Name: _____

This homework is substantially shorter and simpler than those that will follow, since it is based on only the very little material covered so far.

You should submit 1. a hard-copy of pages 1–6 of this assignment with your answers filled in, and 2. an electronic package that contains the source files for your work on the programming questions, by following the submission procedure described in class and on the class newsgroup. You are welcome to use any inanimate resources (e.g., books, Web sites, publicly available code) to help you with your work. However, *all such help must be clearly noted* in your submissions. Further, no matter what you use, *you must be able to explain, in detail, how it works.* (You may be called upon to explain your homework in person.) Refer to the class policy for details, and ask for clarifications if you are unsure if something is allowed.

1. (1 pt.) Write your name in the space provided above.

2. (1 pt.) Package and submit your solutions to the programming questions via `http://cs.umaine.edu/~chaw/u/`. *After* submitting your work, fill in the following:

   File name: _____
   Size, in bytes: _____
   MD5 checksum: _____
   Timestamp: _____

3. (9 pts.) Given the sets $A = \{1, 2, 42\}$, $B = \{24, 7, 1, 2\}$, provide a brief description (prose) *and* list the elements (explicitly) of:

   (a) $A \times B$.

   (b) $\{a \times b \mid a \in A,\ b \in B\}$.

   (c) $\mathcal{P}(A \setminus B) \times B$.

4. (9 pts.) Provide formal definitions of the following graphs:

   (a) An *n-cycle*, i.e., a graph with n vertices (n being a parameter) and edges forming a single cycle that connects all vertices.

   (b) A *complete* graph with n vertices: there is an edge for every (unordered) pair of vertices.

   (c) An *n-star*: edges connecting a single vertex (the *center*) with every other vertex.

5. (10 pts.) For each of the graphs of Question 4:

   (a) Depict illustrative examples for small n (say, 5 or 10).

   (b) Verify that your earlier definition is consistent with the informal description.

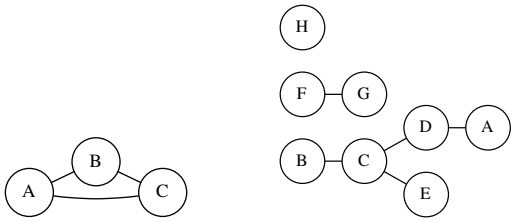
[additional space for answering the earlier question]

6. (10 pts.) Solve Problem 0.11 (p. 27) from the textbook.

7. (10 pts.) Solve Problem. 0.13 from the textbook.

8. (150 pts.) Consider Problem 0.14 (and its solution) from the textbook. Write a program that reads in a graph and outputs the clique from *pile A* and the independent set from *pile B*. Your program should read from standard input and write to standard output.

   The input is a parenthesized list of elements. Each element is a parenthesized identifier or parenthesized pair of identifiers. In the former case, it represents a vertex with the given identifier. In the latter case, it represents an edge between vertices with the given identifiers. There are no ordering constraints, so a vertex may be presented after an edge that uses it. Further, vertices that are mentioned in edges are implicitly assumed to exist, even if they are not explicitly provided in the input. You may assume that the identifiers follow the rules for Java identifiers.



   For example, the graph depicted on the left above may be presented as input `((a b) (b c) (b) (c a) (a))` The `(a)` and `(b)` are redundant and may (or may not) be skipped in such input. The graph on the right may be provided as: `((a) (b c) (d a) (c d) (c e) (f g) (h))`. The input may contain arbitrary whitespace (including none, except when needed to separate identifiers). So the second example may also be provided as `((a)(b c)(d a)(c d)(c e)(f g)(h))` or with several blank lines inserted between the first `b` and `c` occurrences.

   The output should be a the clique followed by the independent set, both encoded as above. However, the output encoding should use vertex specifiers of the form `(x)` only when strictly needed. Further, each graph's representation should be lexicographically sorted (with identifiers compared lexicographically as well).