Midterm Spring 2000 Prof. Wanjiun Liao

1. (10%) Please compress the following message using LZ77:
   ββαβββααβαβααβαβααβαβααα

2. Using the machine language of Appendix, write a program that
   (a) (10%) copies the middle four bits from memory cell A1 in the most significant four bits of memory cells A2, while placing 1s in the least significant four bits of the cell at location A2.
   (b) (15%) Suppose the machine described in Appendix communicates with a printer using the technique of memory mapped I/O. Suppose also that address FF is used to send characters to the printer, and address FE is used to receive information about the printer's status. In particular, suppose the least significant bit at the address FE indicates whether the printer is ready to receive another character (with a 0 indicating "not ready" and a 1 indicating "ready"). Starting at address 00, write a machine language routine that waits until the printer is ready for another character and then send the character represented by the bit pattern in register 5 to the printer.

3. (15%) Suppose each non-shareable resource in a computer system is classified as a level 1, level 2, or level 3 resource. Moreover, suppose each process in the system is required to request the resources it needs according to this classification. That is, it must request all the required level 1 resources at once before requesting any level 2 resources. Once it receives the level 1 resources, it can request all the required level 2 resources, and so on. Can deadlock occur in such a system? Why or why not?

4. (20%, 5 points each) Terminology
   (a) booting
   (b) DMA
   (c) lossy compression
   (d) process

5. (10%) What bit pattern represents the sum of 01101010 and 10001001 if the patterns represent values stored in two's complement notation? What if the patterns represent values stored in the floating-point format shown as follows.

   -- -- -- -- -- -- -- --: S(1) E(3) M (4)

6. (10%) A 2-hour, black-and-white video program with the frame size of (row: 480 pixels and column: 640 pixels) and a frame rate of 30 fps (frames per second) is stored in single-side floppy disks. How many floppy disks with 80 tracks per side, 16 sectors per track and 512 bytes per sector at a spinning rate of 300 revolutions per minutes do you need to store the entire program? Also, how much time do you need to spend to upload the program using your 33.6Kbps modem to the class's ftp server?

7. (10%) Each of two robot arms is programmed to lift assemblies from a conveyor belt, test them for tolerances, and place them in one of two bins depending on the results of the test. The assemblies arrive one at a time wit a sufficient interval between them. To keep both arms from trying to grab the same assembly, the computers controlling the arms share a common memory cell. If an arm is available as an assembly approaches, its controlling computer reads the value of the common cell. If the value is nonzero, the arm lets the assembly pass. Otherwise, the controlling computer places a nonzero value in the memory cell, directs the arm to pick up the assembly, and places the value 0 back into the memory cell. What sequence of events could lead to a tug-of-war between the two arms?

APPENDIX A

Op-code	*Operand*	*Description*
1	RXY	LOAD the register R with the bit pattern found in the memory cell whose address is XY. *Example:* 14A3 would cause the contents of the memory cell located at address A3 to be placed in register 4.
2	RXY	LOAD the register R with the bit pattern XY. *Example:* 20A3 would cause the value A3 to be placed in register 0.
3	RXY	STORE the bit pattern found in register R in the memory cell whose address is XY. *Example:* 35B1 would cause the contents of register 5 to be placed in the memory cell whose address is B1.
4	0RS	MOVE the bit pattern found in register R to register S. *Example:* 40A4 would cause the contents of register A to be copied into register 4.
5	RST	ADD the bit patterns in registers S and T as though they were two's complement representations and leave the result in register R. *Example:* 5726 would cause the binary values in registers 2 and 6 to be added and the sum placed in register 7.
6	RST	ADD the bit patterns in registers S and T as though they represented values in floating-point notation and leave the floating-point result in register R. *Example:* 634E would cause the values in registers 4 and E to be added as floating-point values and the result to be placed in register 3.
7	RST	OR the bit patterns in registers S and T and place the result in register R. *Example:* 7CB4 would cause the result of ORing the contents of registers B and 4 to be placed in register C.
8	RST	AND the bit patterns in register S and T and place the result in register R. *Example:* 8045 would cause the result of ANDing the contents of registers 4 and 5 to be placed in register 0.
9	RST	EXCLUSIVE OR the bit patterns in registers S and T and place the result in register R. *Example:* 95F3 would cause the result of EXCLUSIVE ORing the contents of registers F and 3 to be placed in register 5.
A	R0X	ROTATE the bit pattern in register R one bit to the right X times. Each time place the bit that started at the low-order end at the high-order end. *Example:* A403 would cause the contents of register 4 to be rotated 3 bits to the right in a circular fashion.
B	RXY	JUMP to the instruction located in the memory cell at address XY if the bit pattern in register R is equal to the bit pattern in register number 0. Otherwise, continue with the normal sequence of execution. *Example:* B43C would first compare the contents of register 4 with the contents of register 0. If the two were equal, the execution sequence would be altered so that the next instruction executed would be the one located at memory address 3C. Otherwise, program execution would continue in its normal sequence.
C	000	HALT execution. *Example:* C000 would cause program execution to stop.