Exam 3

Fall 2009

CSCI 2160

Name: _____
Last First Middle

Note: The total points on this exam are > 100. Your grade will be determined by a percentage

For Questions 2,3,4 use the portion of the data segment given in #2

1. (12pts -each answer worth 1 pt) Given the below portion of a data segment of a program, determine the values for the location counter and write the actual bytes in memory.

```
02FC  X      dd    25,29          _____________________________________
_____ Y      DW    2 dup(-1)      _____________________________________
_____ T      DB    -1,14          _____________________________________
_____        DB    2 dup(23)      _____________________________________
_____ strPr  DB    "123",100      _____________________________________
_____ Z      DD    X              _____________________________________
_____
```

2. (6 pts) Assuming that the following instructions have occurred, give the value that will be in the AX register after the instruction is executed. Assume for all questions that the same two instructions were executed just above a) . Be sure to reference #1 where necessary and note the question is asking what is in **AX** register.

```
MOV    AX, 0ABCDh
MOV  EBX,02FCh

 a)  MOV    AX,BX          ___________
 b)  MOV    AX,[EBX+4]     ___________
 c)  MOV    AEX, T[3]      ___________
```

3). Given the below coding and addresses

(8) describe what is on the stack just as the system has completed executing the CALL statement. (you do not have to write the values in reverse-byte order even though the stack IS memory - just for convenience sake, but you DO have to write in HEX). The memory addresses are in the first column. Write EVERY hex digit of the answer.

```
04220024h  A        dw      8h
04220026h  B        dw      14h
04220028h  C        dd      25h
0422002Ch  D        dd      A


04110150   PUSH     offset B
04110154   PUSH     C
04110158   PUSH     D
0411015C   CALL     MYFUNCT
04110160   ADD      ESP,W   ;restore ESP
04110164   MOV      A,AX


041C0002   MYFUNCT PROC  NEAR32
```

Write the values for the stack register in the first column.

ESP→100h just before first push

100h	xxxxxx

(2) What value will be required to replace the **W** that is in the instruction that restores ESP? _____

(2) What are the contents of the EIP register at the exact completion of the CALL instruction? _____

4) (10 pts) Given the below data segment, write the assembly code that would be generated as a result of the C++/Java statements: You may assume that a prefix of “s” means the variable is a “short” (2-byte), “i” means “int” – a dword, and “q” means the value is 8-bytes (a long), “b” means “byte”

a) i A = sB * sC;

b) iB = bE /dC

5. (10 pts) Write the recursive function func defined below in assembly language using the high-level PROC directives.

```
             1     when n=0
             2     when n=1
f(n) = n *f(n-1) + (n-1) * f(n-2)     for n>2
```

6) (15 pts) Write the function **getavail** (use PROC directives)that accepts the address of a storage location named **Avail**.. That storage location (**Avail**) contains a 4-byte value that is the address of the first node available in a linked list. The node has the below structure:

```
node  struc
key         dd    ?     ;a 4-byte value representing the “key” of a record
RecNum      dd    ?     ;a 4-byte value representing the record number in a file
Next        dd    ?     ;the address of the “next” cell in the chain
      Ends
```

The field “Next” is set to -1 if there is nothing else in the list, otherwise, it contains the address of the “next” available node. . If Avail has a value of -1 there is nothing “available”. The method returns -1 if there is nothing available, otherwise, it returns the value in Avail.

7.(15) Given the above node structure; pred contains the address of the predecessor to a node; succ contains the address of the successor of the node in a linked list that keeps cells arranged in increasing order; given cell contains the address of the cell to be “inserted” into the chain; start contains the address of the first cell in the chain; write the procedure insert using directives that match the below INVOKE statement:

INVOKE insert, offset start, cell, pred, succ

8. (15) Write the procedure delete that will remove a cell from a linked list. The value cell contains the address of the cell to be deleted; start contains the address of the first cell in the chain; pred contains the address of the predecessor in the chain; succ contains the address of the successor in the chain. Write the procedure using directives that matches the below INVOKE statement:

INVOKE delete, offset start, cell, pred, succ

9. (8) Given the contents of the registers and data segment: EAX: FFCE3D02 EBX: 00000E45 ESP: 0402AE28

00A3	A	dword	37
00A5	B	word	25
00A6	C	dword	A

The partial contents of the stack are given below. I have only listed the last 5 hex digits of the even-numbered addresses because of space.

2A	E2	40	52	6A	5D	F2	02	41	FF	2C	B2	22	53	7E	42	87	A5	29	33	AA	E4
2AE20		2AE22		2AE24		2AE26		2AE28		2AE2A		2AE2C		2AE2E		2AE30		2AE32		2AE34	

Give the results **after** execution of the below statements. Assume in **EACH** case that the registers are as defined above.

1) (4 pts) POP EAX EAX: _____ ESP: _____

2) (2 pts) PUSH C ESP: _____

3) (2 pts) Show below what bytes were affected in 2) by crossing out and writing above the addresses the new values:

2A	E2	40	52	6A	5D	F2	02	41	FF	2C	B2	22	53	7E	42	87	A5	29	33	AA	E4
2AE20		2AE22		2AE24		2AE26		2AE28		2AE2A		2AE2C		2AE2E		2AE30		2AE32		2AE34	

Complete the below table:

10. (13 pts)XXXXX mean you are not required to fill in the box.

