ORG 0000H

LJMP MAIN

ORG 0003H

CLR TR0

LJMP MODESELECT

ORG 000BH

CLR TR0

LJMP UPDATETIME

ORG 001BH

CLR TR1

LJMP BUZZERTIME

ORG 30H

MAIN:

MOV SP, #50H

DIGIT EQU P0

TRAN1 EQU P2.7

TRAN2 EQU P2.6

TRAN3 EQU P2.5

TRAN4 EQU P2.4

TRAN5 EQU P2.3

TRAN6 EQU P2.2

LEFT EQU P1.0

RIGHT EQU P1.1

UP EQU P1.2

DOWN EQU P1.3

BUZZER EQU P1.4

MODE EQU P3.2

ALARM EQU P2.0

TIMESET EQU P2.1

MOV 45H, #19

MOV 46H, #15

MOV 47H, #19

MOV 48H, #15

MOV 49H, #13

MOV 4AH, #12

CLR BUZZER

MOV TMOD, #11H

MOV IE, #8BH

MOV DPTR, #BITPATTERN

MOV R0, #2

MOV R1, #5

MOV R2, #19

MOV R3, #5

MOV R4, #13

MOV R5, #2

MOV TH0, #0EFH

MOV TL0, #0DBH

SETB TR0

MOV TH1, #0DCH

MOV TL1, #00H

ENDLESS:

MOV A, R0

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN1

CLR TRAN1

MOV A, R1

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN2

CLR TRAN2

MOV A, R2

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN3

CLR TRAN3

MOV A, R3

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN4

CLR TRAN4

MOV A, R4

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN5

CLR TRAN5

MOV A, R5

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN6

CLR TRAN6

SETB PSW.3

MOV R6, #6

MOV R0, #0H

MOV R1, #31H

CONTINUE: MOV A, @R0

MOV B, @R1

CJNE A, B, EXITCHECK

INC R0

INC R1

DJNZ R5, CONTINUE

SETB TR1

EXITCHECK:

CLR PSW.3

LJMP ENDLESS

ORG 0C0H

UPDATETIME:

MOV TH0, #0EFH

MOV TL0, #0DBH

SETB TR0

MOV 1FH, PSW

CLR PSW.3

CLR PSW.4

INC R6

CJNE R6, #223, ENDISR1

MOV R6, #0

INC R0

CJNE R0, #10, ENDISR1

MOV R0, #0

INC R1

CJNE R1, #6, ENDISR1

MOV R1, #0

INC R2

CJNE R2, #20, ENDISR1

MOV R2, #10

INC R3

CJNE R3, #6, ENDISR1

MOV R3, #0

CJNE R5, #2, NH24

INC R4

CJNE R4, #14, ENDISR1

MOV R4, #10

MOV R5, #0

SJMP ENDISR1

NH24: INC R4

CJNE R4, #20, ENDISR1

MOV R4, #10

INC R5

ENDISR1: MOV R7, 1FH

ENDISR:

MOV A, R7

ANL A, #00011000B

CJNE A, #0H, NOT0

SJMP ENDGAME

NOT0: CJNE A, #08H, NPSW3

SETB PSW.3

SJMP ENDGAME

NPSW3: SETB PSW.4

ENDGAME: RETI

ORG 120H

BUZZERTIME:

MOV TH1, #0DCH

MOV TL1, #00H

SETB TR1

MOV 0FH, PSW

CLR PSW.3

CLR PSW.4

SETB PSW.3

JNB ALARM, ALARMSTOP

INC R2

CJNE R2, #50, ENDISRT1

MOV R2, #0

CPL BUZZER

INC R3

CJNE R3, #10, ENDISRT1

SJMP BUZSTOP

ALARMSTOP:

MOV R4, #255

BACK2: MOV R5, #255

BACK1: DJNZ R5, BACK1

DJNZ R6, BACK2

BUZSTOP: MOV R2, #0

MOV R3, #0

CLR BUZZER

CLR TR1

MOV TH1, #0DCH

MOV TL1, #00H

ENDISRT1:

CLR PSW.3

MOV R7, 0FH

LJMP ENDISR

ORG 150H

MODESELECT:

MOV 17H, PSW

CLR PSW.3

CLR PSW.4

SETB PSW.4

LCALL DEBOUNCE

MOV P2, #0

SETB P2.1

SETB P2.0

AGAIN: JB ALARM, NEXT4

MOV IP, #02H

LCALL DEBOUNCE

MOV R6, #6

MOV R0, #0H

MOV R1, #31H

TRANSFER: MOV A, @R0

MOV @R1, A

INC R0

INC R1

DJNZ R6, TRANSFER

MOV R0, #31H

MOV R1, #31H

LJMP SETMODE

NEXT4: JB TIMESET, AGAIN

LCALL DEBOUNCE

MOV R0, #0H

MOV R1, #0H

LJMP SETMODE

SETMODE:

MOV R5, #-10

MOV R4, #2

LCALL RADDX

LCALL ADDX

MOV R4, #2

LCALL RADDX

LCALL ADDX

MOV R5, #10

MOV R4, #-4

LCALL RADDX //This function performs R0+=R4

LCALL ADDX //This function performs @R0+=R5

CHECK: JB LEFT, NEXT0

LCALL DEBOUNCE

MOV A, R1

ADD A, #5

MOV B, A

MOV A, R0

CJNE A, B, NORM0

MOV R5, #-10

LCALL ADDX

MOV R4, #-5

LCALL RADDX

MOV R5, #10

LCALL ADDX

SJMP NEXT0

NORM0: MOV R5, #-10

LCALL ADDX

INC R0

MOV R5, #10

LCALL ADDX

NEXT0: JB RIGHT, NEXT1

LCALL DEBOUNCE

MOV B, R1

MOV A, R0

CJNE A, B, NORM1

MOV R5, #-10

LCALL ADDX

MOV R4, #5

LCALL RADDX

MOV R5, #10

LCALL ADDX

SJMP NEXT1

NORM1: MOV R5, #-10

LCALL ADDX

DEC R0

MOV R5, #10

LCALL ADDX

NEXT1: JB UP, NEXT2

LCALL DEBOUNCE

CJNE @R0, #10, NORM2

MOV B, R1

PUSH B

MOV A, R0

MOV R6, #6

MOV R1, #45H

SOMENAME: CJNE A, B, NDIGIT

MOV A, R2

CJNE A, #4, NCHECK24

INC R0

CJNE @R0, #2, NOWORRY

DEC R0

NCHECK24: MOV A, @R1

MOV @R0, A

SJMP UPEXIT

NDIGIT: INC B

INC R1

INC R2

DJNZ R6, SOMENAME

NOWORRY: DEC R0

MOV @R0, #19

UPEXIT: POP B

MOV R1, B

SJMP NEXT2

NORM2: MOV R5, #-1

LCALL ADDX

NEXT2: JB DOWN, NEXT3

LCALL DEBOUNCE

MOV B, R1

PUSH B

MOV A, R0

MOV R6, #6

MOV R1, #45H

SOMENAME2: CJNE A, B, NDIGIT2

MOV A, R2

CJNE A, #4, N2CHECK24

INC R0

CJNE @R0, #2, NOWORRY2

DEC R0

N2CHECK24: MOV A, @R0

MOV B, @R1

CJNE A, B, NORM3

MOV @R0, #10

SJMP DOWNEXIT

NDIGIT2: INC B

INC R1

INC R2

DJNZ R6, SOMENAME2

NOWORRY2: DEC R0

CJNE @R0, #19, NORM3

MOV @R0, #10

DOWNEXIT: POP B

MOV R1, B

SJMP NEXT3

NORM3: POP B

MOV R1, B

MOV R5, #1

LCALL ADDX

NEXT3: JB MODE, STOPC

LCALL DEBOUNCE

LJMP TERMINATE

STOPC: MOV A, @R1

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN1

CLR TRAN1

INC R1

MOV A, @R1

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN2

CLR TRAN2

INC R1

MOV A, @R1

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN3

CLR TRAN3

INC R1

MOV A, @R1

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN4

CLR TRAN4

INC R1

MOV A, @R1

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN5

CLR TRAN5

INC R1

MOV A, @R1

MOVC A, @A+DPTR

MOV DIGIT, A

SETB TRAN6

CLR TRAN6

MOV A, R1

ADD A, #-5

MOV R1, A

LJMP CHECK

TERMINATE:

MOV R5, #-10

LCALL ADDX

CJNE R0, #30H, COMPARE

COMPARE: JC LESSER

MOV R0, #31H

SJMP ENDIT

LESSER: MOV R0, #0H

ENDIT:

MOV R5, #10

MOV R4, #2

LCALL RADDX

LCALL ADDX

MOV R4, #2

LCALL RADDX

LCALL ADDX

CLR PSW.4

MOV IP, #00H

MOV R7, 17H

MOV TH0, #0EFH

MOV TL0, #0DBH

SETB TR0

LJMP ENDISR

RADDX:

MOV A, R0

ADD A, R4

MOV R0, A

RET

ADDX:

MOV A, @R0

ADD A, R5

MOV @R0, A

RET

DEBOUNCE:

MOV R2, #255

LOOP5: MOV R3, #255

LOOP4: DJNZ R3, LOOP4

DJNZ R2, LOOP5

RET

ORG 300H

BITPATTERN:

DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH,0BFH,86H,0DBH,0CFH,0E6H,0EDH,0FDH,87H,0FFH,0EFH

END