2018:

[25 Marks]

; Given the definitions

; N = 8

; .data

; A DWORD DUP N (0)

; B DWORD DUP N (0)

; Write MASM code that will put the larger of corresponding components of A and B into

; A and the smallest into B, i.e., for i in 0..7, A[i]should be greater than or equal B[i].

; For example, given A = [2,2,2,2,2,-10,2,2] and B = [0,2,2,4,2,-12,2,10], the

; final value of A should be [2,2,2,4,2,-10,2,10] and the final value of B should be

; [0,2,2,2,2,-12,2,2].

;

; Data & Code

;

N= 8

.data

A DWORD N DUP(0)

B DWORD N DUP(0)

i DWORD 0

message BYTE 'Press OK to continue',0

caption BYTE 'Welcome to CA296',0

.code

main:nop

invoke version

invoke fillWithRandomValues, ADDR A, N, 20

invoke fillWithRandomValues, ADDR B, N, 20

invoke writeArray, ADDR A, N

invoke writeArray, ADDR B, N

mov Ai, offset A

mov Bi, offset B

find\_max:

mov ebx, 0 ; ebx = 0

cmp i, LENGTHOF A ; if i < len(A)

je finish ; we are done

mov eax, i ; i = eax

mov edx, [A + eax\*4] ; edx = A[i]

cmp edx, [B + eax\*4] ; compare our A[i] to B[i]

jl switcheroo ; if less we need to swap em

inc i ; else we need to go to next position

jmp find\_max

switcheroo:

mov ecx, [B + eax\*4] ; first make sure we store B[i] - use ecx as our temp

mov [B + eax\*4], edx ; put min into B[i]

mov [A + eax\*4], ecx

inc i

jmp find\_max

finish:

invoke writeArray, addr A, N

invoke writeArray, addr B, N

invoke MessageBox,0,ADDR message,ADDR caption,MB\_OK

invoke ExitProcess,0

end main

2017:

[25 Marks]

;

; Write MASM code to read a list of integers from the console until either a negative

; value is detected or there is a duplicate value.

;

; For example if the user supplies the values 23 45 76 300 -1

; your code should output the message: There were no duplicate values.

;

; However if the user supplies the values 23 45 76 300 45 your code

; should output the message: Duplicate value 45 detected.

; Hint: Use the stack.

;

; Data & Code

;

.data

i DWORD -1

no\_d\_found BYTE "There were no duplicate values.", 10, 0

d\_found BYTE "Duplicate value %d detected.", 10, 0

message BYTE 'Press OK to continue',0

caption BYTE 'Welcome to CA296',0

.code

main:nop

invoke version

enternum:

invoke readInteger

cmp eax, 0 ; compare integer with 0

jl no\_dups ; if negative jump to exit

mov ebx, i ; mov i into ebx

inc i ; increment i

search\_array:

cmp ebx, -1 ; compare -1 with old i

je addStack ; if its even, jump to addStack

mov ecx, ebx ; move old i into ecx

sal ecx, 2 ; multiply old i by 4

add ecx, esp ; add old i to stack pointer

mov ecx, [ecx] ; overwrite ecx with the value in memory location in ecx

cmp eax, ecx ; compare ecx with number entered

je duplicate ; if number is there, jump to exit at duplicate

dec ebx ; decrement i to look for duplicate values, if none are found

; ebx will become -1

jmp search\_array ; continue searching array

addStack:

push eax ; put number entered onto stack

jmp enternum ; go back to request next number

duplicate:

invoke crt\_printf, addr d\_found, eax ; printing duplicate found

jmp finish

no\_dups:

invoke crt\_printf, addr no\_d\_found, eax ; print no duplicate found

finish:

invoke ExitProcess, 0

invoke MessageBox,0,ADDR message,ADDR caption,MB\_OK

end main

2017R:

[25 Marks]

;

; Given the definitions

; N = 50

; .data

; A DWORD DUP 2\*N (0)

; B DWORD DUP N (0)

; Write MASM code that adds pairs of elements in A and stores the result in B such that

; B[i] will be assigned the value A[2\*i] + A[2\*i+1] where 0 ≤ i < N.

