CSC201, Section 002
EXAM #4
November 20, 2000

Name: _____

Student ID Number: _____

(Also fold your paper lengthwise and write your name on the outside before turning in.)

Instructions

1. There are 100 points total; points for each problem are indicated.
2. No calculators or papers (except your 1-page study sheet) are allowed.
3. For all of the questions about procedures, assume our "procedure conventions" are used.

QUESTIONS

1. *(4 points)* Write a Pentium floating-point instruction that compares the numbers stored in the top two registers of the floating point stack.

2. *(12 points)* Write the Pentium floating-point code to evaluate the following expression. Include any data definitions that your code needs. Assume "var1" through "var4" have already been defined as doubleword-length floating-point values.

```
resultvar = var3 + var4 / (var2 - var1)
```

3. *(4 points)* Use EQU to define three labels. The first, LENGTH, is equal to 18, the second, WIDTH, is equal to 49, and the third, AREA, is equal to LENGTH * WIDTH.

4. *(10 points)* Write a macro named COPY2 (with parameters var1 and var2) that will generate the code

```
move  var1, var2
```

   if a flag called SASM is defined. Otherwise it will generate the code

```
mov EAX, var2
mov var1, EAX
```

5. *(6 points)* What are the steps involved in converting a high-level language program (e.g., a .c file) into an executable program (i.e., a .exe file)? (A picture, a list, or a one-sentence answer is sufficient).

6. *(6 points)* A program is written as two separate files. File 1 references the doubleword variable "xctr" in file 2; file 1 also has a jump to the label "alldone" in file 2. Show the statements that must be added to files 1 and 2 so that linking is done properly.

7. *(6 points)* Show how a procedure would save the EBX, ESI, and EDI registers on entry. Then show how it would restore their values on exit. You do not need to show any other part of the procedure.

8. *(6 points)* Show the code to pass two input parameters ("var1" and "var2"), call procedure "addem", and on return, copy the return value into the EAX register. You do not need to show any of addem's code.

9. *(14 points)* A program is executing its main routine, and the stack is empty.

The main routine calls procedure "proc2" with a single input parameter ("varx").
proc2 saves the EBX and ESI registers and calls procedure "proc3" with no input parameters.
proc3 does not save any general purpose registers, but does allocate space for one locally-scoped variable (a doubleword).
Show the contents of the stack at this point, and indicate where ESP and EBP are currently pointing. (On exit, proc3 and proc2 both return a value which is a doubleword.)

10. Show (in hex) the machine code for the following instructions.
    1. *(6 points)* sub EDX, EBX

    2. *(14 points)* mov EAX, dword ptr 8[ESI][EBX]

11. Show the assembly-language instruction that generated the following machine code.
    1. *(12 points)* 81 3A 0C 51 4D 38 20h

End of Exam