Quiz 3 Solution

Set A

Consider the following do ... while loop. Here Array is WORD array. Translate the following code into Assembly code (MASM). You are not allowed to make any logical reduction to the code.

array[4] = {1,2,3,4}

esi = 0

ax = 4

bx = 0

do {

     bx = bx + array[esi]

     ax = ax - 1

}while (esi < ax)

Solution:

.data

Array WORD 1,2,3,4

.code

mov esi, 0

mov ax, 4

mov bx, 0

begin\_loop:

add bx, array[esi]

dec ax

; check the loop condition

Movsx eax, ax

cmp esi, eax

jl begin\_while

Set B

Consider the following while loop. Here Array is WORD array. Translate the following code into Assembly code (MASM). You are not allowed to make any logical reduction to the code. You need to implement both while loop and if condition.

array[4] = {1,2,3,4}

esi = 0

while(esi < ax){

     if(ax != 0){

          bx = bx + 1

          ax = ax – 1

     }

}

Answer:

.data

Array WORD 1,2,3,4

.code

mov esi, 0

begin\_while:

; check the loop condition

movsx eax, ax

cmp esi, eax

JGE next ;when esi >= ax, break the loop

; check if condition

cmp ax, 0

JE After\_if\_block ; when ax == 0 ; skip the if block

; otherwise enter if block

inc bx

dec ax

After\_if\_block:

jmp begin\_while

Next:

Set C

Consider the following while loop. Here Array is WORD array. Translate the following code into Assembly code (MASM). You are not allowed to make any logical reduction to the code. You need to implement both while loop and if condition.

array[4] = {1,2,3,4}

esi = 0

while(true){

bx = bx + 1

     ax = ax – 1

     if(array[esi] == 4)

          break;

}

Answer:

.data

Array DWORD 1,2,3,4

.code

mov esi, 0

begin\_while:

inc bx

dec ax

;check if condition

cmp array[esi], 4

JE Next

JMP begin\_while

Next:

Question 3.

Set A

Consider the following code. Here Array is WORD array. Translate the following code into Assembly code (MASM). You are not allowed to make any logical reduction to the code.

array[4] = {1,2,3,4}

esi = 0

if(esi < ax OR array[esi] <= bx){

          bx = bx + array[esi]

          ax = ax – 1

}

Solution:

.data

array WORD 1,2,3,4

.code

mov esi, 0

; check the first condition esi < ax

movsx eax, ax

cmp esi, eax

JL if\_block ; when condition is true (esi < ax), enter the if block

cmp array[esi], bx ;otherwise check the second condition

JG next ;when condition is false(array[esi]>bx),skip the if block

;otherwise enter the if block

if\_block:

add bx, array[esi]

dec ax

next:

Set B

Consider the following code. Here Array is WORD array. Translate the following code into Assembly code (MASM). You are not allowed to make any logical reduction to the code.

esi = 0

if(esi < ax OR cx <= bx){

          bx = bx + 2

          ax = ax – 1

}

else{

     bx = bx + 1

     ax = ax + 4

}

Solution:

mov esi, 0

; check the first condition

movsx eax, ax

cmp esi, eax

JL if\_block ;when condition is true(esi<ax),enter the if block

cmp cx, bx ; otherwise compare second condtion cx <= bx

JG else\_block ;when second condition is false(cx>bx),goto else block

;otherwise enter the if block

if\_block:

add bx, 2

dec ax

JMP next ; after executing if block skip the else block

else\_block:

inc bx

add ax, 4

next:

Set C

Consider the following code. Here Array is WORD array. Translate the following code into Assembly code (MASM). You are not allowed to make any logical reduction to the code.

if(esi > ax AND cx >= bx){

          bx = bx + 2

          ax = ax – 1

}

else{

     bx = bx + 1

     ax = ax + 4

}

Solution:

movsx eax, ax

cmp esi, eax

JLE else\_block ;when first exp is false (esi<=ax), goto else block

cmp cx, bx ;otherwise compare the second instruction (cx >=bx)

JL else\_block ;when second condition false(cx<bx),go to else block

; when both are true enter if\_block

if\_block:

add bx, 2

dec ax

jmp next ; skip the else block after executing if block

else\_block:

inc bx

add ax, 4

next: