Mon  
; +3:10 Min +10:10 Mon

TEMPC: DS 1 ; TEMPERATURE IN CELSIUSE DEGREE (BIN)  
TEMPTRIK: DS 1 ; TEMPORARY FOR TRICKS D – D2 – V – C  
TEMPDIV: DS 1 ; MEASURE FOR PERIOD TEMPERATURE

STACK EQU \$

```
; macro def
%*DEFINE(MOVW(MEMLOC,CONST))
(MOV %MEMLOC,#LOW %CONST
MOV %MEMLOC+1,#HIGH %CONST
)
```

```
; ; start
CSEG
ORG 0
LJMP INICI
; interrupts
ORG 3
LJMP CONT0
; int 350 Hz
```

```
ORG 1BH ; TIMER 1 INTERRUPT
PUSH ACC
PUSH PSW
MOV A,#-40
ADD A,TL1
MOV TL1,A
MOV TH1,#-12 ; 11*256+41=2857 2857*350*12=12MHz
PUSH DPL
PUSH DPH
```

```
ORL PSW,#18H ; SEL RB3 (USES R0 BANK 3)
```

```
MOV A,GLBRO ; ----- TEMPERATURE
CJNE A,#1,STILLNT
MOV R0,#TEMPDIV ; TEMPERATURU CITA SAMO U KRUGU 1
INC @R0
MOV A,@R0
CJNE A,#80,STILLNT ; OD #NN LINEARNO ZAVISI TEMP GAIN
CLR TR0
MOV A,TH0
MOV @R0,A ; PRIVREMENO KORISTI TEMPDIV LOKACIJU ZA XCHD
MOV A,TL0
XCHD A,@R0
MOV @R0,#0
SWAP A ; A = (TH0,TL0)/16
MOV TEMPC,A
MOV TL0,#0
MOV TH0,#0
SETB TR0
STILLNT: ; ----- KRAJ TEMPERATURE
```

```
MOV A,GLBRO ; INC GLBRO, RANGE 1..6
INC A
ANL A,#7
CJNE A,#7,NSKIP7
```

Re: 8031 question

```
CLR A
NSKIP7: MOV GLBRO,A
```

```
SWAP A
RL A ; NEW GLBRO * 32
MOV DPL,A
MOV DPH,#HIGH BUFFER2
```

```
GOSHIF: NOP ; 1 T
NOP ; 1 T
MOVX A,@DPTR ; 2 T
MOV SBUF,A ; 1 T SHIFT OUT
INC DPTR ; 2 T
MOV A,DPL ; 1 T
ANL A,#1FH ; 1 T
CJNE A,#WIDTH,GOSHIF ; 2 T PETLJA MAX 280 T (ZA WIDTH=28)
```

```
; ----- RTC START -----
SETB RXD
JB INHIBIT,EQSEC
INC RTCCOU
MOV A,RTCCOU
CJNE A,#13,NNSEC
CLR NEWSEC ; CIKLUS 13: UPRAVO ZAVRSEN UPIS
JMP NSEC
NNSEC: JC NOVFRTC ; RTCCOU = 0...12: CITANJE U TOKU
; RTCCOU > 12: BAJATA SEKUNDA
NSEC: MOV A,#0C0H ; ADRESA REGISTRA ZA SEKUNDE RTC
SETB P3.5 ; DISABLE DECODER
MOV P1,A
SETB P1.5 ; AW PULSE
CLR P1.5 ; AW LO
MOV P1,#0CFH
SETB P1.4 ; RD PULSE
DW 0
MOV A,P1
CLR P1.4 ; RD LO
ANL A,#0FH
CJNE A,RTC,NOTEQSEC
MOV RTCCOU,#0FEH ; OSTACE ZALEPLJEN ZA FE DOK NE DODJE NOVA SEC
JMP EQSEC ; NIJE BILO PROMENE U SEKUNDAMA
NOTEQSEC: MOV RTCCOU,#0FFH ; NOVA SEKUNDA: INICIRA CITANJE VREMENA
JMP EQSEC

NOVFRTC: ADD A,#RTC
MOV R0,A
MOV A,RTCCOU
ORL A,#0C0H
SETB P3.5 ; DISABLE DECODER
MOV P1,A
SETB P1.5 ; AW PULSE
```

Re: 8031 question

Re: 8031 question

```
CLR P1.5 ; AW LO
MOV P1,#0CFH
SETB P1.4 ; RD PULSE
DW 0
MOV A,P1
CLR P1.4 ; RD LO
ANL A,#0FH
MOV @R0,A ; WRITE RTC DATA IN INTERNAL RAM
MOV A,RTC+5
ANL A,#3
MOV RTC+5,A
EQSEC:
; ----- RTC END -----
MOV C,RXD ; TEST SERIAL INPUT
JC NEMADATA
JMP IMADATA
```

```
NEMADATA: MOV A,GLBRO
ORL A,#0C8H
CLR P3.3
SETB P3.3 ; STROBE PULSE
MOV P1,A ; SET DECODER STATE
CLR P3.5 ; ENABLE DECODER
```

```
MOV A,GLBRO
CJNE A,#6,NO20MS
; CEO SCROLL: PREMESTITI BUFFER U BUFFER2
```

```
ANL P2,#7 ;
INC P2 ; PUSH L A T C H P2
MOV A,#0FFH ;
READP2: INC A ;
DJNZ P2,READP2 ;
PUSH ACC ;
MOV DPTR,#BUFFER+219
MOV R0,#219
MOV P2,#HIGH BUFFER2
LDIR1: MOVX A,@DPTR
MOVX @R0,A
DEC DPL
DJNZ R0,LDIR1 ; PETLJA 1533 T (1.533 ms)
MOVX A,@DPTR
MOVX @R0,A
POP P2 ; /
```

```
CLR HANDS ; HANDSHAKING FLAG
```

```
MOV A,KEYCOUNT
JZ NOSHON
DEC KEYCOUNT
CJNE A,#1,NOSHON
```

Re: 8031 question

Re: 8031 question

```
MOV KEYFLAG,#0 ; GASENJE SHIFT,CTRL,ALT AKO OTKAZE OTP  
NOSHON:  
INC FREERUN  
MOV A,FREERUN  
ANL A,#3  
JNZ NO20MS  
INC FREERUN ; MALO UBRZANJE KURSORA
```

```
NO20MS:  
POP DPH  
POP DPL  
POP PSW  
POP ACC  
RETI  
; ; RS232 input  
IMADATA: CLR EA  
MOV R6,#250  
WAITB0: MOV R5,#30 ; \  
DJNZ R5,$  
DJNZ R6,STILLPAT  
JMP NO20MSE ; PATIENCE LOST AFTER 18.5 mS  
STILLPAT: MOV R7,#200  
INNER1: JNB RXD,AAA ; \  
JNB RXD,AAA  
JB RXD,WAITB0 ; /  
AAA: DJNZ R7,INNER1 ; // 1.6 mS TEST (3.84 BITA)
```

```
MOV R6,#250  
INNER2: JNB RXD,BBB  
JNB RXD,BBB  
JB RXD,FOUSPACE ; \  
BBB: DW 0,0,0  
DJNZ R6,INNER2 ; /  
NO20MSE: MOV TH1,#-5  
MOV GLBRO,#7  
SETB EA  
JMP NO20MS ; PATIENCE LOST AFTER 2.5 mS
```

```
FOUSPACE:  
MOV TMOD,#25H ; TIMER 1 BAUD RATE DIVIDER  
MOV TH1,#-13 ; BAUD RATE = (1000000)/(13*32)=2403.8  
MOV TCON,#41H ; START TIMER 1, STOP TEMPERATURE COUNTER  
MOV SCON,#70H ; ENABLE SERIAL RECEPTION
```

```
LEADLOOP: CALL RXUPPER  
CJNE A,#'D',LEADLOOP  
CALL RXUPPER  
CJNE A,#'I',LEADLOOP  
CALL RXUPPER  
CJNE A,#'S',LEADLOOP  
CALL RXUPPER
```

```
CJNE A,#'P',LEADLOOP
CALL RXUPPER
CJNE A,#'0',NOTALL
JMP YESALL
NOTALL: CJNE A,#'1',LEADLOOP ; BROJ DISPLEJA
YESALL: ; SYNC FOUND
CLR PISMO2
CALL RXBYTE
CJNE A,#'-',NOCLRM
CALL CLRMEM

SERLOOP: CALL RXBYTE
NOCLRM: CJNE A,#128,$+3
JNC SERLOOP ; IGNORE =>127
CJNE A,#32,$+3
JC SERLOOP ; IGNORE < 32
CJNE A,#',',SNOMETA
; KOMANDNI ZNAK: "/"
CALL RXBYTE
CJNE A,#',',SNOKROZ ; "/" SE PREDSTAVLJA KAO "/"
JMP SNOMETA
SNOKROZ: CJNE A,#'[',SNOCIR ; [ = CIRILICA
SETB PISMO2
JMP SERLOOP
SNOCIR: CJNE A,#']',SNOLAT ; ] = LATINICA
CLR PISMO2
JMP SERLOOP
SNOLAT: CJNE A,#60H,$+3
JC SVELSL
ADD A,#-20H ; LOWERCASE ADJUST UPPERCASE
SVELSL: ADD A,#0A0H ; TRICK CODE 0C0H-0FFH

SNOMETA: CJNE A,#41H,$+3
JC SNOLAT1
CJNE A,#80H,$+3
JNC SNOLAT1
JNB PISMO2,SNOLAT1
ADD A,#40H ; KONVERTUJE LATINICU U CIRILICU

SNOLAT1: PUSH ACC
CALL MOVDPET
POP ACC
MOVX @DPTR,A
INC DPTR
CALL MOVETDP
JMP SERLOOP

RXUPPER: CALL RXBYTE
CJNE A,#60H,$+3
JC UPC
```

ADD A,#-20H  
UPC: RET

RXBYTE: MOV R7,#150  
OUTSER: MOV R6,#250  
INNSER: JB RI,RECEIVED  
MOV R5,#30  
DJNZ R5,\$  
DJNZ R6,INNSER  
DJNZ R7,OUTSER  
; PATIENCE LOST AFTER 1 sec  
MOV SP,#STACK  
MOV SCON,#0  
MOV PSW,#0  
MOV DPTR,#0  
PUSH DPL  
PUSH DPH  
MOV TH1,#-5  
RETI ; UNLEGAL RESTART 0

RECEIVED: CLR RI  
MOV A,SBUF  
RET  
; ; dalje INT0  
CONT0: PUSH ACC  
PUSH PSW  
MOV KEYCOUNT,#250  
CLR TR1 ; STOP TIMING COUNT  
MOV C,P3.5  
MOV F0,C  
SETB P3.5 ; DISABLE ANODE DECODER  
ORL PSW,#18H ; SEL RB3 (USES R6,R7 BANK 3)  
MOV R7,#8 ; BITS TO SHIFT IN

DATA8: MOV R6,#80 ; PATIENCE TIMER  
LOCLK: JB P3.2,NOLO  
NOP  
DJNZ R6,LOCLK ; STILL LO  
JMP TIMEOUT ; PATIENCE LOST

NOLO: MOV R6,#80 ; PATIENCE TIMER  
HICLK: JNB P3.2,HILO  
NOP  
DJNZ R6,HICLK ; STILL HI  
JMP TIMEOUT ; PATIENCE LOST  
HILO: ; OPADAJUCA IVICA  
MOV C,P1.7  
RRC A  
DJNZ R7,DATA8

