1.
```
# Walter Rasmussen - Spring 2018
TARGET=boat
CC=gcc
OBJS=titanic.o sink.o gone.o
CFLAGS=-g -Wall
LIBS=

${TARGET}: ${OBJS}
        ${CC} -o ${TARGET} ${OBJS} ${LIBS}

clean:
        rm -f ${OBJS} ${TARGET}
```

2.
```python
#!/usr/bin/python
# Walter Rasmussen - Spring 2018
# Prints list of unique user shells found in /etc/passwd

try: # Checks if file can be opened(permissions, etc.).
        rfile = open( "/etc/passwd", "r") # Opens file.
except:
        print("Could not open file.")
        quit()

used = [] # Empty list; Stores shell names that have already been printed.

for line in rfile: # Reads file line by line.
        try:
                col = line.split(":") # Splits line by colon delimiter.
                if col[6] in used: # Checks the 7th column for new shell name.
                        pass
                else:
                        print(col[6])
                        used.append(col[6]) # Adds shell name to list
        except:
                pass

rfile.close() # Closes file.
```

3. r"([0-1]?[0-9]|2[0-4]):([0-5][0-9]|60):([0-5][0-9]|60)"

4.
```c
// Walter Rasmussen - Spring 2018
// Creates a child, whom waits, then sends SIGHUP

#include <sys/types.h>
#include <unistd.h>
#include <signal.h>

int main(int argc, char *argv[])
{
        pid_t pid; // Stores pid of child.

        signal(SIGHUP, SIG_DFL); // Uses default handling for SIGHUP.
        pid = fork(); // Creates child.

        if (pid > 0) { // If parent.
                kill(pid,SIGHUP); // Sends SIGHUP, then exits.
        } else if (pid == 0) { // If child.
                while (1) sleep(1); // Sleeps in 1 second intervals.
        } else { // If fork() fails, exit.
                printf("fork() failed. \n");
```

```
                return 1;
        }
        return 0;
}
```

5.
       a.   `yy`
               `G`
               `p`

       b.   `sed -n '432,524p' /usr/local/xyzzy/theonering/pat`

       c.   `ln -s /var/lib/cache/storm four*two`

       d.   `chmod -R go+r ./`
               `chmod -R go+x ./`

       e.   `ps -N -d | wc`

       f.   `apt-cache search atari 2600`
               `sudo apt-get update`
               `sudo apt-get install stella`

6.   `sudo apt-get update`
       `sudo apt-get install enscript`
       `sudo apt-get install cups-pdf`
       `enscript -PPDF --header='ECE 331|Exam 01|Walter Rasmussen' ans.txt`