Jadavpur University

Department of Electronics and Tele-Communication Engineering

Faculty of Engineering and Technology

System Software Lab

UG-IV Semester-I

Submitted By – Oumi Mandal

Roll - 002210701135

Group – G2

DAY-2

Q1. Write an Assembly language program to find the maximum element in an array.

CODE:

```
section .data
   arr    dq   10, 25, 3, 47, 18, 99, 42   ; array of 64-bit integers
   n      equ ($ - arr) / 8            ; number of elements (7 here)
   fmt    db  "Maximum value = %d", 10, 0 ; printf format string

section .bss
   maxval  resq 1                    ; reserve space for maximum

section .text
   extern printf
   global _start

_start:
   ; ----------------------------------
   ; Initialization
   ; ----------------------------------
   mov    rcx, n           ; loop counter = number of elements
   mov    rbx, arr          ; rbx points to start of array
   mov    rax, [rbx]        ; first element → rax
   mov    [maxval], rax      ; store as current max
   add    rbx, 8            ; move to second element
   dec    rcx              ; processed one element already

find_max_loop:
   cmp    rcx, 0           ; check if done
   je     done             ; exit loop if rcx == 0

   mov    rdx, [rbx]        ; load next element
   cmp    rdx, [maxval]      ; compare with current max
   jle    skip_update        ; if ≤ current max, skip update
   mov    [maxval], rdx      ; else update max
```

```
skip_update:
   add    rbx, 8            ; move to next element
   dec    rcx              ; decrement counter
   jmp    find_max_loop      ; repeat

done:
   ; ----------------------------------
   ; Print result using printf
   ; ----------------------------------
   mov    rdi, fmt          ; 1st arg: format string
   mov    rsi, [maxval]      ; 2nd arg: max value
   xor    rax, rax          ; rax = 0 (for varargs call)
   call   printf

   ; ----------------------------------
   ; Exit program
   ; ----------------------------------
   mov    rax, 60           ; sys_exit
   xor    rdi, rdi          ; return code = 0
syscall
```