MOV R6,A

```
MOV A,JUST
JZ INJUST
MOV A,R6
MOV JUST+1,A ; HI FIFO
JMP TIMEOUT
INJUST: MOV A,R6
MOV JUST,A ; LO FIFO
```

```
TIMEOUT: SETB TR1 ; RESTART TIMER
MOV C,F0
MOV P3.5,C ; REENABLE ANODE DECODER
POP PSW
POP ACC
RETI
; ; inici
```

```
INICI:
MOV R0,#2
MOV R1,#126
GOCLRAM: MOV @R0,#0 ; BRISANJE RAM-A
INC R0
DJNZ R1,GOCLRAM
```

```
MOV SP,#STACK
%MOVW(EPTR,TEXT)
%MOVW(RPTR,TEXT)
%MOVW(CURSOR,TEXT)
```

```
CALL CLRBUF
CALL CLRBUF2
MOV TMOD,#15H
MOV TCON,#51H
MOV SCON,#0
MOV IP,#1 ; INT0 (KYBD CLK) HIGHEST PRIORITY
MOV IE,#89H ; ENABLE INTERRUPTS
```

```
; ; farm
RUN: %MOVW(RPTR,TEXT)
%MOVW(REFRENO,0)
%MOVW(REFRENX,0)
RUN2: %MOVW(RETADRO,0)
%MOVW(RETADRX,0)
CALL CLRBUF
```

```
CALL MEM
MOV A,DPL
ORL A,DPH
JNZ FARM
%MOVW(CURSOR,TEXT)
%MOVW(EPTR,TEXT)
JMP EDIT ; GO EDIT IF NO TEXT IN MEM
```

```
FARM: MOV SP,#STACK
CLR INHIBIT
MOV KEYFLAG,#0
MOV A,TEMPTRIK
JZ NOTEMPN
MOV TEMPTRIK,#0 ; DOVRSAVA D – D2 – C – V AKO JE POCETO TRIKOM
CJNE A,#'V'+0A0H,NOTVRE
CALL VRESUB
JMP NOTEMPN
NOTVRE: CALL TEMPSUB
```

```
NOTEMPN: CLR A
XCH A,JUST
JNZ PRESS
XCH A,JUST+1
JZ NOPRES
PRESS: CJNE A,#1,NOPRES ; 7EH = ESC, BIT 7 SET = PRIT, CPL!
JMP EDIT
```

```
NOPRES: MOV DPL,RPTR
MOV DPH,RPTR+1
MOVX A,@DPTR
PUSH ACC
CALL INCWRAP
MOV RPTR,DPL
MOV RPTR+1,DPH
POP ACC
CJNE A,#0C0H,$+3
JC REGULAR
; OVDE JE TRIK: >= 0C0H
ANL A,#3FH
MOV DPTR,#FARM ; RET ADDRESS ...
PUSH DPL
PUSH DPH ; ... ON STACK
MOV DPTR,#TRICKTAB
JMP EXECA ; IZVRSENJE TRIKA
```

```
REGULAR: CALL SYNC
CALL WRDZEP
CALL LEFT8
JMP FARM
; ; trikovi
```

TRICKTAB EQU \$ ; TABLICA TRIKOVA (KAD JE CHAR >= 0C0H)

```
DW FARM, FARM, FARM, FARM, FARM, FARM, FARM, FARM ; blk – '
DW OTZAG,ZATZA,FARM, FARM, FARM, FARM, FARM, FARM ; ( – /
DW TRIK0,TRIK1,TRIK2,TRIK3,TRIK4,TRIK5,TRIK6,TRIK7 ; 0 – 7
DW TRIK8,TRIK9,FARM, FARM, MANJE,FARM, VECE, FARM ; 8 – ?
DW FARM, FARM, NUMER,TEMP, DOLE, FARM, FARM, GORE ; @ – G
```

Re: 8031 question

DW DATH, FARM, FARM, DATUM,LEVO, FARM, FARM, CTRL0 ; H – O  
DW PAUZA,FARM, FARM, DATS, BLINK,FARM, VREME,FARM ; P – W  
DW CTRLX,FARM, FARM, FARM, FARM, FARM, FARM, FARM ; X – DJ

```
CTRL0: ; ---  
CALL IFBRK  
MOV A,REFRENO  
ORL A,REFRENO+1  
JZ IGNORE1 ; NEMA REFRENA  
MOV A,RETADRO  
ORL A,RETADRO+1  
JNZ IGNORE1 ; NIJE SE VRATIO SA PRETHODNOG  
MOV RETADRO,RPTR  
MOV RETADRO+1,RPTR+1  
MOV RPTR,REFRENO  
MOV RPTR+1,REFRENO+1  
IGNORE1: RET
```

```
OTZAG: ; ---  
MOV REFRENO,RPTR  
MOV REFRENO+1,RPTR+1  
RET  
ZATZA: ; ---  
MOV A,REFRENO  
ORL A,REFRENO+1  
JZ IGNORE2 ; NEMA REFRENA  
MOV A,RETADRO  
ORL A,RETADRO+1  
JZ IGNORE2 ; NEMA GDE DA SE VRATI  
MOV RPTR,RETADRO  
MOV RPTR+1,RETADRO+1  
MOV RETADRO,#0  
MOV RETADRO+1,#0  
IGNORE2: RET
```

```
CTRLX: ; ---  
CALL IFBRK  
MOV A,REFRENX  
ORL A,REFRENX+1  
JZ IGNORE3 ; NEMA REFRENA  
MOV A,RETADRX  
ORL A,RETADRX+1  
JNZ IGNORE3 ; NIJE SE VRATIO SA PRETHODNOG  
MOV RETADRX,RPTR  
MOV RETADRX+1,RPTR+1  
MOV RPTR,REFRENX  
MOV RPTR+1,REFRENX+1  
IGNORE3: RET
```

MANJE: ; ---

```
MOV REFRENX,RPTR
MOV REFRENX+1,RPTR+1
```

```
MOV DPL,RPTR
MOV DPH,RPTR+1
SRCHVE: CALL IFBRK
MOVX A,@DPTR
CJNE A,#>' +0A0H,NFOUVE
CALL INCWRAP ; NASAO ZNAK ">"
JMP MOVRPDP
```

```
NFOUVE: CALL CPDPET
JNC NEMAVE ; NIJE NASAO ">" DO KRAJA
CALL INCWRAP
JMP SRCHVE
```

```
NEMAVE: MOV DPTR,#TEXT
MOVRPDP: MOV RPTR,DPL
MOV RPTR+1,DPH
RET
```

```
VECE: ; ----
MOV A,REFRENX
ORL A,REFRENX+1
JZ IGNORE4 ; NEMA REFRENA
MOV A,RETADRX
ORL A,RETADRX+1
JZ IGNORE4 ; NEMA GDE DA SE VRATI
MOV RPTR,RETADRX
MOV RPTR+1,RETADRX+1
MOV RETADRX,#0
MOV RETADRX+1,#0
IGNORE4: RET
```

```
TEMP: ; ---- TEMPERATURA ----
CALL CLRBUF
MOV A,#5 ; PRVO PAUZA 500 mS...
CALL MS100
CALL TEMPAUX
SKRTEMP: CALL ODJED
TEMPSUB: CALL IFNUM2
CALL MS100
MOV DPL,RPTR ; READ MEM POINTER
MOV DPH,RPTR+1
MOVX A,@DPTR
CJNE A,#0C0H,$+3
JNC NCLR ; NIJE CLEAR 0.5 S AKO SLEDI NOVI TRIK
CALL CLRBUF
MOV A,#5 ; ...PA NA KRAJU JOS JEDNA PAUZA 500 mS
JMP MS100
```

NCLR: RET

```
DATUM: ; ---- DATUM LATINICOM ----  
CALL CLRBUF  
MOV A,#5 ; PRVO PAUZA 500 mS...  
CALL MS100  
CALL DATUMAUX  
JMP SKRTEMP
```

```
DATS: ; ---- DATUM CIRILICOM ----  
CALL CLRBUF  
MOV A,#5 ; PRVO PAUZA 500 mS...  
CALL MS100  
CALL DAT2AUX  
JMP SKRTEMP
```

```
DATH: ; ---- DATUM HRVATSKI ----  
CALL CLRBUF  
MOV A,#5 ; PRVO PAUZA 500 mS...  
CALL MS100  
CALL DATHAUX  
JMP SKRTEMP
```

```
NUMER: ; ---- DATUM SA MESECIMA NUMERICKI ----  
CALL CLRBUF  
MOV A,#5 ; PRVO PAUZA 500 mS...  
CALL MS100  
CALL DATNAUX  
JMP SKRTEMP
```

```
VREME: ; ---- HH:MM:SS ----  
CALL CLRBUF  
MOV A,#3 ; PRVO PAUZA 300-800 mS...  
CALL MS100  
VRESUB:  
CALL IFNUM2 ; VRATICE SE SA SEKUNDAMA U R4  
GOVRE:  
CALL WAITSEC  
CALL VREMEAUX  
CALL ODJED ; VREME HH:MM:SS  
GVR2: MOV A,#5  
CALL MS100  
CALL VREMEBL  
CALL ODJED ; VREME HH MM SS  
DJNZ R4,GOVRE  
CALL WAITSEC
```

```
MOV DPL,RPTR ; READ MEM POINTER  
MOV DPH,RPTR+1  
MOVX A,@DPTR  
CJNE A,#0C0H,$+3
```

Re: 8031 question

```
JNC NCLR V ; NIJE CLEAR 0.5 S AKO SLEDI NOVI TRIK
CALL CLRBUF
MOV A,#5 ; ...PA NA KRAJU JOS JEDNA PAUZA 500 mS
JMP MS100
NCLR V: CALL VREMEAUX
```

```
JMP ODJED
```

```
DOLE: ; --- POMERANJE NADOLE ---
CALL AUX
MOV R6,#4
DNB: CALL IFBRK
MOV DPTR,#BUFFER+32 ; 4x SAMO BAFER POMERA DOLE
MOV P2,#HIGH BUFFER
MOV R0,#LOW BUFFER
CALL SHIFTDN
CALL CLR6 ; I BRISE GORNJI RED
DJNZ R6,DNB
```

```
MOV R6,#7
DNAB: CALL IFBRK
MOV DPTR,#BUFFER+32 ; 7x POMERA BAFER DOLE...
MOV P2,#HIGH BUFFER
MOV R0,#LOW BUFFER
CALL SHIFTDN
```

```
MOV DPTR,#AUXBUF ; ...PREMESTA DONJI RED IZ AUXBUF-a...
MOV P2,#HIGH BUFFER
MOV R0,#LOW BUFFER+32*6
MOV R4,#32
CALL GD2
```

```
MOV DPTR,#AUXBUF+32 ; ...POMERA AUXBUF DOLE
MOV P2,#HIGH AUXBUF
MOV R0,#LOW AUXBUF
CALL SHIFTDN2
DJNZ R6,DNAB
RET
```

```
SHIFTDN: CALL SYNC4
SHIFTDN2: MOV R4,#6*32
GD2: MOVX A,@DPTR
INC DPL
MOVX @R0,A
INC R0
DJNZ R4,GD2
RET
```

```
CLR7: MOV DPTR,#BUFFER+7*32
JMP CLRX
CLR6: MOV DPTR,#BUFFER+6*32
```

