Homework Assignment 1.

Due: September 15, 2004 at class.

Problem 1:

In the *Non-Uniform* Consensus problem, the agreement requirement is replaced with the following weaker property:

Non-Uniform Agreement: No two *correct* processes decide on different values.

Show how to modify the t-resilient, $t < n$, Consensus algorithm for stopping failures presented in class to solve the non-uniform Consensus problem. Your algorithm should be both t-resilient and *early-deciding* in the following sense: every correct process decides in at most $f + 1$ rounds, where f, $1 \leq f \leq t$ is the actual number of failures that occur in the run.

Carefully prove the correctness and round complexity guarantees of your algorithm.

For your reference, the Consensus algorithm presented in class is as follows:

Process i:

```
W := {x_i};
For each round r, 1 ≤ r ≤ t + 1 do:
Send W to all processes;
For each process j, 1 ≤ j ≤ n do:
    if process j sent a message X_j then
        W := W ∪ X_j;
if (r = t + 1) then
    if (|W| = 1) then
        decide x, where W = {x};
    else
        decide a default value x_0;
```

Problem 2:

Design a t-resilient algorithm that solves non-uniform Consensus for general omission failures.

Does there exist an algorithm that solves *uniform* Consensus and tolerates up to t, $t < n$, general omission failures? Give an algorithm, or argue informally why you think it does not exist.