Midterm   MARCH 8, 1995.

There are four problems for a total of 100 points. Show all your work, partial credit will be awarded. When there is not enough room on the test page itself, write in the provided blue books and write and sign your name on each one. No notes, no collaboration.

Name: ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Problem	Credit
1	
2	
3	
4	
Total	

1. Explain the operation of the following C program. Give its output.

```
#include<stdio.h>
int main() {
   char * s = "I'm melting!" ;
   while (*s) printf("%s\n", s++ ) ;
}
```

2. The Unix inode has 10 direct points, 1 single indirect pointer, 1 double indirect pointer and 1 triple indirect pointer. Using *only the direct, single and double indirect pointers*, what is the largest file possible in Unix?

   Give the file size in bytes to 1 significant figure. You are not allowed and there is no need to use a calculator. Assume 1K byte blocks on the disk. As you are the system designer, you decide how many pointers should fit in a block. Justify this number.

3. This problem is about P and V semaphores. At Blue Sky Foods, you take a numbered ticket upon entering and are served when your number is called. This is a method of resolution for a common resource — at first glance the common resource is the servers, but more precisely, the focus of contention is the box which distributes numbered tickets.

Sometimes there is a critical race, and two people try to grab a ticket at the same time.

Use P and V semaphores to resolve the critical race. Phrase your answer as a collection of programs in a infinite loops, continually contending to have the next numbered ticket in a sequence of tickets.

4. Running at 10 megabits per second, how long does it take to *dump* (see glossary below) a 1 Gigabyte disk? Does it make any difference if the disk has 1 K sectors, 16 sectors per track and rotates at 90 revolutions per second? How long does the dump take under these assumptions? All answers to 1 significant figure, no calculators needed or allowed.

Glossary: `dump`: To read each byte of a disk in the order the byte appears on the disk, often for the purpose of making a verbatim copy of the disk for backup.