Re: 8031 question

Re: 8031 question

```
JMP CLRX
CLR0: MOV DPTR,#BUFFER
CLRX: MOV R5,#32
CLR A
GOCLR06: MOVX @DPTR,A
INC DPL
DJNZ R5,GOCLR06
RET
```

```
GORE: ; ---- POMERANJE NAGORE ----
CALL AUX
MOV R6,#4
UPB: CALL IFBRK
MOV DPTR,#BUFFER+255-64 ; 4x POMERA BAFER GORE
MOV P2,#HIGH BUFFER
MOV R0,#LOW BUFFER+255-32
CALL SHIFTUP
CALL CLR0 ; I BRISE DONJI RED BAFERA
DJNZ R6,UPB
```

```
MOV R6,#7
UPAB: CALL IFBRK
MOV DPTR,#BUFFER+255-64 ; 7x POMERA BAFER GORE...
MOV P2,#HIGH BUFFER
MOV R0,#LOW BUFFER+255-32
CALL SHIFTUP
```

```
MOV DPTR,#AUXBUF+6*32 ; ...PREMESTA GORNJI RED AUXBUF-a...
MOV P2,#HIGH BUFFER
MOV R0,#LOW BUFFER
MOV R4,#32
CALL GD2
```

```
MOV DPTR,#AUXBUF+255-64 ; ...I POMERA AUXBUF GORE
MOV P2,#HIGH AUXBUF
MOV R0,#LOW AUXBUF+255-32
CALL SHIFTUP2
DJNZ R6,UPAB
RET
```

```
SHIFTUP: CALL SYNC4
SHIFTUP2: MOV R4,#6*32
GG2: MOVX A,@DPTR
DEC DPL
MOVX @R0,A
DEC R0
DJNZ R4,GG2
RET
```

```
LEVO: ; ---- POMERANJE NALEVO ----
RET
```

Re: 8031 question

```
TRIK1: ; ---- ODJEDNOM ----  
CALL AUX  
JMP ODJED
```

```
TRIK2: ; ---- SLOVO PO SLOVO ----  
CALL CLR7  
CALL AUX  
MOV R7,#WIDTH  
MOV R5,#0 ; CHAR COUNTER  
GOSLSL2: MOV R4,#8  
GOSLSL: CALL IFBRK  
CALL SYNC ; PRVO SLOVO BAFERA NADOLE  
CALL SYNC  
MOV A,R5  
MOV R0,A  
MOV P2,#HIGH BUFFER  
ADD A,#32  
MOV DPL,A  
MOV DPH,#HIGH BUFFER  
CALL SLOVODN
```

```
MOV A,R5 ; JEDAN BAJT IZ AUXBUF-a (DOLE) U BUFFER (GORE)  
MOV DPL,A  
MOV DPH,#HIGH AUXBUF  
ADD A,#7*32  
MOV R0,A  
MOVX A,@DPTR  
MOVX @R0,A
```

```
MOV A,R5 ; PRVO SLOVO AUXBUF-a NADOLE  
MOV R0,A  
MOV P2,#HIGH AUXBUF  
ADD A,#32  
MOV DPL,A  
MOV DPH,#HIGH AUXBUF  
CALL SLOVODN  
DJNZ R4,GOSLSL  
INC R5  
DJNZ R7,GOSLSL2  
RET
```

```
ADJDN: MOV R6,#1  
JMP XDN  
SLOVODN: MOV R6,#7  
GOSD: MOVX A,@DPTR  
MOVX @R0,A  
XDN: MOV A,DPL  
ADD A,#32  
MOV DPL,A  
MOV A,R0
```

```
ADD A,#32
MOV R0,A
DJNZ R6,GOSD
RET
```

```
TRIK3: ; --- MASKA NALEVO ---
CALL AUX
MOV DPTR,#AUXBUF+WIDTH-1
MOV R0,#LOW BUFFER+WIDTH-1
MOV P2,#HIGH BUFFER
MOV R6,#WIDTH
```

```
GOT3: CALL IFBRK
MOV R4,#0C0H
CALL MASKR4
MOV R4,#30H
CALL MASKR4
MOV R4,#0CH
CALL MASKR4
MOV R4,#3
CALL MASKR4
```

```
DEC DPL
DEC R0
DJNZ R6,GOT3
RET
```

```
MASKR4: CALL SYNC
MOV R3,#8
GOM4: MOVX A,@DPTR
ANL A,R4
MOV B,A
MOVX A,@R0
CPL A
ORL A,R4
CPL A
ORL A,B
MOVX @R0,A
MOV A,R0
ADD A,#32
MOV R0,A
MOV A,DPL
ADD A,#32
MOV DPL,A
DJNZ R3,GOM4
RET
```

```
TRIK4: ; --- MASKA NADESNO ---
CALL AUX
MOV DPTR,#AUXBUF
MOV R0,#LOW BUFFER
```

```
MOV P2,#HIGH BUFFER
MOV R6,#WIDTH
```

```
GOT4: CALL IFBRK
MOV R4,#3
CALL MASKR4
MOV R4,#0CH
CALL MASKR4
MOV R4,#30H
CALL MASKR4
MOV R4,#0C0H
CALL MASKR4
```

```
INC DPL
INC R0
DJNZ R6,GOT4
RET
```

```
TRIK5: ; --- CIK-CAK MASKA ---
CALL AUX
MOV DPTR,#AUXBUF+6*32
CALL UDESNO
MOV DPTR,#AUXBUF+5*32+WIDTH-1
CALL ULEVO
MOV DPTR,#AUXBUF+4*32
CALL UDESNO
MOV DPTR,#AUXBUF+3*32+WIDTH-1
CALL ULEVO
MOV DPTR,#AUXBUF+2*32
CALL UDESNO
MOV DPTR,#AUXBUF+1*32+WIDTH-1
CALL ULEVO
MOV DPTR,#AUXBUF ; ...
```

```
UDESNO: MOV R0,DPL
MOV P2,#HIGH BUFFER
MOV R5,#WIDTH
GOUES: CALL SYNC
CALL IFBRK
MOVX A,@DPTR
INC DPL
MOVX @R0,A
INC R0
DJNZ R5,GOUES
RET
```

```
ULEVO: MOV R0,DPL
MOV P2,#HIGH BUFFER
MOV R5,#WIDTH
GOULEVO: CALL SYNC
CALL IFBRK
```

```
MOVX A,@DPTR
DEC DPL
MOVX @R0,A
DEC R0
DJNZ R5,GOULEVO
RET
```

```
TRIK6: ; --- POKRET GORE-DOLE ---
CALL CLRAUX
CALL AUX
CALL CLR7
MOV R3,#8
GOT6: CALL SYNC4
CALL SYNC4
CALL IFBRK
MOV R4,#WIDTH/2
MOV DPTR,#BUFFER+6*32 ; PRVO SLOVO NAGORE
MOV R0,#LOW BUFFER+7*32
ZUB: MOV P2,#HIGH BUFFER
CALL SLOVOUP
PUSH 0
PUSH DPL
```

```
MOV DPH,#HIGH AUXBUF ; PRVO SLOVO PRENOS 1 BAJTA IZ AUX
MOVX A,@DPTR
MOVX @R0,A
CALL ADJUP
MOV P2,#HIGH AUXBUF
CALL SLOVOUP ; I AUXBUF NAGORE
CALL ADJUP
```

```
POP 0 ; DRUGO SLOVO NADOLE
POP DPL
CALL ADJDN
INC R0
INC DPL
MOV DPH,#HIGH BUFFER
MOV P2,#HIGH BUFFER
CALL SLOVODN
```

```
MOV DPH,#HIGH AUXBUF ; DRUGO SLOVO PRENOS 1 BAJTA IZ AUX
MOVX A,@DPTR
MOVX @R0,A
CALL ADJDN
MOV P2,#HIGH AUXBUF
CALL SLOVODN ; I AUXBUF NADOLE
```

```
MOV A,DPL
ADD A,#6*32+1
MOV DPL,A
```

```
INC R0
MOV DPH,#HIGH BUFFER
DJNZ R4,ZUB
DJNZ R3,GOT6
RET
```

```
ADJUP: MOV R6,#1
JMP XUP
SLOVOUP: MOV R6,#7
GOSU: MOVX A,@DPTR
MOVX @R0,A
XUP: MOV A,DPL
ADD A,#-32
MOV DPL,A
MOV A,R0
ADD A,#-32
MOV R0,A
DJNZ R6,GOSU
RET
```

```
TRIK7: ; ---- POKRETNA MASKA IZ SREDINE ----
CALL AUX
MOV R5,#8*WIDTH/2
GOT7: CALL IFBRK
MOV DPL,#WIDTH/2-1
CALL SHLPOLA
MOV DPL,#WIDTH/2
CALL SHRPOLA2
DJNZ R5,GOT7
RET
```

```
SHLPOLA: CALL SYNC
SHLPOLA2: MOV R6,#7
GOSHLP: MOV R7,#WIDTH/2
MOV DPH,#HIGH AUXBUF
PUSH DPL ; \
GOSHLP1: MOVX A,@DPTR
RRC A
MOVX @DPTR,A
DEC DPL
DJNZ R7,GOSHLP1
MOV R7,#WIDTH/2
MOV DPH,#HIGH BUFFER
POP DPL ; /
PUSH DPL ; \
GOSHLP2: MOVX A,@DPTR
RRC A
MOVX @DPTR,A
DEC DPL
DJNZ R7,GOSHLP2
```

```
POP ACC ; /  
ADD A,#32  
MOV DPL,A  
DJNZ R6,GOSHLP  
RET
```

```
TRIK8: ; --- POKRETNA MASKA KA SREDINI ---  
CALL AUX  
MOV R5,#8*WIDTH/2  
GOT8: CALL IFBRK  
MOV DPL,#0  
CALL SHRPOLA  
MOV DPL,#WIDTH-1  
CALL SHLPOLA2  
DJNZ R5,GOT8  
RET
```

```
SHRPOLA: CALL SYNC  
SHRPOLA2: MOV R6,#7  
GOSHRP: MOV R7,#WIDTH/2  
MOV DPH,#HIGH AUXBUF  
PUSH DPL ; \  
GOSHRP1: MOVX A,@DPTR  
RLC A  
MOVX @DPTR,A  
INC DPL  
DJNZ R7,GOSHRP1  
MOV R7,#WIDTH/2  
MOV DPH,#HIGH BUFFER  
POP DPL ; /  
PUSH DPL ; \  
GOSHRP2: MOVX A,@DPTR  
RLC A  
MOVX @DPTR,A  
INC DPL  
DJNZ R7,GOSHRP2  
POP ACC ; /  
ADD A,#32  
MOV DPL,A  
DJNZ R6,GOSHRP  
RET
```

```
TRIK9: ; --- RAFALNA GRAFIKA ---  
CALL AUX  
MOV R7,#0 ; HASH COUNTER  
GOT9: CALL IFBRK  
MOV DPTR,#HASH  
MOV A,R7  
MOVC A,@A+DPTR
```

Re: 8031 question

```
MOV R4,A ; RANDOM NUMBER IN R4
ANL A,#7
MOV DPTR,#BITTAB
MOVC A,@A+DPTR
XCH A,R4 ; RANDOM BIT IN R4
RR A
RR A
RR A
ANL A,#1FH
MOV DPTR,#NETTAB
MOVC A,@A+DPTR
MOV DPL,A
MOV A,R7
JB ACC.0,NSY
CALL SYNC
NSY: CALL PIXEL8
DJNZ R7,GOT9
RET
```

```
NETTAB: DB 00H,01H,02H,03H,20H,21H,22H,23H
DB 40H,41H,42H,43H,60H,61H,62H,63H
DB 80H,81H,82H,83H,0A0H,0A1H,0A2H,0A3H
DB 0C0H,0C1H,0C2H,0C3H,0E0H,0E1H,0E2H,0E3H
```

```
BITTAB: DB 1,2,4,8,10H,20H,40H,80H
```

```
PIXEL8: MOV R5,#8
PIX8: MOV DPH,#HIGH AUXBUF
MOVX A,@DPTR
ANL A,R4
MOV B,A
MOV DPH,#HIGH BUFFER
MOVX A,@DPTR
CPL A
ORL A,R4
CPL A
ORL A,B
MOVX @DPTR,A
INC DPL
INC DPL
INC DPL
INC DPL
DJNZ R5,PIX8
RET
```

```
TRIK0: ; --- LETECA SLOVA ---
CALL CLRBUF
CALL AUX
MOV R7,#WIDTH ; CHAR COUNT
MOV R6,#0 ; CHAR No.
```

