```
section .data
 array db 11h,59h,33h,22h,44h
 msg1 db 10,"ALP to find the largest number in an array",10
 msg1_len equ $ - msg1

 msg2 db 10,"The Array contains the elements : ",10
 msg2_len equ $ - msg2

 msg3 db 10,10, "The Largest number in the array is : ",10
 msg3_len equ $ - msg3
section .bss
 counter resb 1
 result resb 4

%macro write 2
 mov rax,1
 mov rdi,1
 mov rsi,%1
 mov rdx,%2
 syscall
%endmacro
section .text
 global _start

_start:

 write msg1 , msg1_len
 write msg2 , msg2_len

 mov byte[counter],05
 mov rsi,array
next: mov al,[rsi]
 push rsi
 call disp
 pop rsi
 inc rsi
 dec byte[counter]
 jnz next

 write msg3 , msg3_len

 mov byte[counter],05
 mov rsi, array
 mov al, 0 ; al is an 8 bit register , al stores max
repeat: cmp al,[rsi] ;cmp opr1 , opr2 : opr1 - opr2
 jg skip
 mov al,[rsi]
skip: inc rsi
 dec byte[counter]
```

```
 Jnz repeat

 call disp
 mov rax,G0
 mov rdi,1
 syscall
disp:
 mov bl,al ;store number in bl
 mov rdi, result ;point rdi to result variable
 mov cx,02 ;load count of rotation in cl
up1:
 rol bl,04 ;rotate number left by four bits
 mov al,bl ;move lower byte in dl
 and al,0fh ; get only LSB
 cmp al,09h ;compare with 39h
 jg add_37 ;if greater than 39h skip add 37
 add al,30h
 jmp skip1 ;else add 30
add_37: add al,37h
skip1: mov [rdi],al ;store ascii code in result variable
 inc rdi ;point to next byte
 dec cx ;decrement the count of digits to display
 jnz up1 ;if not zero jump to repeat

 write result , 4

 ret

```

Output :