Name:_____

Question	Points
I.	/34
II.	/30
III.	/36
TOTAL:	/100

Instructions:

1. This is a closed-book, closed-notes exam.
2. You have 75 minutes for the exam.
3. Calculators are not allowed.

I. Multiple Choice (2 points each)

For each question in this section, circle **1** answer. Choose the best answer.

1. In an n-bit one's complement representation,
   (a) $2^n - 1$ integers can be represented.
   (b) an end-around carry is required when performing addition.
   (c) negation is performed by complementing every bit.
   (d) all of the above are true.

2. An overflow has occurred when adding two's complement numbers, if
   (a) there is a carry out of the most-significant bit (MSB).
   (b) there is a carry into the MSB.
   (c) the carry into the MSB differs from the carry out of the MSB.
   (d) the carry into the MSB equals the carry out of the MSB.

3. When 2 two's complement numbers are added, a carry out of the most-significant bit indicates
   (a) an overflow has occurred.
   (b) the result is negative.
   (c) the result is correct as long as the addends are negative.
   (d) none of the above.

4. The IEEE floating point representation uses an excess 127 representation for the exponent and a signed magnitude representation for the mantissa because
   (a) it is the easiest format for addition.
   (b) the size of two numbers can be compared directly.
   (c) multiplication and division are possible with this format.
   (d) this representation requires fewer bits than the alternatives.

5. In the process of normalizing a floating point number, a mantissa with leading zeroes is shifted left and the exponent is
   (a) decreased by the number of positions the mantissa is shifted.
   (b) increased by the number of positions the mantissa is shifted.
   (c) divided by 2.
   (d) unchanged.

6. A CPU designed using a RISC architecture allows

   (a) assembly language programs to be written more easily.
   (b) programs to be written with fewer instructions.
   (c) the processor to run at a faster clock speed.
   (d) all of the above.

7. On the Intel Pentium CPU, the ModRM byte

   (a) designates the registers to be used by an instruction.
   (b) specifies the memory location to be referenced.
   (c) permits indirect addressing.
   (d) changes the instruction to 16-bit mode.

8. Executing a CALL instruction on the Intel Pentium CPU causes changes in

   (a) the ESP register.
   (b) the EBP register.
   (c) the ESP and EBP registers.
   (d) the ESP, EBP and EAX registers.

9. Which of the following is not a legal instruction for the Intel Pentium CPU?

   (a) `add eax, ebx`
   (b) `add [eax], [ebx]`
   (c) `add [eax], ebx`
   (d) `add eax, [ebx]`

10. The difference between a short jump and a near jump is:

    (a) NASM does not handle conditional near jumps.
    (b) a short jump uses an 8-bit relative address and a near jump uses a 32-bit relative address
    (c) a short jump uses relative addresses and a near jump uses absolute addresses.
    (d) a short jump uses an 8-bit absolute address and a near jump uses a 32-bit absolute address.

11. What is the effect of the following instruction?

```
mov     ecx, [esi + eax]
```

(a) The content of `esi` is added to the content of `eax` and the sum is stored in `ecx`.

(b) The content of the memory location whose address is stored in `esi` is added to the content of `eax`. This sum is stored in `ecx`.

(c) The source operand is stored in memory. The address of this memory location is computed by adding the content of `esi` to the content of `eax`.

(d) The source operand is the sum of two memory locations. The first memory location's address is stored in `esi`, the second's in `eax`.

12. The difference between a signed conditional jump (e.g., `jge`) and an unsigned conditional jump (e.g., `jae`) is:

(a) A signed jump must be preceded by a signed comparison and an unsigned jump by an unsigned comparison.

(b) An unsigned jump can only jump to locations with higher memory addresses, but a signed jump can jump to higher or lower addresses.

(c) Nothing, `jge` and `jae` are mnemonics for the same machine instruction.

(d) none of the above.

13. The PDP-8 was

(a) the fastest computer available when it was first sold.

(b) the first widely sold minicomputer.

(c) a high-performance RISC machine.

(d) an early 8-bit computer.

14. The PDP-8 instructions referenced addresses throughout the entire memory

(a) using index registers.

(b) directly in the instruction's address field.

(c) using relative addressing.

(d) none of the above.

15. Consider a label `foo` in the `.data` section of an assembly language program. If the program is transformed into an executable program with the tools used for the projects in this class, a virtual memory address is assigned to `foo`

    (a) by the assembler.

    (b) by the linking-loader during the linking phase.

    (c) by the linking-loader during the loading phase.

    (d) none of the above.

16. The Linux operating system has a virtual memory system that

    (a) uses segment registers to achieve memory protection.

    (b) limits each process to 64 megabytes of physical memory in 32-bit mode.

    (c) provides each process with 4 gigabytes of physical memory.

    (d) assigns every process its own set of page tables.

17. A two-pass assembler takes two passes to

    (a) determine the number of bytes in each instruction.

    (b) resolve external references.

    (c) resolve forward references.

    (d) all of the above

II. Short Answers (3 points each)

For each question in this section, *show all of your work.* Clearly indicate your final answer.

1. Describe the fetch-execute cycle.

2. Give the binary representation of -15 in each of the following data representations
   (a) 8 bit one's complement

   (b) 8 bit signed magnitude

   (c) 8 bit excess 127

3. What decimal number is represented by the following 8 bit two's complement integers?
   (a) 00010110

   (b) 11110101

   (c) 11110001

4. Add the hexadecimal numbers 7C9 and 77.

5. A floating point representation uses a sign bit, a 4-bit exponent in excess 7 notation, and a 5-bit fractional mantissa with no hidden bit.

   (a) (2 points) What number is represented by the binary sequence: 0 1001 11000

   (b) (1 point) Is the binary sequence in part a) in normalized form?

6. Four contiguous bytes in memory contain the hexadecimal values 12, 34, 56 and 78 in the order of increasing address. If these bytes are used to store a 32-bit integer in little-endian format, what is the integer? (Give your answer in hexadecimal.)

7. What is 24_{10} in base 7 ?

8. Multiply the following 4 bit two's complement numbers: 1110×1101. What is the result in decimal?

9. Give one reason for a CPU to have an interrupt mechanism.

10. Consider the virtual memory system used by Linux running on an Intel Pentium CPU.

    (a) (2 points) Name two advantages for having virtual memory.

    (b) (1 point) Name one disadvantage for having virtual memory.

III. Short Programs (12 points each)

1. Consider the following code fragment. What are the contents of the `eax`, `ebx` and `ecx` registers after the code fragment has been executed? Show your work.

```
        section .data
x:      dd  32
y:      dd  17


        section .text
        mov     eax, x
        mov     ebx, [eax]
        mov     ecx, [eax + 4]
        add     ebx, [x]
        add     ecx, ebx
        mov     eax, [eax]
```

2. Consider a C function `prime` with the following function prototype:

```
int prime(int n, int m) ;
```

   This function returns the first prime number between `n` and `m`, inclusive. Write an assembly language program which calls the function `prime` to find the first prime number between 1500 and 3000 and stores this number in the `ecx` register. Recall that 4-byte values returned from a C function are stored in the `eax` register.

3. Consider the following C function prototype:

```
void larger(int *result_ptr, int *ptr1, int *ptr2) ;
```

A program that implements this function should examine the `int` values stored at the locations designated by `ptr1` and `ptr2`, then store the larger of these two `int` values at the location specified by `result_ptr`. Write an assembly language program that implements the function `larger`. Your program must follow the parameter passing convention for C functions as described for Project 2.

This page intentionally left blank.