ECE242, Fall 2002, Section 2, Quiz 2 **Tuesday October 29**

Name: **Student Number:**

1. [**2 marks:**] Choose the smallest complexity class that describes the code fragment below, in terms of the parameter `n`. Choose one of: $O(1)$, $O(\log n)$, $O(\sqrt{n})$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(n^3)$, $O(2^n)$. Explain your choice.

```
for (i = 1; i < n / i; i++)
    printf("%d\n", n / i);
```

   $i < n/i$ is equivalent to $i^2 < n$ (multiply both sides of the inequality by i. So i runs from 1 to $\lfloor \sqrt{n-1} \rfloor$, and the code fragment is $O(\sqrt{n})$.

2. [**2 marks:**] Program `prog` has its source code in `prog.c`, and it uses functions from module `mod1.o` (with source code in `mod1.c`). Module `mod1.o` uses no functions from `prog`. Declarations for `extern` functions for `mod1.o` are in `mod1.h`. Which files need `#include mod1.h`? Explain your answer.

   `prog.c` **must** include `mod1.h` in order to have access to the declarations of functions that it uses (implicit declarations are too risky to consider feasible). It's also wise for `mod1.c` to include `mod1.h`, to ensure that the implementations of its functions match the way they are declared in `mod1.h`.

3. [**2 marks:**] Under what conditions (if any) is the time efficiency of Selection Sort no worse than Merge Sort or Quick Sort? Explain.

   In a short list, say of length 2, it is no less efficient to swap two elements than to make a couple of recursive function calls.

4. [**2 marks:**] Which implementation of the stack ADT is most suitable for a "To-do" list where the most recent task is `push()`ed onto the top of the list, and you don't know in advance the maximum number of tasks there may be "To-do" at any given time? Explain your choice.

   The linked list implementation of a stack allows $O(1)$ pushing and popping to the head of the list and grows dynamically (no need to know in advance the maximum size of the stack).

[End of quiz]