Practical - 3

Problem statement: Write an X86/64 ALP to find the largest of given Byte/Word/Dword/64-bit numbers.

Program:

```
section .data
    arr_msg db &#39;Array Elements Are:: &#39;,10
    dq &#39;0fa10001h&#39;,10
    dq &#39;0b200002h&#39;,10
    dq &#39;0fff0003h&#39;,10
    dq &#39;0d400004h&#39;,10
    dq &#39;0fffffffh&#39;,10
    arr_len : equ $-arr_msg
    larg_msg db &#39;Largest Number is::&#39;
    larg_len: equ $-larg_msg

    nwline db 10
    array dq 0fa10001h,0b200002h,0fff0003h,0d400004h,0fffffffh   ;array
elements
    arrcnt dd 05h
section .bss
    arr_num resb 16
    large resq 1

%macro dispmsg 2
    mov rax,1    ;System call for write
    mov rdi,1    ;standard output stream
    mov rsi,%1    ;message start address
    mov rdx,%2    ;message length
    syscall
%endmacro
section .text
    global _start
_start:

    dispmsg arr_msg,arr_len

    mov rsi,array
    mov rcx,[arrcnt]
    mov rax,[rsi]

    dec rcx
lup1:    add rsi,08    ;Point to next element
    cmp rax,[rsi]
    ja lskip1
    xchg rax,[rsi]
lskip1:    loop lup1
    mov [large],rax
    dispmsg larg_msg,larg_len
    mov rbx,[large]
    call disp_num
    dispmsg nwline,1
```

```
exit:   mov rax,60
    mov rdi,0
    syscall

disp_num:
    mov rdi,arr_num    ;point esi to buffer
    mov rcx,16   ;load number of digits to display

dispup1:
    rol rbx,4   ;rotate number left by four bits
    mov dl,bl   ;move lower byte in dl
    and dl,0fh   ;mask upper digit of byte in dl
    add dl,30h   ;add 30h to calculate ASCII code
    cmp dl,39h   ;compare with 39h
    jbe dispskip1   ;if less than 39h akip adding 07 more
    add dl,07h   ;else add 07
dispskip1:
    mov [rdi],dl   ;store ASCII code in buffer
    inc rdi   ;point to next byte
    loop dispup1   ;decrement the count of digits to display
    ;if not zero jump to repeat
    dispmsg arr_num,16

    ret
```

Output:

```
atharva@atharva:~$ gedit ass3.asm

atharva@atharva:~$ nasm -f elf64 ass3.asm

atharva@atharva:~$ ld -o ass3 ass3.o


atharva@atharva:~$ ./ass3

Array Elements Are::

0fa10001h

0b200002h

0fff0003h

0d400004h

0fffffffh

Largest Number is::000000000fffffffh
```