CSCI2150Quiz12F11ConditionalDirectivesStructures Name: Brandon Buchanan

Given the below structure definition and .data segment, answer the questions

90

```
employee    struct
ID          byte    9 dup(?)
eName       byte    20 dup(?)
age         word    ?
years       word    ?
salary      dword   ?
            dword   ?
            dword   ?
            dword   ?
            dword   ?  ;5 years of salaries, most recent first
Employee    ends
```

1. (10) Write the assembly statement that will set up a structure named **fred** using the above definition whose ID is 222334567, and whose name is Fred Smith and he is 25 years old.

fred employee {("222334567", "Fred Smith", 25)}

-1

2. (10) Write the assembly statement that will set up an array of structures referenced by **employees** (100)

employees employee 100 dup(< >)

3. (10) Assuming that you were able to create the reference **fred** to a structure of this type, write the code that will do the following:

Assuming that the field called referenced as **salary** is actually a reference to an array of salaries where the first of the five (the most-left one) represents the most recent salary, the second field represents a year ago, the third field represents 2 years ago, etc., Write the sequence of assembly instructions which will assign to fred the values 35000, 34000, 32000, 25000, 21000 respectively.

```
mov  fred.salary, 35000
mov  fred.salary+4, 34000
mov  fred.salary+8, 32000
mov  fred.salary+12, 25000
mov  fred.salary+16, 21000
```

-1

4. (10) Suppose that **employees** references an array of 20 structures each of which has been populated with all the necessary values. Write the code which will total up the most current salary for all of the employees in the array.

```
xor  eax, eax
mov  ecx, 20
mov  ebx, offset employees
mov  esi, lengthof employee
stLoop:
    add eax, (employee ptr [ebx]).salary
    add ebx, esi
    dec ecx
    cmp ecx, 0
    jne stLoop
```

good

loop stLoop does dec ? of them

5.(15) Suppose that you have the variables x, and y as declared below, and that you used the conditional directive .IF and .ELSE to write the statements

a)
```
x    dword ?
y    dword ?
```

b)
```
x    sdword ?
y    sdword ?
```

Write "equivalent" assembly instructions that would illustrate what the conditional directives would produce in each of case a) and case b). The reason that I say "equivalent" is that it is hard to predict the order of comparision, but you CAN predict how the comparison will be made:

```
.IF ( x < y)
        mov    eax, 5
.ELSE
        move   eax, 10

.ENDIF
```

A.

```
mov ebx, x
cmp ebx, y
jb five
jnb ten

five:
    mov eax, 5
    jmp done
ten:
    mov eax, 10

done:
```

yes!

B.

```
mov ebx, x
cmp ebx, y
jl five
jnl ten

five:
    mov eax, 5
    jmp done
ten:
    mov eax, 10
done:
```

-0

6.(15) Suppose that the values for iVal1 and iVal2 are always positive. Write the equivalent assembly instructions using the conditional directives for the following Java statements:

```
if (iVal1 > iVal2)
        iVal1 =20;
else
        iVal1 = 30;
```

```
mov eax, iVal1
.IF (eax > iVal2)
    mov iVal1, 20
.ELSE
    mov iVal1, 30
.ENDIF
```

7. (15) Suppose that you have three fields lVal1, lVal2, iVal3 (where the prefix "el" means these are long ints (8 bytes).

```
lVal1   qword ?          ;8 bytes long
lVal2   qword ?
lVal3   qword ?
```

Write the code which would correctly do the sum lVal3 = lVal1 + lVal2

```
mov eax, dword ptr lVal1 + 4
mov ebx, dword ptr lVal2 + 4
add eax, ebx
mov lVal3 + 4, eax

mov eax, dword ptr lVal1
mov ebx, dword ptr lVal2
add eax, CF
add eax, ebx
mov lVal3, eax
```

8. (15)Suppose that you have defined the below structure:

```
node    struct
sData   dword
next    dword
node    ends
```

And that you have an array of 100 of these nodes referenced by **myStorage**. Suppose, further the value stored in the first next field is the **address** of the very next node, the second next field is the **address** of the third node. etc. If I were to draw a table representing this it would look like the following (I'll put the address of the first node beside it) according to the below table

	sData	next
CE000		CE008
CE008		CE010
CE010		CE018
etc.		etc
		-1

Suppose further that the first 99 of these nodes contains the address of the next node, but the very last node contains a -1 in the next field. Write the code which will initialize the block of nodes.

```
myStorage  node  100 dup(?)
mov  esi, 0CE000h
mov  ecx, 99

stLoop:
    mov  (node ptr [esi]).next, esi + 8
    add  esi, 8
    dec  ecx
    cmp  ecx, 0
    jne  stLoop

mov  (node ptr [esi]).next, -1
```