60% of your points come from these problems

1. Read my book’s Chapter 2. Ask if any questions (or you find typos)

2. Read my book’s Chapter 5. Summarize the proof of the equivalence of arithmetic and complete induction in your own two paragraphs.

3. Read Chapter 18

4. Read Chapter 23, Section 23.3 onwards

5. Based on the above readings (10% of the 70% for these readings), try and do the following problems. Ask for help in class next week
   
   (a) (20%) Problem 5.3
   (b) (20%) Read Section 18.3.8 and apply this polynomial time algorithm to show that \((a \lor b) \land (a \lor \lnot b) \land (a \lor b)\)
      is satisfiable.
   (c) (10%) Problem 18.3

25% of your points come from this problem: Using weakest preconditions, find two inputs that can make the Binary search loop print HL*. Apply these inputs and run the program to show that this printing happens.

```c
#include <stdio.h>
#include <string.h>
#define STRLEN 16

int main()
    char item; char str[STRLEN]; int lo=0; int hi; int mid, found=0;
    printf("Please give char to be searched: \n"); scanf("%c", &item);
    printf("Please give string within which to search: \n"); scanf("%s", str);
    hi=strlen(str)-1; // strlen ignores null at the end of string
    // hi points to last char in a 0 based array
    while(!found && lo <= hi) {
        mid = (lo+hi)/2;
        if (item == str[mid]) { printf("*" ); found = 1; }
        else if (item < str[mid]) { hi = mid-1; printf("L"); }
        else { lo = mid+1; printf("H"); }
        }
    if (found) printf("\nitem %c found at posn %d\n", item, mid);
    else printf("\nitem %c not in str %s\n", item, str);
    return found;
```
6. (5%) Show why Peterson’s protocol can fail on a TSO memory model. Explain using an example.

7. (10%) Show that the loop invariant guessed for Tournament min is one (a loop invariant) by walking back through the $A[2.y-1] \leq A[2.y]$ path.