**Q1. Assembly language program to find largest number in an Array.**

section .data

    array db 10, 5, 8, 2, 15, 3

    array\_size equ $-array

section .text

    global \_start

\_start:

    mov esi, array

    mov cl, [esi]

    inc esi

    dec array\_size

loop\_start:

    cmp byte [esi], cl

    jle skip\_update

    mov cl, [esi]

skip\_update:

    inc esi

    dec array\_size

    jnz loop\_start

    mov eax, 1

    xor ebx, ebx

    int 0x80

Q2. Assembly language program to find smallest number in an array.

section .data

    array db 10, 5, 8, 2, 15, 3

    array\_size equ $-array

section .text

    global \_start

\_start:

    mov esi, array

    mov cl, [esi]

    inc esi

    dec array\_size

loop\_start:

    cmp byte [esi], cl

    jl update\_smallest

    inc esi

    dec array\_size

    jnz loop\_start

    jmp done

update\_smallest:

    mov cl, [esi]

    inc esi

    dec array\_size

    jnz loop\_start

done:

    mov eax, 1

    xor ebx, ebx

    int 0x80

Q3. Assembly language program for adding to two arrays

section .data

    array1 db 1, 2, 3, 4, 5

    array2 db 6, 7, 8, 9, 10

    array\_size equ 5

section .bss

    result\_array resb 5

section .text

    global \_start

\_start:

    mov esi, array1

    mov edi, array2

    mov ebx, result\_array

    mov ecx, array\_size

add\_arrays:

    mov al, [esi]

    add al, [edi]

    mov [ebx], al

    inc esi

    inc edi

    inc ebx

    loop add\_arrays

    mov eax, 1

    xor ebx, ebx

    int 0x80

Q4. ) Assembly language program to separate even and odd numbers from an array.

section .data

    array db 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

    array\_size equ 10

section .bss

    even\_array resb 10

    odd\_array resb 10

section .text

    global \_start

\_start:

    mov esi, array

    mov edi, even\_array

    mov ebx, odd\_array

    mov ecx, array\_size

separate\_numbers:

    mov al, [esi]

    test al, 1

    jz store\_even

    mov [ebx], al

    inc ebx

    jmp next\_iteration

store\_even:

    mov [edi], al

    inc edi

next\_iteration:

    inc esi

    loop separate\_numbers

    mov eax, 1

    xor ebx, ebx

    int 0x80

**Q5.** Assembly language rogram to find prime numbers between a given range

section .data

    start\_number dw 10

    end\_number dw 30

    newline db 10

section .text

    global \_start

\_start:

    mov cx, start\_number

check\_prime:

    mov bx, 2

    mov ax, cx

prime\_loop:

    cmp bx, ax

    jg next\_number

    mov dx, 0

    div bx

    cmp dx, 0

    je next\_number

    inc bx

    jmp prime\_loop

next\_number:

    cmp ax, cx

    jne increment

    mov eax, 4

    mov ebx, 1

    mov ecx, cx

    mov edx, 2

    int 0x80

    mov eax, 4

    mov ebx, 1

    mov ecx, newline

    mov edx, 1

    int 0x80

increment:

    inc cx

    cmp cx, end\_number

    jle check\_prime

    mov eax, 1

    xor ebx, ebx

    int 0x80

**Q6.** Assembly language program to find factorial of the given number.

section .data

    number dw 5

section .text

    global \_start

\_start:

    mov cx, number

    mov ax, 1

factorial\_loop:

    mul cx

    loop factorial\_loop

    mov eax, 1

    xor ebx, ebx

    int 0x80