%macro scall 4

mov eax,%1

mov ebx,%2

mov ecx,%3

mov edx,%4

int 80h

%endmacro

section .data

m1 db "Enter size of array: "

l1 equ $-m1

m2 db "Enter array elements: "

l2 equ $-m2

m3 db 10d,13d,"Largest: "

l3 equ $-m3

m4 db 10d,13d

l4 equ $-m4

section .bss

cnt resb 3

arr resb 3

cnt1 resb 3

arr1 resb 50

temp resb 2

char\_ans resb 2

section .text

global \_start

\_start:

scall 4,1,m1,l1 ;macro call to display m1

scall 3,0,arr,3 ;macro call to input in arr

mov esi,arr ;esi points to arr

call asciihextohex

;calling procedure to convert into hex nos

mov byte[cnt],dl ;moving values of dl into cnt

mov byte[cnt1],dl ;moving values of dl into cnt

scall 4,1,m2,l2 ;macro call to display m2

mov edi,arr1 ;edi points to arr1

back: scall 3,0,arr,3 ;macro call to input in arr

mov esi,arr ;esi points to arr

call asciihextohex ;calling procedure to convert into hex nos

mov [edi],dl ;move contents of dl at address of edi

inc edi ;increment edi to point to next element

dec byte[cnt] ;decrement cnt variable

jnz back ;jump if cnt is not zero to back label

mov esi,arr1 ;esi points to arr1

mov al,[esi] ;move contents at esi into al

inc esi ;increment esi to point to next element

up1: mov bl,[esi] ;move contents at esi into bl

cmp al,bl ;compare al with bl

jg next1 ;if al is greater, jump to next1

mov byte[temp],al ;copying al into temp

mov al,bl ;copying bl into al

mov bl,byte[temp] ;copying temp into bl

next1: inc esi ;increment esi

dec byte[cnt1] ;decrement cnt1

jnz up1 ;jump to up1, if cnt1 not zero

mov ecx,02 ;copy 02 into ecx

mov esi,char\_ans ;esi points to char\_ans

HtoA: rol al,4 ;roll contents of al by 4 bits

mov dl,al ;copy al into dl

and dl,0FH ;AND dl with 0Fh

cmp dl,09h ;compare dl with 09h

jbe next2 ;jump to next2, if dl is below or equal

add dl,07h ;add 07h to dl

next2: add dl,30h ;add 30h to dl

mov[esi],dl ;move dl at esi address

inc esi ;increment esi

dec ecx ;decrement ecx

jnz HtoA ;jump to up4, if ecx not zero

scall 4,1,m3,l3 ;macro call to display m3

scall 4,1,char\_ans,2 ;macro call to display char\_ans

scall 4,1,m4,l4 ;macro call to display m4

mov eax,1 ;sys\_Exit

mov ebx,0 ;sucessfull termination

int 80h ;call the kernel

asciihextohex: ;procedure

mov ecx,2 ;copy 2 into ecx

mov dl,0 ;copy 0 into dl

top: rol dl,4 ;roll contents of dl by 4 bits

mov al,[esi] ;copy esi contents into al

cmp al,39h ;compare al with 39h

jbe down ;jump to down, if al is below or equal

sub al,07h ;subtract 07h

down: sub al,30h ;subtract 30h

add dl,al ;add al with dl

inc esi ;increment esi

loop top ;jump if ecx not equal to zero,decrement ecx

ret