Character	Hollerith Code (punches)	EBCDIC code in Binary	AScii Code in hex
F			
M			
V			
b	XXXXX	XXXXX	
7			

11. (30 pts) Given the contents of the below registers (in hex). Give what will be in the register as a result of the instructions:

```
CX: C9 A5   sal  cx,1        CX: _____________ SF ____  ZF ____ CF ____  OF ______

AX: C9 A5   shr ax,1         AX: _____________  SF ____  ZF ____ CF ____  OF ______

BX: C9 A5   sar bx,1         BX: _____________  SF ____  ZF ____ CF ____  OF ______

AX: C9 D5   rol ax,1         AX: _____________  SF ____  ZF ____ CF ____  OF ______

AX: C9 D5   ror ax,1         AX: _____________  SF ____  ZF ____ CF ____  OF ______

ECX: 34 56 78 12    shld     ecx,eax,8   ECX: ____________  CF ____      ZF _____
EAX: AB CD EF 90
                             EAX: ___________________________________      SF ____
```

12. (14) Assume that you have the method **putChar** that has been defined so that it can be invoked where the prototype is defined as below. To use it, you would first ensure that the character to display would be placed in the rightmost (low-order)byte of the dword.

```
putChar  PROTO near32 stdcall, value:dword
```

a) (4) . Assume that you would like the character C displayed on the screen. Write the instructions which would accomplish that.

b) (10) Write a macro **puts** that will display an entire NULL-terminated string on the screen using the putChar method as defined above. Make sure that you invoke the putChar method. For example, if you had the below string you would use the statement:

```
puts   msg
```

to display it on the screen.

```
msg    BYTE  10,13,"Happy Birthday",0
```

13. (20) Given the below defined storage

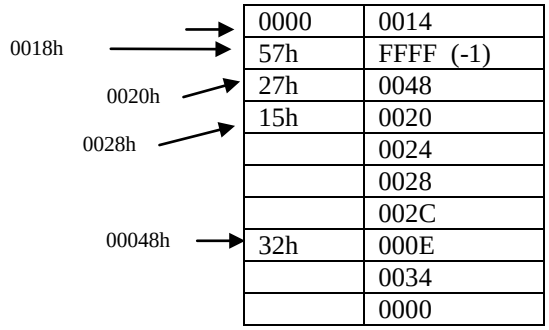
```
004   foundAddr         dword ?
008   start             dword ?
00C   numToLookFor      dword ?
010   dataTable         dword 20 dup ( 2 dup(?))
```

```
foundAddr = findNum( start, numToLookFor)
```

Write the code for the function **findNum** which will search the linked list data table IN SEQUENCE looking for the address of the Cell that contains the desired value. If the value is found, the **address** of the cell will be returned. If the value is NOT found, a value of -1 is returned. Note that the sequence is in ascending order, so do NOT look through the whole table. You should exit the sub as soon as you discover the value is NOT in the table. You have reached the end of the linked values if the value in the cell you are looking at is -1. In the example below, start has the value **0028h**. At that address is the value 15h (the smallest value in the table). The dword after that value is **0020**. Which means the next value to be displayed is at address 0020. There we see the value 0027h (the next value in sequence). The 2nd dword there is 0048 which is the address of the NEXT value in sequence , 32h etc.

Write the method assuming **stdcall** so that any used registers are automatically preserved and the method can be invoked as: (use the back if necessary)

```
INVOKE findNum, start, numToLookFor
mov foundAddr, eax
```



0000	0014
57h	FFFF (-1)
27h	0048
15h	0020
	0024
	0028
	002C
32h	000E
	0034
	0000

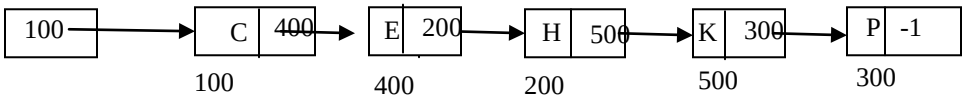
14. (15) Assume that you have the linked list above described pictorially below. The data is in the first half of the structure, and the address of the next cell in sequence is contained in the 2nd half. Suppose that you wish to delete a cell from the "chain". When the cell is removed the chain needs to be re-linked so that it stays intact. You may assume that predecessor and successor of any cell to be deleted are known.

```
start   dword   ?       ;contains the address of the first node in the chain
pred    dword   ?       ;contains the address of the predecessor of a cell
succ    dword   ?       ; contains the address of the successor of a cell
```

The prototype for the deleteCell method is below where lpStart is the address of the variable start. You need this in case the cell to be deleted is the first cell in the chain.

```
deleteCell PROTO Near stdcall, lpStart:dword, pred:dword, succ:dword
```

start



For example, if the cell containing "H" is to be deleted, pred contains 400 and succ contains 500. Once the deleteCell method is invoked, the cell with E in it would contain 500 in its 2nd dword.

Remember to allow for the cell to be deleted to be the first in the chain (the pred is -1) , in the middle of the chain, and the last cell in the chain (the successor is -1). Also, the cell to be deleted may be the only