Finding Minimum Element

```
cr      equ     0dh
lf      equ     0ah
data    segment
table   db      '0123456789abcdef'
array   db      05h,09h,03h,01h
result  db      00h
msg     db      'the smallest number is '
asciir  db      2 dup(?)
        db      cr,lf,'$'
data    ends
code    segment
        assume cs:code,ds:data
start:  mov     ax,data
        mov     ds,ax
        mov     cx,00004h
        dec     cx
        lea     si,array
        mov     al, [si]
top:    inc     si
        cmp     al,[si]
        jnc     down
        loop    top
        jmp     last
down:   xchg    al,[si]
        loop    top
last:   mov     result,al
        lea     bx,table
        lea     di,asciir
        inc     di
        mov     al,result
        and     al,00fh
        xlat
        mov     [di],al
        dec     di
        mov     al,result
        and     al,0f0h
        mov     cl,04h
        shr     al,cl
        xlat
        mov     [di],al
        mov     ah,09h
        lea     dx,msg
        int     21h
quit:   mov     al,00h
        mov     ah,04ch
        int     21h
code    ends
        end     start
```

```
C:\MASM611\BIN>smallest
the smallest number is 01
```

Finding Maximum Element

```
cr      equ     0dh
lf      equ     0ah
data    segment
table   db      '0123456789abcdef'
array   db      40h,39h,81h,11h
result  db      00h
msg     db      'the largest number is '
asciir  db      2 dup(?)
        db      cr,lf,'$'
data    ends
code    segment
        assume cs:code,ds:data
start:  mov     ax,data
        mov     ds,ax
        mov     cx,004h
        dec     cx
        lea     si,array
        mov     al,[si]
top:    inc     si
        cmp     al,[si]
        jc      down
        loop    top
        jmp     last
down:   xchg    al,[si]
        loop    top
last:   mov     result,al
        lea     bx,table
        lea     di,asciir
        inc     di
        mov     al,result
        and     al,00fh
        xlat
        mov     [di],al
        dec     di
        mov     al,result
        and     al,0f0h
        mov     cl,04h
        shr     al,cl
        xlat
        mov     [di],al
        mov     ah,09h
        lea     dx,msg
        int     21h
quit:   mov     al,00h
```

```
        mov     ah,04ch
        int     21h
code    ends
        end     start
```

```
C:\MASM611\BIN>largest
the largest number is 81
```

Sorting – Ascending Order

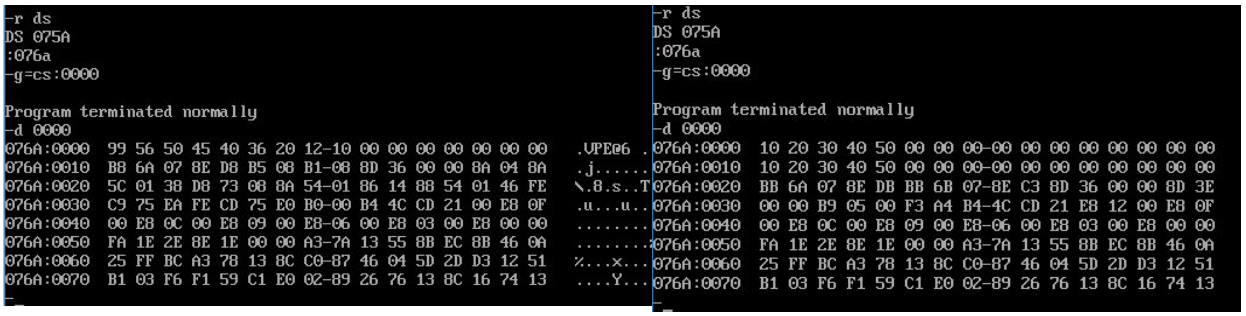
```
data segment
string1 db
40h,10h,50h,20h,99h,12h,56h,45h,36h
data ends
code segment
        assume cs:code,ds:data
start:  mov ax,data
        mov ds,ax
        mov ch,08h
up2:    mov cl,08h
        lea si,string1
up1:    mov al,[si]
        mov bl,[si+1]
        cmp al,bl
        jc down
        mov dl,[si+1]
        xchg [si],dl
        mov [si+1],dl
down:   inc si
        dec cl
        jnz up1
        dec ch
        jnz up2
quit:   mov al,0
        mov ah,04ch
        int 21h
code    ends
        end start
```

```
-r ds
DS 075A
:076a
-g=cs:0000

Program terminated normally
-d 0000
076A:0000  10 12 20 36 40 45 50 56-99 00 00 00 00 00 00 00   .. 6@EPV........
076A:0010  B8 6A 07 8E D8 B5 08 B1-08 8D 36 00 00 8A 04 8A   .j........6.....
076A:0020  5C 01 38 D8 72 08 8A 54-01 86 14 88 54 01 46 FE   \.8.r..T....T.F.
076A:0030  C9 75 EA FE CD 75 E0 B0-00 B4 4C CD 21 00 E8 0F   .u...u....L.!...
076A:0040  00 E8 0C 00 E8 09 00 E8-06 00 E8 03 00 E8 00 00   ................
076A:0050  FA 1E 2E 8E 1E 00 00 A3-7A 13 55 8B EC 8B 46 0A   ........z.U...F.
076A:0060  25 FF BC A3 78 13 8C C0-87 46 04 5D 2D D3 12 51   %...x....F.]-..Q
076A:0070  B1 03 F6 F1 59 C1 E0 02-89 26 76 13 8C 16 74 13   ....Y....&v...t.
-
```

