CS 376 **Operating Systems** Test 1
Name: Maximum Points: 40

Important: This is an open book test. You can use any books, notes, or paper. *Do not log into the computer during the test.* Any calculations and rough work can be done on the back side of the test pages.

1. (a) [8 pt] Write a shellscript to accept a character from the terminal as soon as it is typed (that is, do not wait for the user to hit return).

   (b) [4 pt] I issued the following command while searching for a file on arch

   ```
   find / -name aaron.ms -print
   ```

   but I got at least three screenfuls of error messages. The actual path of the file was buried somewhere in those messages. Can I do something to get rid of the error messages which occur as I do not have read and search permissions on a number of directories.

2. [10 pt] Consider the following program in a version of C that supports `parbegin` and `parend`

```
#define N 50

int tally; /* Global shared variable */

void total ( void )
{
    int count;
    for ( count = 0; count < N; tally++, count++ );
}

main()
{
    tally = 0;
    parbegin
        total();
        total();
    parend;
    printf ( "The value of tally is %d\n", tally );
}
```

   (a) Determine the proper lower bound and upper bound on the final value of the shared variable `tally` output by this concurrent program. Assume processes can execute at any relative speed and that a value can be incremented only after it has been loaded into a register by a separate machine instruction.

(b) Suppose that an arbitrary number of these processes are permitted to execute in parallel under the assumptions of part (a). What effect will this modification have on the range of final values of `tally`?

3. [8 pt] Can we use binary semaphores to access the critical section of more than two processes? If not, explain your reasoning. If yes, explain how you propose to do it by giving a code template to solve the critical section problem.

4. [10 pt] We have seen in the lectures how to implement a monitor using semaphores. Assume that the monitor is already implemented. Write code to implement semaphore operations using the monitor. Your operations are meant for generalized semaphores (not binary semaphores).