

Assembly

11.15.11

page 1

-- Review for Exam 2

7. create two dimensional table and return the value
; int findRecord(int ~~+~~, int)

findRecord proc Near32

push ebp

mov ebp, esp

push ebx push esi

mov esi, [ebp+12]

mov ebx, [ebp+8] ; points to table

mov eax, -1

stloop:

cmp dword ptr [ebx], -1 ; end of table?

je done ; find key?

cmp [ebx], esi

je found

add ebx, 8 ; ebx points to next record

jmp stloop

found:

mov eax, [ebx+4]

done:

pop ebx

pop esi

pop ebp

ret

8.
push iValue
push offset iTableIndex
call findRecord
add esp, 8
cmp eax, -1
jg displayFound
puts strError
jmp around

displayFound:
puts strFound

10.
mov ax, word ptr iD
mov dx, word ptr iD+2
idiv sB
cwde
cdq
mov ebx, offset qA
mov [ebx], eax
mov [ebx+4], edx

15. ; int myFun(int N)

myFun PROC

push ebp

mov ebp, esp

sub esp, 12

mov eax, 1 ; assume N is 1

N equ [ebp+8] ; N

cmp dword ptr N, 0 ; if =, done

je done ; return 1

cmp dword ptr N, 1

je returnTwo ; return 2

mov esi, N

dec esi ; esi = N-1

t equ [ebp-4]

v equ [ebp-8]

z equ [ebp-12]

push esi

call myFun

add esp, 4

mov t, eax

dec esi

push esi

call myFun

add esp, 4

mov v, eax

add eax, t

jmp done

returnTwo:

mov eax, 2

done:

add esp, 12

pop ebp

ret

11.15.11

page 4

- mod ebx, offset strOut =
lea ebx, strOut

21. 001004CD CS
+ FFFFFFF5

 00100492