Sorting – Descending Order

```
data segment
string1 db
40h,10h,50h,20h,99h,12h,56h,45h,36h
data ends
code segment
        assume cs:code,ds:data
start:  mov ax,data
        mov ds,ax
        mov ch,08h
up2:    mov cl,08h
        lea si,string1
up1:    mov al,[si]
        mov bl,[si+1]
        cmp al,bl
        jnc down
        mov dl,[si+1]
        xchg [si],dl
        mov [si+1],dl
down:   inc si
        dec cl
        jnz up1
        dec ch
        jnz up2
quit:   mov al,0
        mov ah,04ch
        int 21h
code    ends
        end start
```



Block Transfer

```
ASSUME DS:DATA, CS:CODE, ES:EXTRA
DATA SEGMENT
BLK DB 10H, 20H,30H,40H, 50H
DATA ENDS
EXTRA SEGMENT
RESULT DB 5 DUP(?)
EXTRA ENDS
CODE SEGMENT
 START: MOV BX,DATA
  MOV DS,BX
  MOV BX,EXTRA
  MOV ES,BX
  LEA SI,BLK
  LEA DI,RESULT
   MOV CX,0005H
  REP MOVSB
   MOV AH,4CH
  INT 21H
CODE ENDS
END START
```

Number Conversion
Hexadecimal – Decimal

```
CR EQU 0DH
LF EQU 0AH
DATA SEGMENT
TABLE DB '0123456789'

MESSAGE DB '  THE DECIMAL NUMBER
IS:'
DECIMAL DB 4  dup(' ')
DB CR, LF, '$'
DATA ENDS

CODE SEGMENT
Assume CS:Code, DS: Data
START: MOV AX, DATA
MOV DS, AX
MOV AX,000FFH
MOV CX, 0000AH
LEA BX, TABLE
LEA SI, DECIMAL
ADD SI,3
TOP: CMP AX, CX
JB BOTTOM
XOR DX,DX

DIV CX
PUSH AX
MOV AL,DL
XLAT
MOV [SI], AL
DEC SI
POP AX
JMP TOP
BOTTOM: XLAT
MOV [SI], AL
MOV AH,09H
LEA DX,MESSAGE
INT 21H
QUIT: MOV AL,0
```

```
MOV AH,04CH
INT 21H
CODE ENDS
END START
```

```
C:\MASM611\BIN>hextodec
  THE DECIMAL NUMBER IS: 255
```

Number Conversion
Decimal – Hexadecimal

```
CR EQU 0DH
LF EQU 0AH
DATA SEGMENT
TABLE DB '0123456789ABCDEF'
DECILENGTH DB 4
DECIMAL DB '0010'
HEXADECIMAL DW 0000
MULTIFACTOR          DW 0001
MESSAGE DB '  THE HEXADECIMAL
NUMBER IS:'
HEXASCII DB 4  dup(' ')
 DB CR, LF, '$'
DATA ENDS
CODE SEGMENT
 Assume CS:Code, DS: Data
START: MOV AX,DATA
 MOV DS,AX
 MOV CX,0004
 LEA SI,DECIMAL
 ADD SI,3
TOP: MOV AL, [SI]
 AND AX, 0000FH
 MUL MULTIFACTOR
 ADD HEXADECIMAL, AX
 MOV AX,MULTIFACTOR
 MOV BX,0000AH
 MUL BX
 MOV MULTIFACTOR,AX
 DEC SI
 LOOP TOP
 LEA SI,HEXASCII
 ADD SI,3
 LEA BX,TABLE
 MOV AX,HEXADECIMAL
 AND AX,0000FH
 XLAT
 MOV [SI],AL
 DEC SI
 MOV AX,HEXADECIMAL

 AND AX,000F0H
 MOV CL,04H
 SHR AL,CL
 XLAT
 MOV [SI],AL
 DEC SI
 MOV AX,HEXADECIMAL

 AND AX,00F00H
 MOV CL,08H
 SHR AX,CL
 XLAT
 MOV [SI],AL
 DEC SI
 MOV AX,HEXADECIMAL
 AND AX,0F000H
 MOV CL,00CH
 SHR AX,CL
 XLAT
 MOV [SI],AL
 DEC SI
 MOV AH,09H
 LEA DX,MESSAGE
 INT 21H
QUIT: MOV AL,0
 MOV AH,04CH
 INT 21H
 CODE ENDS
 END START
```

```
C:\MASM611\BIN>dectohex
  THE HEXADECIMAL NUMBER IS:000A
```

