Part 1- 20 multiple choice questions (1.5 points each)

1- For any two integers *n* and *m*, ((*n* XOR *m*) XOR *m*) produces *n*.

2- The TEST instruction always alters the destination operand.

3- ANDing an operand with 1 sets the zero flag if the operand was originally an even number.

4- The XOR instruction inverts each bit in a destination operand.

5- The JBE instruction is used when comparing unsigned integers.

6- The OR instruction can be used to find the intersection of two bit-mapped sets.

7- With unsigned operands, the CMP instruction sets the Carry flag when the destination operand is less than the source operand.

8- With signed operands, the CMP instruction makes the Sign flag equal to the Overflow flag when the destination operand is less than the source operand.

9- The MUL instruction sets the Overflow flag if the upper half of the product is not equal to zero.

10- The binary value of AL after the following instructions have executed is 11101101.

```
mov al, 01101011b
rol al,2
```

11- The hexadecimal values of DX and AX after the following instructions have executed are DX=0005 and AX=0000.

```
mov dx,5000h
mov ax,20h
mul dx
```

12- The following is a valid LOCAL declaration?

```
LOCAL index:DWORD
```

13- Local variables are stored on the runtime stack, at a higher address than the stack pointer.

14- The following is a valid LOCAL declaration:

LOCAL pArray:DWORD PTR

15- The following is a valid LOCAL declaration:

LOCAL LastName:BYTE(25)

16- In 32-bit programs, ADDR and OFFSET return the same value.

17- The CALL instruction can only have a single operand.

18- An indirect operand such as [eax + edi] can be passed to the INVOKE directive.

19- Stack parameters are usually expressed as positive offsets from ESP.

Part 2 – Short programs:

1- (20 pts) Write a program that correct an extra character in a string. For example, in "Good morrning" program should remove the extra r.

. data

str1 BYTE " Good morrning"

.code

2- (20 pts) Write a procedure named **Str_concat** that concatenates a source string to the end of a target string. Sufficient space must exist in the target string to accommodate the new characters. Pass pointers to the source and target strings. Here is a sample call:

.data

```
targetStr BYTE "ABCDE",10 DUP(0)
sourceStr BYTE "FGH",0
.code
INVOKE Str_concat, ADDR targetStr, ADDR sourceStr
```

3- (20 pts) Create a procedure named **FindLargest** that receives two parameters: a pointer to a unsigned doubleword array, and a count of the array's length. The procedure must return the value of the largest array member in EAX. Use the PROC directive with a parameter list when declaring the procedure. Preserve all registers (except EAX) that are modified by the procedure.

4- (5 pts) What will be the final value in EDX after this code executes?

```
mov     edx,1
mov     eax,7FFFh
cmp     eax,8000h
jl       L1
mov     edx,0
```

L1:

5- (5 pts) In the following code sequence, show the value of AL after each shift or rotate instruction has executed:

```
mov   al,0D4h
ror   al,3        ; a.
mov   al,0D4h
rol   al,7        ; b.
stc
mov   al,0D4h
rcl   al,1        ; c.
stc
mov   al,0D4h
rcr   al,3        ; d.
```