Re: 8031 question

Re: 8031 question

```
GOT0: MOV DPL,R6
CALL AUXDZEP ; PRVO U DZEP
JZ BIOBLK ; NEMA LETECEG BLANKA
MOV A,R7
MOV B,#8
MUL AB
MOV R4,A
CALL SHIFTR6
BIOBLK: INC R6
DJNZ R7,GOT0
RET
```

```
SHIFTR6: ; SHIFT NALEVO BAFER OD R6 DO WIDTH, R4 PUTA
MOV R5,#7
MOV DPH,#HIGH BUFFER
MOV DPL,#WIDTH
MOV A,R4
JB ACC.0,GOSR6P
CALL SYNC
GOSR6P: CALL IFBRK ; SHIFT ZA 1 MESTO
PUSH DPL
MOV A,R7
INC A
MOV R3,A
CLR C
GOSR6: MOVX A,@DPTR ; SHIFT ZA 1 HORIZONTALNU LINIJU
RRC A
MOVX @DPTR,A
DEC DPL
DJNZ R3,GOSR6 ; KRAJ SHIFTOVANJA ZA 1 HORIZONTALNU LINIJU
POP ACC
ADD A,#32
MOV DPL,A
DJNZ R5,GOSR6P ; KRAJ SHIFTOVANJA ZA 1 MESTO
DJNZ R4,SHIFTR6 ; KRAJ SHIFTOVANJA ZA R4 MESTA
RET
```

```
AUXDZEP: MOV R5,#7 ; IZ (AUXBUF DPL) U DZEP BAFERA
MOV R2,#0
MOV DPH,#HIGH AUXBUF
MOV R0,#WIDTH ; POZICIJA DZEPA
MOV P2,#HIGH BUFFER
GOAD: MOVX A,@DPTR
MOVX @R0,A
ORL A,R2
MOV R2,A
MOV A,R0
ADD A,#32
MOV R0,A
MOV A,DPL
ADD A,#32
```

Re: 8031 question

```
MOV DPL,A
DJNZ R5,GOAD
MOV A,R2 ; A=0 AKO JE BIO BLANK
RET
```

```
PAUZA: ; ---
CALL IFNUM
MOV R3,#30 ; ZA 3 SEKUNDE DEFAULT
JC GOPAU
MOV R3,#5
JZ GOPAU ; AKO JE A=0, ONDA JE PAUZA 0.5 SEC
MOV B,#10
MUL AB ; A = 10 TO 90 STEP 10
```

```
MS100: MOV R3,A ; PAUZA A*100 mS, GO EDIT AKO SE PRITISNE ESC
GOPAU: MOV R2,#5 ; PAUZA R3*100 mS, GO EDIT AKO SE PRITISNE ESC
GOPAU2: CALL SYNC
DJNZ R2,GOPAU2 ; INNER LOOP 100 mS
CALL IFBRK
DJNZ R3,GOPAU
RET
```

```
IFBRK: CLR A ; IF ESC, GO TO EDIT
XCH A,JUST
JNZ QUITE
XCH A,JUST+1
JZ NOQUITE
QUITE: CJNE A,#1,NOQUITE
JMP EDIT
NOQUITE: RET
```

```
BLINK: ; ---
CALL BUFAUX
CALL IFNUM
MOV R4,#4 ; ZA 4 BLINKA DEFAULT
JC STBLINK
MOV R4,#10
JZ STBLINK ; AKO JE A=0, ONDA IMA 10 BLINKOVA
MOV R4,A ; R3 = 1 TO 9
STBLINK: MOV A,#8
CALL MS100 ; PRVO MALO PAUZE
GOBLINK: CALL CLRBUF
CALL MS400 ; OFF PAUZA
CALL ODJED
CALL MS400 ; ON PAUZA
DJNZ R4,GOBLINK
MS400: MOV A,#4
JMP MS100 ; JOS MALO PAUZE NA KRAJU
```

```
AUX: ; PUNI AUX BAFER SA POZICIJE (RPTR)
MOV R6,#WIDTH
```

```
MOV R2,#0 ; WRITE BUFFER POINTER
MOV DPL,RPTR ; READ MEM POINTER
MOV DPH,RPTR+1
MOVX A,@DPTR
CJNE A,#'K'+0A0H,NCK ; 'K' = DATUM LATINICOM
MOV TEMPTRIK,A
CALL DATUMAUX
SF2: MOV DPL,RPTR
MOV DPH,RPTR+1
INC DPTR
JMP SPECFIL
```

```
NCK: CJNE A,#'S'+0A0H,NCS ; 'S' = DATUM CIRILICOM
MOV TEMPTRIK,A
CALL DAT2AUX
JMP SF2
```

```
NCS: CJNE A,#'H'+0A0H,NCH ; 'H' = DATUM HRVATSKI
MOV TEMPTRIK,A
CALL DATHAUX
JMP SF2
```

```
NCH: CJNE A,#'B'+0A0H,NCB ; 'B' = DATUM NUMERICKI
MOV TEMPTRIK,A
CALL DATNAUX
JMP SF2
```

```
NCB: CJNE A,#'V'+0A0H,NCV ; 'V' = VREME
MOV TEMPTRIK,A
CALL HMAUX
JMP SF2
```

```
NCV: CJNE A,#'C'+0A0H,GOAUX ; 'C' = TEMPERATURA
MOV TEMPTRIK,A
CALL WAITSEC
CALL TEMPAUX
JMP SF2
```

```
GOAUX: CALL CPDPET ; RPTR – ENDTXT
JC OSTACHX
MOV A,#' ' ; BLANK AKO JE IZLETEO IZ TEKSTA ...
JMP NOVTRIK ; ... I NE UVECAVA SE DPTR
OSTACHX: MOVX A,@DPTR
CJNE A,#0C0H,$+3
JNC YNOUT
JMP NENOV
YNOUT:
; POCETAK OBRADJE REFRENA U TRIKU -----
```

```
CJNE A,#'O'+0A0H,NIXCO
CALL IFBRK ; CTRL O
```

```
INC DPTR
MOV A,REFRENO
ORL A,REFRENO+1
JZ IGNORE5 ; NEMA REFRENA
MOV A,RETADRO
ORL A,RETADRO+1
JNZ IGNORE5 ; NIJE SE VRATIO SA PRETHODNOG
MOV RETADRO,DPL
MOV RETADRO+1,DPH
MOV DPL,REFRENO
MOV DPH,REFRENO+1
IGNORE5: JMP GOAUX
```

```
NIXCO: CJNE A,#('+0A0H,NIXCOZ
INC DPTR
MOV REFRENO,DPL
MOV REFRENO+1,DPH
JMP GOAUX
```

```
NIXCOZ: CJNE A,#')'+0A0H,NIXCZZ
INC DPTR
MOV A,REFRENO
ORL A,REFRENO+1
JZ IGNORE6 ; NEMA REFRENA
MOV A,RETADRO
ORL A,RETADRO+1
JZ IGNORE6 ; NEMA GDE DA SE VRATI
MOV DPL,RETADRO
MOV DPH,RETADRO+1
MOV RETADRO,#0
MOV RETADRO+1,#0
IGNORE6: JMP GOAUX
```

```
NIXCZZ: CJNE A,#'X'+0A0H,NIXCX
CALL IFBRK
INC DPTR
MOV A,REFRENX
ORL A,REFRENX+1
JZ IGNORE7 ; NEMA REFRENA
MOV A,RETADRX
ORL A,RETADRX+1
JNZ IGNORE7 ; NIJE SE VRATIO SA PRETHODNOG
MOV RETADRX,DPL
MOV RETADRX+1,DPH
MOV DPL,REFRENX
MOV DPH,REFRENX+1
IGNORE7: JMP GOAUX
```

```
NIXCX: CJNE A,#<'+0A0H,NIXCM
INC DPTR
MOV REFRENX,DPL
```

Re: 8031 question

```
MOV REFRENX+1,DPH
JMP GOAUX
```

```
NIXCM: CJNE A,#'>'+0A0H,NIXCV
INC DPTR
MOV A,REFRENX
ORL A,REFRENX+1
JZ IGNORE8 ; NEMA REFRENA
MOV A,RETADRX
ORL A,RETADRX+1
JZ IGNORE8 ; NEMA GDE DA SE VRATI
MOV DPL,RETADRX
MOV DPH,RETADRX+1
MOV RETADRX,#0
MOV RETADRX+1,#0
IGNORE8: JMP GOAUX
; KRAJ OBRADJE REFRENA U TRIKU -----
```

```
NIXCV: CJNE A,#'N'+0A0H,BCN ; CTRL-N TREBA SAMO IGNORISATI
IGNCOM: INC DPTR
JMP GOAUX
BCN: MOV A,#' ' ; BLANK AKO JE DOSAO DO NOVOG TRIKA ...
JMP NOVTRIK ; ... I NE UVECAVA SE DPTR
NENOV: INC DPTR
NOVTRIK: PUSH DPL
PUSH DPH
CALL WRAUX ; UPISUJE CHAR U AUXBUF
POP DPH
POP DPL
INC R2
DJNZ R6,GAREL
```

```
SPECFIL: CALL WRAP
MOV RPTR,DPL
MOV RPTR+1,DPH
RET
```

```
GAREL: JMP GOAUX
```

```
; ; sabrutine
WAITSEC: ; VRACA SE POSLE CELE SEKUNDE
SETB NEWSEC
JB NEWSEC,$
RET
```

```
HMAUX: ; ISTO KAO VREMEAUX, SAMO BEZ SEKUNDI
CALL CLRAUX
MOV R2,#(WIDTH-8)/2
MOV R1,#RTC+5 ; H 10
CALL PRINT2
MOV A,#:'
```