String Manipulations - String Reversal

```
CR EQU 0DH
LF EQU 0AH
DATA SEGMENT
MESSAGE1 DB 'THE GIVEN WORD IS'
DB CR, LF, '$'
STRING1 DB 'HELLO'
                DB      CR,LF,'$'
STRLENGTH EQU $-STRING1

MESSAGE2 DB 'THE REVERSED STRING
IS'
DB CR, LF, '$'
```

```
STRING2 DB 5 DUP(' ')
DB CR, LF, '$'
DATA ENDS

CODE SEGMENT
Assume CS:Code, DS: Data,ES:Data
START: MOV AX, DATA
MOV DS, AX
MOV ES,AX
LEA SI,STRING1

LEA DI, STRING2+4
MOV CX, 0005h
TOP: CLD
LODSB
        STD
        STOSB
LOOP TOP
MOV AH,09H
LEA DX, MESSAGE1
        INT 21H
        MOV AH,09H
LEA DX, STRING1
        INT 21H
        MOV AH,09H
LEA DX, MESSAGE2
        INT 21H
        MOV AH,09H
LEA DX, STRING2
        INT 21H
QUIT: MOV AL,0
MOV AH, 04CH
INT 21H
        CODE ENDS
        END START
```

```
C:\MASM611\BIN>rev
THE GIVEN WORD IS
HELLO
THE REVERSED STRING IS
OLLEH
```

String Palindrome

```
CR EQU 0DH
LF EQU 0AH
DATA SEGMENT
STRING1 DB 'LIRIL'
STRLENGTH EQU $-STRING1
MESSAGE1 DB 'THE GIVEN WORD IS
PALINDROME'
DB CR, LF, '$'
MESSAGE2 DB 'THE GIVEN WORD IS
NOT PALINDROME'
DB CR, LF, '$'
STRING2 DB 5 DUP(' ')
DB CR, LF, '$'
DATA ENDS
CODE SEGMENT
Assume CS:Code, DS: Data
START: MOV AX, DATA
MOV DS, AX
MOV ES,AX
LEA SI,STRING1
LEA DI, STRING2+STRLENGTH-1
MOV CX, STRLENGTH
TOP: CLD
LODSB
STD
STOSB
LOOP TOP
LEA SI, STRING1
LEA DI, STRING2
CLD
MOV CX, STRLENGTH
REPE CMPSB
CMP CX,00H
JNZ NOTPALINDROME
MOV AH,09H
LEA DX, MESSAGE1
INT 21H
JMP QUIT
NOTPALINDROME: MOV AH,09H
LEA DX, MESSAGE2
INT 21H
QUIT: MOV AL,0
MOV AH, 04CH
INT 21H
CODE ENDS
END START
```

```
C:\MASM611\BIN>pal
THE GIVEN WORD IS PALINDROME
```

Uppercase to Lowercase Conversion

```
cr equ 0dh
lf equ 0ah
data segment
message2 db cr,lf,' the translatedstring:','$'
table db 48 dup(' '), '0123456789', 7 dup(' ')
```

```
db 'abcdefghijklmnopqrstuvwxyz', 6 dup(' ')
db 'abcdefghijklmnopqrstuvwxyz', 133 dup(' ')

namepar         label   byte
maxlen db       0ah
actlen  db      ?
string  db      0ffh dup('$')
db      cr,lf.'$'
message1 db cr,lf,'enter string:$'
data ends
code segment
assume cs:code, ds:data
start: mov ax,data
mov ds,ax
mov es,ax
mov     ah,09h
lea dx,message1
int 21h
mov ah,0ah
lea dx, namepar
int 21h
lea si,string
lea di,string
lea bx,table
mov     ch,000h
mov     cl,actlen
cld
top: lodsb
xlat
stosb
loop top
mov     ah,09h
lea dx, message2
int 21h
mov ah,09h
lea dx, string
int 21h
quit: mov al,0
mov ah,04ch
int 21h
code ends
end start
```

```
C:\MASM611\BIN>uptolow

ENTER STRING:INDIA
 THE TRANSLATEDSTRING:india
```

Lowercase to Uppercase Conversion

```
cr equ 0dh
lf equ 0ah

data segment
message2 db cr,lf,' the translatedstring:','$'
table db 48 dup(' '), '0123456789', 7 dup(' ')

db 'abcdefghijklmnopqrstuvwxyz', 6 dup(' ')
db 'abcdefghijklmnopqrstuvwxyz', 133 dup(' ')
namepar         label   byte
maxlen db       0ah
actlen  db      ?
string  db      0ffh dup('$')
db      cr,lf.'$'
message1 db cr,lf,'enter string:$'
data ends
code segment

assume cs:code, ds:data

start: mov ax,data
mov ds,ax
mov es,ax
mov     ah,09h
lea dx,message1
int 21h
mov ah,0ah
lea dx, namepar
int 21h
lea si,string
lea di,string
lea bx,table
mov     ch,000h
mov     cl,actlen
cld
top: lodsb
xlat
stosb

loop top
mov     ah,09h
lea dx, message2
int 21h
mov ah,09h
lea dx, string
int 21h
quit: mov al,0
mov ah,04ch
int 21h
code ends
end start
```


C:\MASM611\BIN>lowtoup
ENTER STRING:india
THE TRANSLATEDSTRING:INDIA