**16 bit mul:**

```
org 00h
mov r1, #0EDh
mov r2, #0FFh
mov r3, #0FDh
mov r4, #0FFh
mov a,r3
mov b,r4
mul ab
mov 20h,a
mov 21h,b
mov a,r3
mov b, r2
mul ab
mov 22h,b
addc a,21h
mov 21h,a
mov a,r1
mov b,r4
mul ab
addc a,21h
mov 21h,a
mov a,b
addc a,22h
mov 22h,a
mov a,r1
mov b,r2
mul ab
addc a,22h
mov 22h,a
mov a,b
addc a,#00h
mov 23h,a
end
```

**ASCII to packed BCD:**

```
org 00h
  mov a,#33h   ;
  anl a,#0fh ;
  swap a ;
  mov b,a;
  mov a,#32h   ;
  anl a,#0fh ;
  orl a,b ;
  mov r1,a
 end
```

**packed BCD to unpacked BCD**

```
org 00h
  mov a, #79h
  mov r0,a
  anl a,#0fh ;
  mov r1,a ;
  mov a,r0
  swap a ;
  anl a,#0fh ;
  mov r2,a;
  end
```

**ALP for Reversal of a given string without null character**

```
org 00h
       mov dptr
       mov r1,#0eh
            mov r0 ,#4eh
back: clr a
     movc a, @a+dptr
     mov @r0,a
     dec r0
     inc dptr
     djnz r1,back
here: nop
     sjmp here

org 300h
mydata: db "ECE department"
    end
```

**Reversal of string with null character**

```
org 00h
       mov dptr,#mydata
            mov r0 ,#4eh
back: clr a
     movc a, @a+dptr
     jz here
     mov @r0,a
     dec r0
     inc dptr
     sjmp back
here: sjmp here
org 300h
mydata: db "ECE department",0
      end
```

**finding a character**

```
org 00h
       mov dptr,#mydata
            mov r0 ,#4eh
back: clr a
     movc a, @a+dptr
     mov @r0,a
     acall count
     jz here
     dec r0
     inc dptr
              sjmp back
here: sjmp here
org 300h
mydata: db "ECE department",0
org 400h
count: cjne  a,#"E", cnt
      inc r1
cnt:   ret
end
```

**sorting of numbers in ascending order**

```
org 00h
       mov r4,#05h
again:   mov r3, #04h
       mov r0, #30h
       clr c
ascnd:   mov a, @r0
       mov r1,a
       inc r0
       mov a,@r0
       subb a,r1   // compare two consecutive numbers
       jnc skip
       mov a,@r0
       dec r0
       mov @r0,a
       mov a,r1
       inc r0
       mov @r0,a
skip:    djnz r3, ascnd
       djnz r4, again
       end
```

**ALP for checksum calculation**

```
org 00h
       mov dptr,#mydata
       mov r1,#10h
back: clr a
     movc a, @a+dptr
     add a,b
              mov b,a
     inc dptr
     djnz r1,back
     mov a,b
     cpl a
     add a,#1
     mov r0,a
org 100h
mydata: db "Test Data Chksum"
    end
```

**stepper motor**

```
org 00
     mov p0,#00h
start:mov a,#66h
     mov p0,a  // drive p0
clkw:  mov r0,#10
clkw1: mov r1,#254
clkw2: rr a
      acall delay
      mov p0,a
      djnz r1,clkw2
      djnz r0,clkw1
aclkw:  mov r0,#10
aclkw1: mov r1,#254
aclkw2: rl a
       acall delay
       mov p0,a
       djnz r1,aclkw2
            djnz r0,aclkw1
            sjmp start
delay: mov r2,#100
here1: mov r3,#100
here2:djnz r3,here2
     djnz r2,here1
     ret
    end
```