Re: 8031 question

```
CALL PT
CALL PRINT2
MOV A,#':'
JMP PT
```

```
VREMEBL: MOV R5,#' '
JMP VREMER5
VREMEAUX: MOV R5,#':' ; UPIS VREMENA U AUXBUF
VREMER5: CALL CLRAUX
MOV R2,#0
; MOV P2,#HIGH AUXBUF
; MOV R0,#LOW AUXBUF
MOV DPTR,#VRIJEME
MOV R6,#8
CALL PRINT ; STAMPA PORUKU "VRIJEME"
MOV R2,#(WIDTH-8)/2+4 ; DVA REDA
MOV R1,#RTC+5 ; H 10
CALL PRINT2
MOV A,R5
CALL PT
CALL PRINT2
MOV A,R5
CALL PT
CALL PRINT2
RET
```

```
VRIJEME: DB 'VRIJEME:'
```

```
TEMPOR: DB 'TEMPER:'
```

```
TEMPAUX: ; UPIS TEMPERATURE U AUXBUF
CALL CLRAUX
MOV R2,#0
MOV DPTR,#TEMPOR
MOV R6,#7
CALL PRINT ; STAMPA PORUKU "TEMPER"
```

```
MOV R2,#(WIDTH-7) ; DVA REDA
```

```
MOV A,TEMPC
MOV R6,A
JZ NEMAZNAK
MOV C,ACC.7
MOV A,#+'
JNC OSTAPLUS
MOV A,R6
CPL A
INC A
MOV R6,A ; R6 = -R6
MOV A,#'-'
```

```
JMP OSTAPLUS
NEMAZNAK: MOV A,#' '
OSTAPLUS: CALL PT ; PREDZNAK TEMPERATURE
MOV A,R6
MOV B,#10
DIV AB
PUSH B
CALL PRINTA ; 10 C
POP ACC
CALL PRINTA ; 1 C
MOV A,#60H
CALL PT ; DEGREE SIGN
MOV A,#'C'
JMP PT
```

```
DATNAUX: SETB DCL ; UPIS DATUMA NUMERICKI
SETB DCH
JMP DAT12
```

```
DATHAUX: CLR DCL ; UPIS DATUMA HRVATSKI
SETB DCH
JMP DAT12
```

```
DAT2AUX: SETB DCL ; UPIS DATUMA CIRILICOM
CLR DCH
JMP DAT12
```

```
DATUMAUX: CLR DCL ; UPIS DATUMA U AUXBUF
CLR DCH
DAT12: CALL CLRAUX
MOV A,#WIDTH
CJNE A,#13,$+3
JC LTHDW ; NE MOZE DA STANE DAY-OF-WEEK
MOV R2,#0
MOV A,RTC+6 ; DAY-OF-WEEK
MOV B,#7
MUL AB
MOV DPTR,#DAYWEEK
JNB DCL,DAT12A
JB DCH,DAT12A
MOV DPTR,#DAYWEEK2
DAT12A: MOV R6,#7
CALL PRINT ; STAMPA DAY-OF-WEEK
MOV R2,#8
JMP GOONDAT
LTHDW: MOV R2,#(WIDTH-10)/2
GOONDAT: MOV R1,#RTC+8 ; 10 DANA
CALL PRINT2
CALL TACKA
MOV A,RTC+10 ; 10 MESECI
MOV B,#10
```

```
MUL AB
ADD A,RTC+9 ; 1 MESEC
MOV B,#4
MUL AB
MOV DPTR,#MONTHS3
JB DCH,DAT12X
MOV DPTR,#MONTHS
JNB DCL,DAT12B
MOV DPTR,#MONTHS2
JMP DAT12B
DAT12X: JNB DCL,DAT12B
MOV DPTR,#MONTHS4
MOV R6,#3
JMP SKIPR6
DAT12B: MOV R6,#4
SKIPR6: CALL PRINT ; STAMPA MESEC
MOV R1,#RTC+12 ; 10 GODINA
JMP PRINT2

TACKA: MOV A,#' '
JMP PT

PRINT2: CALL PRINT1
PRINT1: ; SA (R1) PRETVARA U ASCII I UPISUJE U AUXBUF
MOV A,@R1
DEC R1
PRINTA: ORL A,#30H ; UPISUJE A (ASCII NUMERIK) U AUXBUF (R2)
PT: PUSH DPL
PUSH DPH
CALL WRAUX
POP DPH
POP DPL
INC R2
RET

PRINT: ; MOVE FROM C DPTR+A TO X R0 P2, FORMAT R6
PUSH ACC
MOVC A,@A+DPTR
PUSH DPL
PUSH DPH
CALL WRAUX
POP DPH
POP DPL
INC R2
POP ACC
INC A
DJNZ R6,PRINT
RET

SHOWMEM: ; REZIM POKAZIVANJA STANJA MEMORIJE
CALL CLRBUF
```

```
CALL MOVABET
MOV DPTR,#2000H
CALL DPMINAB
MOV R2,#0
MOV A,#'M'
CALL WRBUFPP
MOV A,#'E'
CALL WRBUFPP
MOV A,#'M'
CALL WRBUFPP
CALL IFBL
MOV A,#'='
CALL WRBUFPP
CALL IFBL
```

```
MOV R3,#0'
L1000: MOV A,#HIGH 1000
MOV B,#LOW 1000
CALL CPDPAB
JC HILJADE
MOV A,#HIGH 1000
MOV B,#LOW 1000
CALL DPMINAB
INC R3
JMP L1000
HILJADE: MOV A,R3
CALL WRBUFPP ; HILJADE
```

```
MOV R3,#0'
L100: MOV A,#HIGH 100
MOV B,#LOW 100
CALL CPDPAB
JC STOTINE
MOV A,#HIGH 100
MOV B,#LOW 100
CALL DPMINAB
INC R3
JMP L100
STOTINE: MOV A,R3
CALL WRBUFPP ; STOTINE
```

```
MOV A,DPL
MOV B,#10
DIV AB
PUSH B
ORL A,#30H
CALL WRBUFPP ; DESETICE
```

```
POP ACC
ORL A,#30H
CALL WRBUF ; JEDINICE
```

```
MOV A,#25
CALL MS100
RET
```

```
WRBUFPP: PUSH DPL
PUSH DPH
CALL WRBUF
POP DPH
POP DPL
INC R2
RET
```

```
IFBL: MOV A,#WIDTH
CJNE A,#10,$+3
JC MALOMEM
MOV A,#' '
CALL WRBUFPP
MALOMEM: RET
```

```
DPMINAB: ; DPTR = DPTR - A hi B lo
PUSH ACC
MOV A,DPL
CLR C
SUBB A,B
MOV DPL,A
MOV A,DPH
POP B
SUBB A,B
MOV DPH,A
RET
```

```
ADJTIME: ; REZIM PODESAVANJA VREMENA
SETB INHIBIT
PUSH EPTR
PUSH CURSOR
GOADJH: MOV CURSOR,#0
GOADJE: MOV EPTR,#0
```

```
ADJFARM: CALL ADJADJ
CALL ADJSCR
CALL ODJED
CLR A
XCH A,JUST
JNZ NOSICHA
XCH A,JUST+1
JZ ADJFARM
NOSICHA: CALL MALIKEY
CJNE A,#9FH,NEBKSP
JMP ADJL ; BKSP PRETVARA U STRELICU LEVO
NEBKSP: CJNE A,#84H,NEADJL
; LEVO
```

```
ADJL: MOV DPTR,#ADJLEFT
SKRADJ: MOV A,CURSOR
MOVC A,@A+DPTR
MOV CURSOR,A
JMP ADJFARM
```

```
NEADJL: CJNE A,#86H,NEADJR
; DESNO
ADJR: MOV DPTR,#ADJRRIGHT
JMP SKRADJ
```

```
NEADJR: CJNE A,#88H,NEADJU
; NAGORE (ZA DAN U NEDELJI)
MOV A,CURSOR
JNZ AF1
INC RTC+6
MOV A,RTC+6
CJNE A,#7,$+3
JC AF1
MOV RTC+6,#0
AF1: JMP ADJFARM
```

```
NEADJU: CJNE A,#82H,NEADJD
; NADOLE (ZA DAN U NEDELJI)
MOV A,CURSOR
JNZ AF2
DEC RTC+6
MOV A,RTC+6
CJNE A,#7,$+3
JC AF2
MOV RTC+6,#6
AF2: JMP ADJFARM
```

```
NEADJD: CJNE A,#87H,NEADJH
; HOME
JMP GOADJH
```

```
NEADJH: CJNE A,#81H,NEADJE
; END
MOV CURSOR,#22
JMP GOADJE
```

```
NEADJE: CJNE A,#80H,NEADJINS
; INSERT = UPIS
INPTIME: CLR EA
```

```
MOV A,RTC+5
ANL A,#3
ORL A,#8
MOV RTC+5,A
MOV A,RTC+8
```

```
ANL A,#3
MOV RTC+8,A
MOV A,RTC+10
ANL A,#1
MOV RTC+10,A

MOV R7,#13 ; NIBBLES TO WRITE
MOV R0,#RTC
MOV R6,#0
SETB P3.5 ; DISABLE DECODER
WRTCLOOP:
MOV A,#0C0H ; ADRESA REGISTRA ZA SEKUNDE RTC
ORL A,R6
MOV P1,A
SETB P1.5 ; AW PULSE
CLR P1.5 ; AW LO
MOV A,#0C0H
ORL A,@R0
MOV P1,A ; BYTE OUT
CLR P1.6 ; ** WR PULSE **
NOP
SETB P1.6 ; ** WR OFF **
INC R0
INC R6
DJNZ R7,WRTCLOOP

SKRA2: POP CURSOR
POP EPTR
CLR INHIBIT
SETB EA
JMP EDIT

NEADJINS: CJNE A,#8AH,NEADJDEL
; DEL = IGNORE
JMP SKRA2

NEADJDEL: CLR C
SUBB A,#30H
JC TOAF
CJNE A,#10,$+3
JNC TOAF
MOV B,A
MOV DPTR,#WRITE
MOV A,CURSOR
MOVC A,@A+DPTR
CJNE A,#6,NIJEW
JMP TOAF
NIJEW: CJNE A,#13,$+3
JNC TOAF
ADD A,#RTC
MOV R1,A
```

Re: 8031 question

```
MOV @R1,B ; UPIS U RTC FIELD
JMP ADJR ; POSLE TOGA SKACE DESNO
TOAF: JMP ADJFARM
```

```
; pos 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
; W W W -- D D . M M . G G -- H H : M M : S S
```

```
ADJRRIGHT: DB 5,5,5,5,5,6,8,8,9,11,11,12,15,15,15,16,18,18,19,21,21,22,22
ADJLEFT: DB 0,0,0,0,0,0,5,6,6,08,09,09,11,12,12,12,15,16,16,18,19,19,21
WRITE: DB 6,6,6,6,6,8,7,6,10,9,06,12,11,06,06,05,04,06,03,02,06,01,00
```

```
ADJADJ: ; PODESAVA EPTR PREMA CURSOR I WIDTH, ZA ADJ
MOV A,CURSOR
CLR C
SUBB A,EPTR
JNC NECE
MOV EPTR,CURSOR ; IF CURSOR<EPTR THEN EPTR=CURSOR
JMP AARET
NECE: CLR C
SUBB A,#WIDTH
JC AARET
MOV A,CURSOR
CLR C
SUBB A,#WIDTH-1
MOV EPTR,A ; IF (CURSOR-EPTR)>WIDTH THEN EPTR=CUR-WID+1
AARET: RET
```

```
ADJSCR: ; KAO SCREEN, SAMO ZA ADJUST CLOCK
MOV A,EPTR
CPL A
INC A
MOV R2,A ; R2 = -(EPTR)
MOV R6,#23 ; TOTAL CHARS
```

```
MOV A,RTC+6 ; DAN U NEDELJI
MOV B,#7
MUL AB
MOV DPTR,#DAYWEEK
JNB PISMO,OLAT
MOV DPTR,#DAYWEEK2
OLAT: MOV R3,#3
DWLOOP: PUSH ACC
MOVC A,@A+DPTR
PUSH DPL
PUSH DPH
CALL PADJ
POP DPH
POP DPL
POP ACC
INC A
```

Re: 8031 question

```
DJNZ R3,DWLOOP
MOV A,#' '
CALL PADJ
MOV A,#' '
CALL PADJ
MOV R1,#RTC+8 ; 10 DANA
CALL TWO
MOV A,#' '
CALL PADJ
INC R0
MOV R1,#RTC+10 ; 10 MESECI
CALL TWO
MOV A,#' '
CALL PADJ
INC R0
MOV R1,#RTC+12 ; 10 GODINA
CALL TWO
MOV A,#' '
CALL PADJ
MOV A,#' '
CALL PADJ
INC R0
MOV R1,#RTC+5 ; 10 SATI
CALL TWO
MOV A,#' '
CALL PADJ
INC R0
MOV R1,#RTC+3 ; 10 MINUTA
CALL TWO
MOV A,#' '
CALL PADJ
INC R0
MOV R1,#RTC+1 ; 10 SEKUNDI
CALL TWO
; SAD JOS SAMO CURSOR
JB FREERUN.4,NOC2 ; BRZINA BLINKA CURSORA
MOV R7,#7
MOV A,CURSOR
CLR C
SUBB A,EPTR ; JER JE RELATIVNA POZICIJA CURSORA!
MOV DPL,A
MOV DPH,#HIGH BUFFER
CURL2: MOVX A,@DPTR
ORL A,#3FH
MOVX @DPTR,A
MOV A,DPL
ADD A,#32
MOV DPL,A
DJNZ R7,CURL2
NOC2: RET
```

