

COP4020 Homework (due date: September 16, 2002)

1. Answer Exercise 1.2 in Section 1.9 on page 27 of the textbook.
2. Answer Exercise 1.6 in Section 1.9 on pages 27 and 28 of the textbook. In addition, logon to 'program' (type `ssh program`) and type in the following program:

```
/* myprogram.c */
#include <stdio.h>
int n;
main()
{ n = 5; printf("%d\n", f(n)); }
f(int k)
{ if (n) if (k = 0) return 1; else return k*f(k-1); }
```

and compile with 'gcc myprogram.c'. What output messages do you get? Do you spot any problems with the code? Does it run? Now compile this program from the command prompt with 'gcc -Wall myprogram.c'. What output do you get this time? Next run 'lint' on this program with 'lint myprogram.c' (make sure that your path environment variable includes /usr/ucb which is done by typing 'set path=(\$path /usr/ucb)' at the command prompt). What output do you get? What do you think about these results? Does it improve the way you would write C (or C++) code?.

3. Answer Exercise 1.11 in Section 1.9 on page 28 of the textbook.
4. Answer Exercise 1.5 in Section 1.9 on page 27 of the textbook. Logon to 'program' (type `ssh program`) and type in the following makefile (save as 'Makefile):

```
myprogram:    myprogram.o
              ld -o myprogram myprogram.o -lc
myprogram.o: myprogram.c
              gcc -c myprogram.c
```

where the indents are TABs. It is assumed that you have kept the 'myprogram.c' code from exercise #2 above. Place both 'Makefile' and 'myprogram.c' in the same directory. Next, from the command prompt type 'make'. The make program reads the 'Makefile' and compiles your program. Type 'make' again. What happens? Now type 'gcc -c myprogram.c' and 'make'. What happens? Type 'man ld' to find more about the 'ld' command. What does it do?