; For example, if N is 3 and A is the array [2,3,5,6,-10,10], then B will set to [5,11,0].

;

; Data & Code

;

N = 10

.data

A DWORD 2\*N DUP(0)

B DWORD N DUP(0)

i DWORD 0

j DWORD 1

k DWORD 0

message BYTE 'Press OK to continue',0

caption BYTE 'Welcome to CA296',0

.code

main:nop

invoke version

invoke fillWithRandomValues, ADDR A, 2\*N, 30

add\_em:

cmp j, LENGTHOF A ; when j == len(A)

jge finish ; end

mov eax, i ; eax = i

mov ebx, j ; ebx = j

mov ecx, k ; ecx = k

mov edx, [A + eax\*4] ; edx = A[i]

add edx, [A + ebx\*4] ; edx = edx + A[j] (sum a[i]+a[i+1])

make\_b:

mov [B + ecx\*4], edx ; B[k] = edx (sum)

add i, 2 ; i += 2

add j, 2 ; j += 2

inc k ; i += 1

jmp add\_em ; go around again

finish:

invoke writeArray, ADDR A, 2\*N

invoke writeArray, ADDR B, N

invoke MessageBox,0,ADDR message,ADDR caption,MB\_OK

invoke ExitProcess,0

end main

2016:

[25 Marks]

; A pair of integer values can be represented by the following MASM structure.

; Pair STRUCT

; x SDWORD

; y SDWORD

; Pair ENDS

; Consider the following fragment of MASM code that creates an array of pairs by

; allocating a block of memory.

; N= 400

; .data

; arrayAddr DWORD 0

;

; .code

; invoke allocate, N

; mov arrayAddr, eax

;

; Using the CA296 function random(), write MASM code to fill this array with

; Random values. For each pair we require

; 1. 0 >= x = 100

; y + x = 100

Can’t answer this question because we didn’t learn pairs/SDWORDs

2016R:

[25 Marks]

; Assuming that N is an integer constant thsat has already been defined, consider

; the following array definition

; .data

; array SDWORD N Dup(0)

; Assuming this array has been modified, write MASM code to determine if the array

; is a palindrome. Your program should output:

; The array is a palinddrome

; Or The array is not a palindrome

; Note. The array isd a palindrome if it reads the same in both directions. For example,

; the aray [3,5,7,2,4] is not a palindrome, whereas the array [4 2 6 2 4] is a

; palindrome.

Can’t answer this question because we didn’t learn SDWORDs

2016 SAMPLE:

[25 Marks]

; Consider the following fragment of MASM code that creates an array of signed 32-bit

; integers by allocating a block of memory.

; N = 400

; .data

; arrayAddr DWORD 0

; .code

; invoke allocate,N

; mov arrayAddr,eax

; Using the CA296 library function random(), write MASM code to fill this array with

; random, non-zero integers x such that −50 < x < 50. Negative values should be

; stored at the bottom end of the array and positive values stored at the top end of the

; array, i.e., no positive value should precede a negative value in the array.

;

; Data & Code

;

N = 50

.data

arrayAddr DWORD 0

i DWORD 0 ; index from position 0

j DWORD 0 ; index from N-1

message BYTE 'Press OK to continue',0

caption BYTE 'Welcome to CA296',0

.code

main:nop

invoke version

mov eax, N

sal eax, 2 ; N \* 4 used for storage

invoke allocate, eax

mov arrayAddr, eax

mov eax, N ; these next three lines make j = N-1

dec eax

mov j, eax

get\_num:

mov ebx, i

cmp ebx, j

jg finish

invoke random, 50

cmp eax, 0

jl lesser

jg greater

jmp get\_num

lesser:

mov ebx, i

sal ebx, 2 ; multiply by 4 for next space in array

add ebx, arrayAddr

mov [ebx], eax

inc i ; i += 1

jmp get\_num

greater:

mov ecx, j

sal ecx, 2 ; multiply by 4 for next space in array

add ecx, arrayAddr

mov [ecx], eax

dec j ; j -= 1

jmp get\_num

finish:

invoke writeArray, arrayAddr, N

invoke MessageBox,0,ADDR message,ADDR caption,MB\_OK

invoke ExitProcess,0

end main