Re: 8031 question

```
TWO: CALL ONE
ONE: MOV A,@R1
DEC R1
ORL A,#30H ; ...
```

```
PADJ: CJNE R2,#WIDTH,$+3
JNC NOWB
CALL WRBUF ; POKAZUJE CHAR
NOWB: INC R2
RET
```

```
IFNUM2: ; KAO IFNUM, SAMO DEFAULT 5 SEC, 0 = 10 SEC
CALL IFNUM
MOV R4,#5 ; ZA 5 SEKUNDI DEFAULT
JC DEFNUM
MOV R4,#10
JZ DEFNUM ; AKO JE = 0, ONDA 10 SEC
MOV R4,A
DEFNUM: MOV A,R4
MOV B,#10
MUL AB
RET ; A*100 mS ILI R4*1 S
```

```
IFNUM: ; IF (RPTR)=ASCII NUMERIK, THEN A=NUM, INC RPTR
MOV DPL,RPTR
MOV DPH,RPTR+1
MOVX A,@DPTR
CLR C
SUBB A,#'0' ; TU POCINJU ASCII NUMERICI
JC NIJENUM
CJNE A,#10,$+3
CPL C
JC NIJENUM
PUSH ACC
CALL INCWRAP
MOV RPTR,DPL
MOV RPTR+1,DPH
POP ACC
CLR C
NIJENUM: RET ; C SET AKO NIJE NUMERIK, NC AKO JESTE
```

```
INCWRAP: INC DPTR
WRAP: ; IF DPTR=>(ENDTXT), THEN DPTR=#TEXT
PUSH ACC
CALL CPDPET
JC NOVADJ
MOV DPTR,#TEXT
NOVADJ: POP ACC
RET
```

```
MEM: ; VRACA KOLICINU MEMORIJE DPTR=ENDTXT-#TEXT
```

Re: 8031 question

```
CALL MOVDPET
MOV A,DPL
CLR C
SUBB A,#LOW TEXT
MOV DPL,A
MOV A,DPH
SUBB A,#HIGH TEXT
MOV DPH,A
RET
```

```
MALIKEY: ; PROSTA OBRADA SA TASTATURE ZA SPEC. NAMENE
MOV DPTR,#KEYTAB
CPL A
JNB ACC.7,RELEASE ; OTPUSTEN TASTER SE IGNORE
CLR ACC.7
CLR C
SUBB A,#27H
JC RELEASE
MOVC A,@A+DPTR
RET
RELEASE: CLR A
RET
```

```
KEY: ; OBRADA PRIMLJENOG ZNAKA SA TASTATURE
MOV DPTR,#KEYTAB
CPL A
CLR C
JB ACC.7,PRESSED
SUBB A,#27H ; OVDE JE OTPUSTEN TASTER
JC KEYZERO
MOVC A,@A+DPTR
CJNE A,#8BH,NOCAPL
CLR CAPSON ; OTPUSTEN CAPS LOCK
SJMP KEYZERO
NOCAPL: CJNE A,#8CH,NONUML
CLR NUMON ; OTPUSTEN NUM LOCK
SJMP KEYZERO
NONUML: CJNE A,#8DH,NOSCRL
CLR SCRON ; OTPUSTEN SCROLL LOCK
SJMP KEYZERO
NOSCRL: CJNE A,#9BH,NOSHIF
CLR SHIFLAG ; OTPUSTEN SHIFT
SJMP KEYZERO
NOSHIF: CJNE A,#9CH,NOCTRL
CLR CTRLFLAG ; OTPUSTEN CTRL
SJMP KEYZERO
NOCTRL: CJNE A,#9DH,NOALT
CLR ALTFLAG ; OTPUSTEN ALT
SJMP KEYZERO
NOALT: CJNE A,#80H,KEYZERO
CLR INSON ; OTPUSTEN INS
```

KEYZERO: CLR A  
RET

PRESSED: CLR ACC.7 ; OVDE JE PRITISNUT TASTER (ILI AUTOREPEAT)  
SUBB A,#27H  
JC KEYZERO ; <27H JE NEMOGUCE  
MOVC A,@A+DPTR  
JZ KEYZERO ; NEPOSTOJECI TASTER U TABLICI  
JNB ACC.7,NOSPEC  
; OVDE JE SPEC TASTER (>=80H)  
CLR ACC.7  
CJNE A,#(SPECEND-SPECTAB)/2,\$+3  
JNC KEYZERO ; VECI OD NAJVECEG KODA ZA SPEC TASTER  
MOV DPTR,#KEYZERO ; RET ADDRESS ...  
PUSH DPL  
PUSH DPH ; ... ON STACK  
MOV DPTR,#SPECTAB

EXECA: MOV B,#2  
MUL AB ; A\*2  
MOV B,A  
MOVC A,@A+DPTR ; HI BYTE OF ENTRY PT  
XCH A,B  
INC A  
MOVC A,@A+DPTR ; LO BYTE OF ENTRY PT  
PUSH ACC  
PUSH B  
RET ; SPEC KEY EXECUTION

NOSPEC: ; NIJE SPEC TASTER (<80H)  
JNB ALTFLAG,NPRALT  
; PRITISNUT ALT  
JMP NPRALT

GOKZ: JMP KEYZERO

NPRALT: JNB CTRLFLAG,NPRCTRL  
; PRITISNUT CTRL  
CJNE A,#'A',NOADJT  
CALL ADJTIME  
JMP GOKZ  
NOADJT: CJNE A,#'M',NOSHOMEM  
CALL SHOWMEM  
JMP GOKZ  
NOSHOMEM:  
CJNE A,#60H,\$+3  
JNC GOKZ ; VECI OD NAJVECEG KODA ZA CTRL TASTER  
CJNE A,#20H,\$+3  
JC GOKZ ; MANJI OD NAJMANJEG KODA ZA CTRL TASTER

Re: 8031 question

```
JNB SHIFLAG,CHANGED
; PRITISNUT SHIFT + CTRL
MOV R7,#(ZNAKEND-ZNAKTAB)/2
MOV B,A
MOV DPTR,#ZNAKTAB
LOOPSH2: CLR A ; TRAZI DA LI IMA SVOJ SHIFT ZNAK NA TASTERU
MOVC A,@A+DPTR
INC DPTR
CJNE A,B,SRCHSH2
CLR A ; FOUND!
MOVC A,@A+DPTR ; LOAD NEW (SHIFT!) CHAR: !@#$%^&*()_+:"<>?
CJNE A,#60H,$+3
JNC GOKZ
CJNE A,#20H,$+3
JC GOKZ
JMP CHANGED
SRCHSH2: INC DPTR
DJNZ R7,LOOPSH2
MOV A,B

CHANGED: MOV DPTR,#CTRLTAB-20H
MOVC A,@A+DPTR
CJNE A,#'-',COM
JMP KEYZERO
COM: ADD A,#0C0H-20H ; IMACE VREDNOST 0C0H-0FFH (TU SU KOMANDE)
RET

NPRCTRL: JNB SHIFLAG,NPRSH
; PRITISNUT SHIFT
MOV R7,#(ZNAKEND-ZNAKTAB)/2
MOV B,A
MOV DPTR,#ZNAKTAB
LOOPSH: CLR A ; TRAZI DA LI IMA SVOJ SHIFT ZNAK NA TASTERU
MOVC A,@A+DPTR
INC DPTR
CJNE A,B,SRCHSH
CLR A ; FOUND!
MOVC A,@A+DPTR ; LOAD NEW (SHIFT!) CHAR: !@#$%^&*()_+:"<>?
RET
SRCHSH: INC DPTR
DJNZ R7,LOOPSH
; NOT FOUND SHIFT SIGN
MOV A,B
JB CAPSFLAG,CASE ; PRIT SHIFT I AKTIVAN CAPS_LOCK - POTIRU SE
SJMP CIRLAT ; PRIT SAMO SHIFT - OSTAJE VELIKO SLOVO

NPRSH: JNB CAPSFLAG,CASE ; NIJE PRIT SHIFT I PASIVAN CAPS_LOCK
SJMP CIRLAT ; AKTIVAN CAPS_LOCK - OSTAJE VELIKO SLOVO

CASE: ; PROMENI VELIKO SLOVO U MALO
CJNE A,#60H,$+3
```

Re: 8031 question

Re: 8031 question

```
JNC CIRLAT ; > VEL SLOVO
CJNE A,#41H,$+3
JC CIRLAT ; < VEL SLOVO
ADD A,#20H ; PRETVORI VELIKO SLOVO U MALO
CIRLAT: JNB PISMO,RETKEY
CJNE A,#41H,$+3
JC RETKEY
ADD A,#40H ; CIRILICA
RETKEY: RET
```

```
BUFAUX: ; PREMESTA BUFFER U AUXBUF
MOV P2,#HIGH BUFFER
MOV DPTR,#AUXBUF+219
JMP SKROD
```

```
ODJED: ; PREMESTA AUXBUF U BUFFER
MOV P2,#HIGH AUXBUF
MOV DPTR,#BUFFER+219
CALL SYNC
SKROD: MOV R0,#219
GOODJED: MOVX A,@R0
MOVX @DPTR,A
DEC DPL
DJNZ R0,GOODJED
MOVX A,@R0
MOVX @DPTR,A
RET
```

```
LEFT8: CALL SHIFTL
MOV R7,#7
GOLEFT8: CALL FSHIFTL
DJNZ R7,GOLEFT8
RET
```

```
SYNC4: CALL SYNC
CALL SYNC
CALL SYNC
SYNC: SETB HANDS ; VRACA SE POSLE 2.85 ms HEARTBEATA
JB HANDS,$
RET
```

```
FSHIFTL: CALL SYNC
SHIFTL: ; ...POMERA LEVO BAFER ZA 1 PIXEL
MOV P2,#HIGH BUFFER
MOV R0,#220
CLR C
GOSHL: MOVX A,@R0
RRC A
MOVX @R0,A
DJNZ R0,GOSHL
```

Re: 8031 question

Re: 8031 question

```
MOVX A,@R0 ; DONJI LEVI BAJT NA EKRANU
RRC A
MOVX @R0,A
RET
```

```
RIGHT8: MOV R7,#8
GORIG8: CALL FSHIFTR
DJNZ R7,GORIG8
RET
```

```
FSHIFTR: CALL SYNC
SHIFTR: ; ...POMERA DESNO BAFER ZA 1 PIXEL
MOV P2,#HIGH BUFFER-1
MOV R0,#255
CLR C
MOVX A,@R0 ; DONJI BAJT LEVOG DZEPA
RLC A
MOVX @R0,A
INC R0
INC P2
MOV R6,#220
GOSHR: MOVX A,@R0
RLC A
MOVX @R0,A
INC R0
DJNZ R6,GOSHR
RET
```

```
WRDZEP: ; ISPISUJE CHAR (A) U DESNI DZEP BAFERA
MOV R0,#WIDTH
MOV P2,#HIGH BUFFER
JMP CONTWR
WRBUF: ; ISPISUJE CHAR (A) U BUFFER (R2) /100T/
MOV 0,2 ; R0,R2,R7,B,DPTR,P2
MOV P2,#HIGH BUFFER
JMP CONTWR
WRAUX: ; ISPISUJE CHAR (A) U AUXBUF (R2) /100T/
MOV 0,2 ; R0,R2,R7,B,DPTR,P2
MOV P2,#HIGH AUXBUF
CONTWR: MOV R7,#7
ONCE: MOV B,#7
MUL AB
ADD A,#LOW(CHARG1-7*20H-1)
MOV DPL,A
MOV A,B
ADDC A,#HIGH(CHARG1-7*20H-1)
MOV DPH,A ; SAD JE DPTR = POZICIJA ZNAKA U CHAR GEN-U
```

```
CHAR7: MOV A,R7 ; 1
MOVC A,@A+DPTR ; 2
MOVX @R0,A ; 2
```

Re: 8031 question

```
MOV A,R0 ; 1
ADD A,#32 ; 1
MOV R0,A ; 1
DJNZ R7,CHAR7 ; 2 10x7=70 T PETLJA
RET
```

```
CLRAUX: MOV P2,#HIGH AUXBUF ; BRISE AUXBUF
JMP SKRCLR
CLRBUF2: MOV P2,#HIGH BUFFER2 ; BRISE BUFFER2
JMP SKRCLR
CLRBUF: MOV P2,#HIGH BUFFER ; BRISE BUFFER
SKRCLR: CLR A
MOV R0,#0
GOCLR: MOVX @R0,A
DJNZ R0,GOCLR
RET
```

CLRBUF7:

```
CPCUET: ; (CURSOR) – (ENDTXT)
PUSH DPL
PUSH DPH
MOV DPL,CURSOR
MOV DPH,CURSOR+1
CALL CPDPET
POP DPH
POP DPL
RET
```

```
CPDPCU: ; DPTR – (CURSOR)
MOV A,CURSOR+1
MOV B,CURSOR
SJMP CPDPAB
```

```
CLRMEM: %MOVW(CURSOR,TEXT) ; BRISANJE EXTERNE MEMORIJE
%MOVW(EPTR,TEXT)
CLR OVERWR
MOV DPTR,#TEXT
; ...
MOVETDP: MOV A,DPH ; MOVX (ENDTXT), DPTR
MOV B,DPL
MOVETAB: MOV DPTR,#ENDTXT+1 ; MOVX (ENDTXT), A hi B lo
MOVX @DPTR,A
DEC DPL
MOV A,B
MOVX @DPTR,A
RET
```

```
MOVDPET: CALL MOVABET ; MOV DPTR, (ENDTXT)
MOV DPH,A
MOV DPL,B
```

RET

```
MOVABET: MOV DPTR,#ENDTXT ; MOV A hi B lo, (ENDTXT)
MOVX A,@DPTR
MOV B,A
INC DPTR
MOVX A,@DPTR
RET
```

```
CPDPET: ; DPTR – (ENDTXT)
PUSH DPL
PUSH DPH
CALL MOVABET
POP DPH
POP DPL ; ...
CPDPAB: ; DPTR – A hi B lo, C VALID, Z VALID
XCH A,DPH
CJNE A,DPH,NEQ
XCH A,DPH
MOV A,DPL
CLR C
SUBB A,B ; VISOKI BAJTOVI JEDNAKI: POREDNI NISKE
RET
```

```
NEQ: XCH A,DPH
MOV A,#0FFH ; DA NE BUDE A=0 AKO SU VISOKI NEJEDNAKI
RET
```

```
DECDPTR: ; DEC DPTR 16-bit
DEC DPL
MOV A,DPL
CJNE A,#0FFH,NB8OVF
DEC DPH
NB8OVF: RET
```

```
EXITMES: POP ACC
RET
```

```
MESTO: ; POMERA 1 MESTO NAGORE OD CURSOR DO ENDTXT
PUSH ACC ; I NA (CURSOR) UPISUJE ACC
JNB OVERWR,PUSHWR
; --- OVERWRITE MODE
CALL CPCUET ; CURSOR – ENDTXT
JC NEKRAJT
; CURSOR NA KRAJU TEKSTA
CALL MOVDPET
INC DPTR
MOV A,DPH
JB ACC.5,EXITMES ; NEMA SLOBODNE MEMORIJE
CALL MOVETDP
NEKRAJT: MOV DPL,CURSOR
```

```
MOV DPH,CURSOR+1
JMP HEREWR
```

```
PUSHWR: CALL MOVDPET ; --- PUSHWRITE MODE
INC DPTR
MOV A,DPH
JB ACC.5,EXITMES ; NEMA SLOBODNE MEMORIJE
PUSH DPL
PUSH DPH
CALL MOVETDP
POP DPH
POP DPL
```

```
GOMES: CALL DECDPTR
CALL CPDPCU ; DPTR – (CURSOR)
JZ HEREWR ; OVDE UPISUJE ZNAK
JC HEREWR
CALL DECDPTR
MOVX A,@DPTR
INC DPTR
MOVX @DPTR,A
JMP GOMES ; KRAJ PETLJE U KOJOJ PRAVI MESTO
```

```
HEREWR: POP ACC
MOVX @DPTR,A
INC DPTR
MOV CURSOR,DPL
MOV CURSOR+1,DPH ; ...
```

```
ADJUST: ; UTERUJE CURSOR U REGULARNE GRANICE MEMORIJE
; I U GRANICE VIDLJIVOG BAFERA (PODESAVA EPTR)
CALL CPCUET ; TEST 1: CURSOR – ENDTXT
JZ EQCUET ; (CURSOR) = (ENDTXT)
JC EQCUET ; (CURSOR) < (ENDTXT)
CALL MOVDPET ; IF CURSOR > ENDTXT THEN CURSOR = ENDTXT
MOV CURSOR,DPL
MOV CURSOR+1,DPH
```

```
EQCUET: MOV DPTR,#TEXT ; TEST 2: #TEXT – CURSOR
CALL CPDPCU ; (TEXT) – (CURSOR)
JC OKTXTCU
%MOVW(CURSOR,TEXT) ; IF CURSOR < #TEXT THEN CURSOR = #TEXT
```

```
OKTXTCU: MOV DPL,EPTR ; TEST 3: EPTR – CURSOR
MOV DPH,EPTR+1
CALL CPDPCU ; (EPTR) – (CURSOR)
JZ EQEPCU ; (EPTR) = (CURSOR)
JC EQEPCU ; (EPTR) < (CURSOR)
MOV EPTR,CURSOR ; IF EPTR > CURSOR THEN EPTR = CURSOR
MOV EPTR+1,CURSOR+1 ; (IZLETEO LEVO OD EKRANA)
```

Re: 8031 question

```
EQEPCU: MOV DPL,EPTR ; TEST 4: (EPTR+#WIDTH) – CURSOR
MOV DPH,EPTR+1
MOV A,#WIDTH
ADD A,DPL
MOV DPL,A
MOV A,DPH
ADDC A,#0
MOV DPH,A ; SAD JE DPTR = (EPTR) + #WIDTH
CALL CPDPCU ; (DPTR) – (CURSOR)
JNC EQDPCU ; (DPTR) >= (CURSOR)
MOV A,CURSOR ; IF DPTR < CURSOR THEN EPTR = CURSOR – #WIDTH
CLR C
SUBB A,#WIDTH
MOV EPTR,A
MOV A,CURSOR+1
SUBB A,#0
MOV EPTR+1,A
```

```
EQDPCU: MOV DPL,EPTR ; TEST 5: EPTR – TEXT
MOV DPH,EPTR+1
MOV A,#HIGH TEXT
MOV B,#LOW TEXT
CALL CPDPAB ; (EPTR) – #TEXT
JNC OKEPTXT ; (EPTR) >= #TEXT
%MOVW(EPTR,TEXT) ; IF EPTR < #TEXT THEN EPTR = #TEXT
OKEPTXT: RET
```

```
; ; edit
EDIT: CALL ADJUST
MOV TEMPTRIK,#0
```

```
KEYFARM: MOV SP,#STACK
CALL SYNC
CALL SCREEN
CLR A
XCH A,,JUST
JNZ NOSICH
XCH A,,JUST+1
JZ KEYFARM
NOSICH:
CALL KEY
JZ KEYFARM ; FALICAN TASTER
CALL MESTO
JMP KEYFARM
```

```
SCREEN: ; ISPISUJE EKRAN (EPTR), PODRZAVA (CURSOR)
MOV R6,#WIDTH
MOV R2,#0 ; WRITE BUFFER POINTER
MOV DPL,EPTR ; READ MEM POINTER
MOV DPH,EPTR+1
GOSCR: CALL CPDPET ; CURSOR – ENDTXT
```

Re: 8031 question

```
MOVX A,@DPTR
JC OSTACHAR
MOV A,#' ' ; BLANK AKO JE IZLETEO IZ TEKSTA
OSTACHAR: PUSH DPL
PUSH DPH
PUSH ACC
CALL WRBUF ; POKAZUJE CHAR
POP ACC
CJNE A,#0C0H,$+3
JC NECTR
JB FREERUN.4,NECTR ; BRZINA BLINKA PODVUCENOG CTRL KARAKTERA
```

```
MOV A,R0
ADD A,#-32*7
MOV R0,A
MOV A,#3FH ; CRTICA KOJA PODVLACI CTRL CHAR
MOVX @R0,A
```

```
NECTR: POP DPH
POP DPL
INC DPTR
INC R2
DJNZ R6,GOSCR
; SAD JOS SAMO CURSOR
JB FREERUN.4,NOCUR ; BRZINA BLINKA CURSORA
MOV R7,#0
MOV A,CURSOR
CLR C
SUBB A,EPTR
MOV R0,A
CJNE A,#WIDTH,NDI
; CURSOR VAN EKRANA DESNO: VIDI SE SAMO IVICA
DEC R0
MOV R7,#1
NDI: MOV P2,#HIGH BUFFER
```

```
MOV A,R7
MOV C,OVERWR
MOV ACC.1,C
MOV C,PISMO
MOV ACC.2,C
MOV R7,#7
MOV B,#7
MUL AB ; A=28*PISMO+14*OVERWR+7*(VAN EKRANA)
ADD A,#LOW(CURSORS-1)
MOV DPL,A
MOV A,B
ADDC A,#HIGH(CURSORS-1)
MOV DPH,A
```

```
OR7: MOV A,R7
```

Re: 8031 question

```
MOVC A,@A+DPTR
MOV B,A
MOVX A,@R0
ORL A,B
MOVX @R0,A
MOV A,R0
ADD A,#32
MOV R0,A
DJNZ R7,OR7
NOCUR: RET
```

```
; ; spec.tasteri
```

```
INSERT: ; ---- INS: PUSHWRITE / OVERWRITE ----
JNB CTRLFLAG,NODEMOT
CLR PISMO2
MOV DPTR,#TEXT ; CTRL-INS: UPIS DEMO TEKSTA
CALL CPDPET
JNZ RETINS ; NE PRIHVATA DEMO JER NIJE PRAZNA MEMORIJA
```

```
%MOVW(CURSOR,TEXT)
%MOVW(EPTR,TEXT)
MOV DPTR,#DEMOTXT
MOV P2,#HIGH TEXT
MOV R0,#LOW TEXT
GODEMO: CLR A
MOVC A,@A+DPTR
INC DPTR
JZ GOTDEMO ; 00 = TERMINATOR
CJNE A,#',,NOMETA
; KOMANDNI ZNAK: "/"
CLR A
MOVC A,@A+DPTR
INC DPTR
CJNE A,#',,NOKROZ ; "/" SE PREDSTAVLJA KAO "/"
JMP NOMETA
NOKROZ: CJNE A,#['',NOCIR ; [ = CIRILICA
SETB PISMO2
JMP GODEMO
NOCIR: CJNE A,#']',NOLAT ; ] = LATINICA
CLR PISMO2
JMP GODEMO
NOLAT: CJNE A,#60H,$+3
JC VELSL
ADD A,#-20H ; LOWERCASE ADJUST UPPERCASE
VELSL: ADD A,#0A0H ; TRICK CODE 0C0H-0FFH

NOMETA: CJNE A,#41H,$+3
JC NOLAT1
CJNE A,#80H,$+3
JNC NOLAT1
```