**keyboard**

```
ORG 00h
      mov p2,#0
      mov p0,#0ffh
      mov p1,#0
k1:   mov p1,#0
      mov a,p0
      anl a,0fh
      cjne a,#0fh,k1
k2:   acall delay
      mov a,p0
      anl a,0fh
      cjne a,#0fh,over
      sjmp k2
over:  acall delay
      mov a,p0
      anl a,0fh
      cjne a,#0fh,over1
      sjmp k2
over1: mov p1, #0feh
      mov a,p0
      anl a,#0fh
      cjne a,#0fh, row_0
      mov p1,#11111101b
      mov a,p0
      anl a,#0fh
      cjne a,#0fh,row_1
     mov p1,#11111011b
     mov a,p0
     anl a,#0fh
     cjne a,#0fh,row_2
     mov p1,#11110111b
     mov a,p0
     anl a,#0fh
     cjne a,#0fh,row_3
     ljmp k2
row_0:mov dptr,#kcode0
     sjmp find
row_1:mov dptr,#kcode1
     sjmp find
row_2:mov dptr,#kcode2
     sjmp find
row_3:mov dptr,#kcode3
find: rrc a
     jnc match
     inc dptr
     sjmp find
match:clr a
     movc a,@a+dptr
     mov p2,a
     ljmp k1
delay: mov r2,#10
here1: mov r3,#100
here2:djnz r3,here2
     djnz r2,here1
     ret
org 300h
kcode0: db   '0','1','2','3'        ;row 0 ~ 3
kcode1: db   '4','5','6','7'
kcode2: db   '8','9','A','B'
kcode3: db   'C','D','E','F'
       end
```

**seven segment display**

```
org 00h
      mov p0,00h
start: mov dptr,#mydata
             mov r3,#21
             mov p0,#00
loop1:   clr a
           movc a,@a+dptr
           acall delay
           inc dptr
           mov p0,a
           djnz r3,loop1
           sjmp start
delay:  mov r2,#04
here:    mov r0,#250
here1:   mov r1,#250
here2:   nop
           nop
           djnz r1,here2
           djnz r0,here1
           djnz r2,here
           ret
org 100h
mydata: db 0c0h,0f9h,0a4h,0b0h, 099h, 092h, 082h, 0f8h, 080h, 090h,  04h, 03h, 46h, 21h, 06h,
0Dh, 09h, 47h, 71h, 47h, 2bh
  end
```

**pwm (25%)**

```
org 00h
      mov p1,#00h
      mov tmod,#10h
again: setb p1.7
      acall delay
      cpl p1.7
      acall delay1
      sjmp again
delay: mov th1,#0ffh
      mov tl1,#06h
      clr tf1
      setb tr1
loop: jnb tf1,loop
      clr tr1
      ret
delay1: mov th1,#0fdh
       mov tl1,#12h
       clr tf1
       setb tr1
here: jnb tf1,here
      clr tr1
      ret
      end
```

**pwm(75%)**

```
org 00h
      mov p1,#00h
      mov tmod,#10h
again: setb p1.7
      acall delay
      cpl p1.7
      acall delay1
      sjmp again
delay: mov th1,#0feh
      mov tl1,#0d4h
      clr tf1
      setb tr1
loop: jnb tf1,loop
      clr tr1
      ret
delay1: mov th1,#0ffh
       mov tl1,#9ch
       clr tf1
       setb tr1
here: jnb tf1,here
      clr tr1
      ret
      end
```

**LCD**

```
org    00h
       sjmp    30h
org    30h
              mov     P3,#1fh
       mov     P2,#00h
              acall   init
       mov     dptr,#msg
disp:  clr     a
       movc    a,@a+dptr
       jz      here
              acall   dat
       inc     dptr
              mov         a,dpl
              cjne        a,#10h,disp
              mov         a,#0c0h
              acall       cmd
              sjmp        disp
here:         sjmp    here
init:
              mov     a,#38h
       acall   cmd
       mov     a,#01h
       acall   cmd
              mov     a,#0ch
       acall   cmd
       mov     a,#80h
       acall   cmd
       ret
cmd:   nop
       clr     RS
       nop
      clr     RW
       nop
       mov     P2,a
       nop
       setb    E
       nop
       clr     E
       mov  r2,#03h
              acall       delay
       ret
dat:   nop
       setb    RS
       nop
       clr     RW
       nop
       mov     P2,a
       nop
       setb    E
       nop
       clr     E
       mov                 r2,#03h
              acall       delay
       ret
delay:  mov     r1,#0ffh
 here1: djnz    r1, here1
       djnz  r2,delay
                           ret
org                        300h
msg:    db     'ECE department',0
       end
```

