Homework 2

Due Date: February 6, 1998 at 11:59PM **Points**: 150
Late Date: February 9, 1998 at 11:59PM (20% penalty)

1. (*10 points*) Consider the following 4 ciphertexts: XXXXX, VWXYZ, RETIC, JZQAT. Which of these ciphertexts could result from enciphering five letter words of English, using:
   a. Any transposition cipher (in which letters are mixed but not changed)?
   b. A substitution cipher, where each letter is replaced with some other letter, but the letters are not necessarily shifted as in the Cæsar cipher (thus A could be replaced with K, B with W, *etc.*)?
2. (*20 points*) 4.4 BSD UNIX has extended the password hashing function by allowing a 4 character (24 bit) salt. DEC Ultrix has extended the password hashing function by allowing passwords of up to 16 characters; the algorithm hashes the first set of 8 characters using the conventional algorithm, hashes the second set of 8 characters using the conventional algorithm with the same salt, and concatenates the result. The salt used is the 2 character (12 bit) salt. When does the first method require more trials to guess a particular password? When does the second?
3. (*20 points*) Consider the RSA encryption scheme with public key $n = 35$ and $e = 7$. Encipher the plaintext $M = 10$. Break the cipher by finding p, q, and d.
4. (*100 points*) This problem asks you to explore file permissions and types in the UNIX operating system. You are to write a command called *access* that takes as arguments a user ID and a path name. It determines how the named user can access the object with the given path name. For example,

   ```
   access bishop /home/bishop/xyz
   ```

   prints

   ```
   bishop can read, write /home/bishop/xyz
   ```

   but

   ```
   access bishop /home/cs153r/handin/hw1/knowles
   ```

   prints

   ```
   bishop cannot access /home/cs153r/handin/hw1/knowles
   ```

   If the option `-l` is given, the same information is to be printed for each element of the path:

   ```
   access -l bishop /home/cs153/public_html/homepage.html
   ```

   prints

   ```
   bishop can list, search /
   bishop can list, search /home
   bishop can search /home/cs153
   bishop can list, search /home/cs153/public_html
   bishop can read /home/cs153/public_html/homepage.html
   ```

   What follows are some suggestions that may help you.
   i. Read, write, and execute permission on directories are really list, create/delete file, and search permission. Without search permission, a process cannot access any files in the directory (even if read and create/delete files permissions are given).
   ii. For symbolic links, list the permissions on the symbolic link, and on the next line the permissions of the referent. Use *lstat*(2) and *stat*(2) to go this.
   iii. If the named user cannot access a component of a path, it cannot access any of the following components.