Re: 8031 question

```
JNB PISMO2,NOLAT1
ADD A,#40H ; KONVERTUJE LATINICU U CIRILICU
```

```
NOLAT1: PUSH DPL
PUSH DPH
PUSH ACC
CALL MOVDPET
POP ACC
MOVX @DPTR,A ; UPIS U EXT RAM
INC DPTR
CALL MOVETDP
POP DPH
POP DPL
JMP GODEMO
```

```
GOTDEMO: JMP RUN
```

```
NODEMOT: MOV C,OVERWR
CPL C
MOV OVERWR,C
RETINS: RET
```

```
GOEND: ; --- END ---
CALL MOVDPET
MOV CURSOR,DPL
MOV CURSOR+1,DPH
CALL EQPCU
JMP ADJUST
```

```
GODOWN: ; --- STRELICA NADOLE ---
MOV R3,#WIDTH
GOGD: CALL GORIGHT
DJNZ R3,GOGD
RET
```

```
PGDN: ; --- PAGE DOWN ---
MOV DPL,CURSOR
MOV DPH,CURSOR+1
GOPGDN: CALL CPDPET
JNC FOUNDPD
INC DPTR
MOVX A,@DPTR
CJNE A,#N'+0A0H,GOPGDN ; TRAZI "CTRL N" ILI KRAJ TEKSTA
FOUNDPD: MOV CURSOR,DPL
MOV CURSOR+1,DPH
MOV EPTR,DPL
MOV EPTR+1,DPH
JMP ADJUST
```

```
GOLEFT: ; --- STRELICA NALEVO ---
```

Re: 8031 question

```
DEC CURSOR
MOV A,CURSOR
CJNE A,#0FFH,NOVFH
DEC CURSOR+1
NOVFH: JMP ADJUST
```

```
PETICA: ; ---- PROMENA PISMA ----
MOV C,PISMO
CPL C
MOV PISMO,C
RET
```

```
GORIGHT: ; ---- STRELICA NADESNO ----
INC CURSOR
MOV A,CURSOR
JNZ NOVFH2
INC CURSOR+1
NOVFH2: JMP ADJUST
```

```
HOME: ; ---- HOME ----
%MOVW(EPTR,TEXT)
%MOVW(CURSOR,TEXT)
JMP ADJUST
```

```
GOUP: ; ---- STRELICA NAGORE ----
MOV R3,#WIDTH
GOGU: CALL GOLEFT
DJNZ R3,GOGU
RET
```

```
PGUP: ; ---- PAGE UP ----
MOV DPL,CURSOR
MOV DPH,CURSOR+1
GOPGUP: MOV A,#HIGH TEXT
MOV B,#LOW TEXT
CALL CPDPAB
JZ FOUNDPU
JC FOUNDPU
CALL DECDPTR
MOVX A,@DPTR
CJNE A,#N'+0A0H,GOPGUP ; TRAZI "CTRL N" ILI POCETAK TEKSTA
FOUNDPU: MOV CURSOR,DPL
MOV CURSOR+1,DPH
JMP ADJUST
```

```
DEL: ; ---- BRISANJE ZNAKA ----
JNB CTRLFLAG,NOCLRALL
JNB ALTFLAG,NOCLRALL
CALL CLRMEM ; CTRL-ALT-DEL: BRISANJE CELE MEMORIJE
JMP EDIT
```

Re: 8031 question

```
NOCLRALL: MOV A,CURSOR
CLR C
SUBB A,EPTR ; TO CE BITI RELATIVNA POZICIJA NA EKРАНU
CJNE A,#WIDTH,NTRIGHT
DELRET: RET ; NE DOZVOLJAVA BRISANJE ZNAKA KOJI SE NE VIDI
```

```
NTRIGHT: CALL CPCUET ; CURSOR – ENDTXT
JNC DELRET ; VEC JE NA KRAJU TEKSTA
```

```
MOV DPL,CURSOR
MOV DPH,CURSOR+1
GODEL: CALL CPDPET ; DPTR – ENDTXT
JNC VOZEND
```

```
INC DPTR
MOVX A,@DPTR
MOV R7,A
CALL DECDPTR
MOV A,R7
MOVX @DPTR,A
INC DPTR
JMP GODEL ; KRAJ PETLJE ZA POMERANJE TEKSTA NADOLE
```

```
VOZEND: CALL MOVDPET
CALL DECDPTR
CALL MOVETDP
JMP ADJUST
```

```
CAPLOCK: JB CAPSON,IGNCF ; ----
SETB CAPSON
CPL CAPSFLAG
IGNCF: RET
```

```
NUMLOCK: JB NUMON,IGNNUM ; ----
SETB NUMON
CPL NUMFLAG
IGNNUM: RET
```

```
SCRLOCK: JB SCRON,IGNSCR ; ----
SETB SCRON
CPL SCRFLAG
IGNSCR: RET
```

```
SHIFT: SETB SHIFLAG ; ----
RET
```

```
CTRL: SETB CTRLFLAG ; ----
RET
```

```
ALT: SETB ALTFLAG ; ----
```

Re: 8031 question

RET

SYSREQ: ; ---

RET

TAB: ; --- TAB ---

MOV CURSOR,RPTR

MOV CURSOR+1,RPTR+1

MOV A,CURSOR

CLR C

SUBB A,#WIDTH/2

MOV EPTR,A

MOV A,CURSOR+1

SUBB A,#0

MOV EPTR+1,A

JMP ADJUST

XF1:

XF2:

XF3:

XF4:

XF5:

XF6:

XF7:

XF8:

XF9:

XFA:

RET

NCTESC: JMP RUN

ESC: ; --- R U N M O D E ---

JNB CTRLFLAG,NCTESC

; CTRL-ESC

MOV DPTR,#TEXT

SRCHZM: MOVX A,@DPTR

CJNE A,#'+0A0H,NIXS1

MOV REFRENO,DPL ; NASAO CTRL (

MOV REFRENO+1,DPH

NIXS1: CJNE A,#'<'+0A0H,NIXS2

MOV REFRENX,DPL ; NASAO CTRL <

MOV REFRENX+1,DPH

NIXS2: INC DPTR

```
CALL CPDPCU
JC SRCHZM ; NIJE JOS UVEK DPTR=(CURSOR)
MOV RPTR,DPL
MOV RPTR+1,DPH
JMP RUN2
```

```
BKSP: ; --- BACKSPACE ---
MOV DPL,EPTR
MOV DPH,EPTR+1
CALL CPDPCU ; EPTR - CURSOR
JNZ NTOTLEFT
RET ; NE DOZVOLJAVA BRISANJE ZNAKA KOJI SE NE VIDI
NTOTLEFT: CALL GOLEFT
JMP DEL
```

ENTER:

RETIT: RET

; ; char gen

```
; gornji red +0 0 1 2 3 4 5 --
; +1 0 1 2 3 4 5 --
; +2 0 1 2 3 4 5 --
; +3 0 1 2 3 4 5 --
; +4 0 1 2 3 4 5 --
; +5 0 1 2 3 4 5 -- (eventualno postoji i bit 6
; donji red +6 0 1 2 3 4 5 -- ako slovo ima vecu sirinu)
```

CHARG1:

```
DB 00H,00H,00H,00H,00H,00H,00H ; blank
DB 0CH,0CH,0CH,0CH,0CH,00H,0CH ; !
DB 0AH,0AH,00H,00H,00H,00H,00H ; "
DB 36H,36H,7FH,36H,7FH,36H,36H ; #
DB 0CH,3EH,0DH,1EH,2CH,1FH,0CH ; dol
DB 23H,33H,18H,0CH,06H,33H,31H ; proc
DB 04H,0AH,04H,2AH,11H,29H,06H ; &
DB 04H,04H,00H,00H,00H,00H,00H ; '
DB 08H,04H,06H,06H,06H,04H,08H ; (
DB 04H,08H,18H,18H,18H,08H,04H ; )
DB 00H,12H,0CH,3FH,0CH,12H,00H ; *
DB 00H,08H,08H,3EH,08H,08H,00H ; +
DB 00H,00H,00H,00H,06H,04H,02H ; ,
DB 00H,00H,00H,3FH,00H,00H,00H ; -
DB 00H,00H,00H,00H,00H,06H,06H ; .
DB 20H,30H,18H,0CH,06H,03H,01H ; /
```

```
DB 1EH,33H,33H,33H,33H,33H,1EH ; 0
DB 08H,0CH,0EH,0CH,0CH,0CH,1EH ; 1
```

DB 1EH,33H,30H,18H,0CH,06H,3FH ; 2  
DB 1EH,33H,30H,1CH,30H,33H,1EH ; 3  
DB 03H,03H,1BH,1BH,3FH,18H,18H ; 4  
DB 3FH,03H,1FH,30H,30H,33H,1EH ; 5  
DB 1EH,33H,03H,1FH,33H,33H,1EH ; 6  
DB 3FH,33H,30H,18H,0CH,0CH,0CH ; 7  
DB 1EH,33H,33H,1EH,33H,33H,1EH ; 8  
DB 1EH,33H,33H,3EH,30H,33H,1EH ; 9  
DB 00H,0CH,0CH,00H,0CH,0CH,00H ; :  
DB 00H,0CH,0CH,00H,0CH,08H,04H ; ;  
DB 18H,0CH,06H,03H,06H,0CH,18H ; <  
DB 00H,00H,3FH,00H,3FH,00H,00H ; =  
DB 06H,0CH,18H,30H,18H,0CH,06H ; >  
DB 1EH,33H,30H,18H,0CH,00H,0CH ; ?

DB 00H,00H,00H,3CH,00H,03H,03H ; .-  
DB 0CH,12H,33H,33H,3FH,33H,33H ; A  
DB 1FH,33H,33H,1FH,33H,33H,1FH ; B  
DB 1EH,33H,03H,03H,03H,33H,1EH ; C  
DB 0FH,1BH,33H,33H,33H,1BH,0FH ; D  
DB 3FH,03H,03H,1FH,03H,03H,3FH ; E  
DB 3FH,03H,03H,1FH,03H,03H,03H ; F  
DB 1EH,33H,03H,3BH,33H,33H,1EH ; G  
DB 33H,33H,33H,3FH,33H,33H,33H ; H  
DB 1EH,0CH,0CH,0CH,0CH,0CH,1EH ; I  
DB 30H,30H,30H,30H,33H,33H,1EH ; J  
DB 23H,13H,0BH,1FH,33H,33H,33H ; K  
DB 03H,03H,03H,03H,03H,33H,3FH ; L  
DB 41H,63H,77H,7FH,6BH,63H,63H ; M  
DB 31H,33H,37H,3FH,3BH,33H,23H ; N  
DB 1EH,33H,33H,33H,33H,33H,1EH ; O

DB 1FH,33H,33H,1FH,03H,03H,03H ; P  
DB 1EH,33H,33H,33H,3BH,1EH,20H ; Q  
DB 1FH,33H,33H,1FH,0FH,1BH,33H ; R  
DB 1EH,33H,03H,1EH,30H,33H,1EH ; S  
DB 3FH,0CH,0CH,0CH,0CH,0CH,0CH ; T  
DB 33H,33H,33H,33H,33H,33H,1EH ; U  
DB 63H,63H,63H,63H,36H,1CH,08H ; V  
DB 63H,63H,6BH,7FH,77H,63H,41H ; W