**sine wave**

```
org 0000h
      mov p0,#00h
start:    mov dptr,#table
                  mov r2,#255
next:     clr a
                  movc a,@a+dptr
                  mov p0,a
                  inc dptr
                  djnz r2,next
                  sjmp start
table:
db 080h,083h,086h,089h,08ch,08fh,092h,095h
db 098h,09ch,09fh,0a2h,0a5h,0a8h,0abh,0aeh
db 0b0h,0b3h,0b6h,0b9h,0bch,0bfh,0c1h,0c4h
db 0c7h,0c9h,0cch,0ceh,0d1h,0d3h,0d5h,0d8h
db 0dah,0dch,0deh,0e0h,0e2h,0e4h,0e6h,0e8h
db 0eah,0ech,0edh,0efh,0f0h,0f2h,0f3h,0f5h
db 0f6h,0f7h,0f8h,0f9h,0fah,0fbh,0fch,0fch
db 0fdh,0feh,0feh,0ffh,0ffh,0ffh,0ffh,0ffh
db 0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0feh,0feh
db 0fdh,0fch,0fch,0fbh,0fah,0f9h,0f8h,0f7h
db 0f6h,0f5h,0f3h,0f2h,0f0h,0efh,0edh,0ech
db 0eah,0e8h,0e6h,0e4h,0e2h,0e0h,0deh,0dch
db 0dah,0d8h,0d5h,0d3h,0d1h,0ceh,0cch,0c9h
db 0c7h,0c4h,0c1h,0bfh,0bch,0b9h,0b6h,0b3h
db 0b0h,0aeh,0abh,0a8h,0a5h,0a2h,09fh,09ch
db 098h,095h,092h,08fh,08ch,089h,086h,083h
db 07fh,07ch,079h,076h,073h,070h,06dh,06ah
db 067h,063h,060h,05dh,05ah,057h,054h,051h
db 04fh,04ch,049h,046h,043h,040h,03eh,03bh
db 038h,036h,033h,031h,02eh,02ch,02ah,027h
db 025h,023h,021h,01fh,01dh,01bh,019h,017h
db 015h,013h,012h,010h,00fh,00dh,00ch,00ah
db 009h,008h,007h,006h,005h,004h,003h,003h
db 002h,001h,001h,000h,000h,000h,000h,000h
db 000h,000h,000h,000h,000h,000h,001h,001h
db 002h,003h,003h,004h,005h,006h,007h,008h
db 009h,00ah,00ch,00dh,00fh,010h,012h,013h
db 015h,017h,019h,01bh,01dh,01fh,021h,023h
db 025h,027h,02ah,02ch,02eh,031h,033h,036h
db 038h,03bh,03eh,040h,043h,046h,049h,04ch
db 04fh,051h,054h,057h,05ah,05dh,060h,063h
db 067h,06ah,06dh,070h,073h,076h,079h,07ch

                  end
```

**serial communication**

```
org 00h
      mov p0,#00h // port 0 as output
  // serial comm initiliazation
      mov p3, #0fdh  // p3.1 rs 232 transmit pin is output
      mov TMOD, #20h // timer1, mode 2
      mov TH1, 0fdh  // 9600 baud rate
      mov scon, #50h // 8 bit, 1 stop, ren enable
      setb TR1  // start timer
      clr TI
ser:     mov a,"K"
ser2:    mov sbuf,a
here:    jnb TI, here
         clr TI
      ret

    end
```