COAL A0-2 21K-3278 Sohaib Sarosh Shamsi

Q1)

mov result, eax

ESP : 00001FF4h

ESP : 00001FE5h

POP ESI

ADD [ESI], EAX [ESI] : 27h

add eax, ecx ; eax = 5

	Address
1150 0004h	00001FE5h 00001FE9h
1150 0000h	00001FECh 00001FF0h
5	→ 00001FF4h
6	→ 00001FF8h

1FF0h	OFFSET[x]
1FF4h	5
1FF8h	6

Q2)

(a)

```
  0 0 1 0 0 1 0 1
  0 0 0 0 1 0 0 1
  ---------------
  0 0 0 0 0 0 0 1
```

ZF = 0

(b)

```
  0 0 1 0 0 1 0 0
  0 0 0 0 1 0 0 1
  ---------------
  0 0 0 0 0 0 0 0
```

ZF = 1

Q3)

```
.data
    array1 sdword 40, -90, -67, 98, 78, -45, 0, 32
    array2 sdword 8 DUP(?)

.code
main PROC
mov esi, offset array1
mov ecx, lengthof array1
push FD
push ecx.
call posvals
pop FD
exit
main endp

posvals PROC

    mov edi, offset array2
    mov eax, 0
    pop ecx

    L1:
        cmp eax, [esi]

        JGE Ignore

        mov ebx, [esi]
        mov [edi], ebx
        ADD esi, 4
        ADD edi, 4
        Jmp endd

    ignore:
        mov ebx, [esi]
        Neg ebx
        mov [edi], ebx
        ADD esi, 4
        ADD edi, 4

    end: loop L1

    ret
posvals endp.
```

Q4.

```
.data
N   sdword ?
a   sdword ?
b   sdword ?

.code
main PROC
    mov eax, 3
    mov ebx, a
    mov edx, b
    L1:
        cmp N, ebx
        JNE con1
        JMP endd

        con1:
            cmp N, ebx
            JGE con2
            JMP next

        con2:
            cmp N, edx
            JLE endd

        next:
            mov ecx, N
            sub ecx, 2
            mov N, ecx
            JMP outt

        end:
            dec N

        out:
            mov ecx, N
            cmp ecx, 0
            JG L1

    exit
main endp.
```

Q5)

```
.data
st1 byte "Enter a number", 0
.code
main PROC
        mov edx, offset st1
        call Writestring
        mov eax, 0
        call Readint

        cmp al, 1
        JE meet1

        cmp al, 3
        JE meet1
        JNE con2

meet1:
        mov al, '0'
        call Writechar
        Jmp endd

con2:
        cmp al, 2
        JE meet2

        cmp al, 4
        JE meet2
        JNE endd

meet2:
        mov al, 'e'
        call WriteChar
        Jmp endd

endd:
        exit
        main endp
```

Q6.)

```
.data
a dword 100
b dword 200
c dword 0
i dword ?
j dword ?

.code
main PROC
    mov ecx, 4
    L1:
        mov eax, a
        add b, eax
        mov al, b
        call writeint .
        Push ecx
        mov ecx, 4.
        L2:
            mov ecx, 4
            dec a
            mov ebx, c
            add ebx, 10
            mov c, ebx
        Loop L2

        Pop ecx al, a
        mov al, a
        Call writeint
        mov al, c
        Call writeint
    Loop L1

    exit
main endp
```

Q7)

```
.data
str1 byte "Enter a number: ", 0
.code
main PROC
    mov edx, offset str1
    call WriteString
    call ReadInt

    mov ecx, eax
    push eax

    L1:
        Push ecx

        L2:
            call WriteInt
            dec al
        Loop L2

        pop ecx
        pop eax
    Loop L1

exit
main endp.
```

Q8).

```
.data
    a byte 11110101b
    str1 byte "even parity", 0
    str2 byte "odd parity", 0

.code
    main PROC
    movzx eax, a
    mov ebx, 0      ; number of 1s.
    mov ecx, 8

    Loop1:
        rcr al, 1
        jnc L1
        inc ebx
        L1:
        loop loop1

    mov eax, ebx
    mov bl, 2
    div bl

    movzx eax, ah

    cmp eax, 0
    je evenn

    mov edx, offset str2
    call writestring
    jmp endd.

    evenn:
        mov edx, offset str1
        call writestring

    endd:
        exit
        main ENDP
        END main.
```