Tutorial 2

- Side effects
- Reading input
- Global variables & mutation
- Stacks & Command Based Testing

Side effects

(there are 3 types of side effects in functions for now)

- print output
- read input
- mutate a global variable

// MANY SIDE EFFECTS

int y = 2;

int main(void) {
    int x = read_int();
    printf("%d", printf("%d", y));
    if ((x = y - 1)) {
        printf("%d", x = x + 2);
    }
    if (x) {
        y *= y;
        printf("%d", y);
    }
    printf("\n");
}
Side Effects: printing vs. returning

```c
int pure_functional(int n) {
    return n * n;
}
```

// effects: displays a message
```c
void just_a_side_effect(int n) {
    printf("n squared is %d\n", n * n);
    return; // (optional)
}
```

// effects: displays a message
```c
int has_both_side_effect_and_return_value(int n) {
    printf("n squared is %d\n", n * n);
    return n * n;
}
```

Read Input: Exercise

Implement the following function using recursion:

```c
// print_reverse(): reads in input from the user until
//       READ_INT_FAIL then prints the numerical
//       inputs in reverse order.
// Effects:
// * reads input
// * prints an output message
```
Stacks

There are a few basic operations that are common when using stacks:

- **Push**: adds an item to the top of the stack.
- **Pop**: removes an item from the top of the stack and returns the removed value.
- **Top**: returns the value on the top of the stack.
- **Empty**: determines if the stack is empty.

Command Based Testing

Command based testing clients are commonly used in CS 136.

Tips:

- Enter commands when prompted for input.
- Enter a proper number of parameters.
- Consult the assignment file for help with documentation.
- Use the "help" command, if available.