Direct addressing and Indirect addressing

TASK-1

A program to add three numbers using byte variables

[org 0x0100]

; mov ax, 0x8787

; xor ax, ax ; We need to make sure AX is empty

mov ah, [num1] ; Intel Sotware Developer Manual

mov bl, [num1+1]

add ah, bh

mov bh, [num1+2]

add ah, bh

mov [num1+3], ah

mov ax, 0x4c00

int 0x21

num1: db 5, 10, 15, 0

TASK-2

[org 0x0100]

mov ax, 0x8787

xor ax, ax ; we need to make sure AX is empty

mov al, [num1]

mov bl, [num1+1]

add al, bl

mov bl, [num1+2]

add al, bl

mov [num1+3], al

; mov ax, bl

mov ax, 0x4c00

int 0x21

num1: db 5, 10, 15, 0

TASK-3

[org 0x0100]

; mov ax, 0x8787

; xor ax, ax

mov ah, [num1]

mov bl, [num1+1]

add ah, bh

mov bh, [num1+2]

add ah, bh

mov [num1+3], ah

mov ax, 0x4c00

int 0x21

num1: db 5, 10, 15, 0

TASK-4

[org 0x0100]

mov ax, 0x8787

xor ax, ax ; We need to make sure AX is empty!

mov al, [num1]

mov bl, [num1+1]

add al, bl

mov bl, [num1+2]

add al, bl

mov [num1+3], al

; mov ax, bl

mov ax, 0x4c00

int 0x21

num1: db 5, 10, 15, 0

TASK-5

[org 0x0100]

xor ax, ax

mov bx, num1

add ax, [bx]

add bx, 2

add ax, [bx]

add bx, 2

add ax, [bx]

add bx, 2

mov [result], ax

mov ax, 0x4c00

int 0x21

TASK-6

[org 0x0100]

; for (int c = 3 c > 0 c--) {

; result += data[c];

;}

; initialize stuff

mov ax, 0 ; reset the accumulator

mov cx, 3 ; set the iterator count

mov bx, num1 ; set the base

outerloop:

add ax, [bx]

add bx, 2

sub cx, 1

jnz outerloop

mov [result], ax

mov ax, 0x4c00

int 0x21