Homework 5 Group 10
Nathan Daily daily043
Wyatt Kormick kormi001
Ross Faber faber047

Problem 1
$H = high$
$\land\ L = low$
$\land\ M = L+H/2$
$\land$ Loop invariant: $\forall\ i, j, 0 \le i < j < size : dictKeys[i] \le dictKeys[j]$
$\land\ \forall\ k, 0 \le k < size : dictKeys[k] = key \Rightarrow L \le key < H$
$\land$ Line 27: $\exists\ p\ dictValues[p] = key \land strcmp(dictKeys[p], key) = 0$
Therefore we can conclude that if there is a key within dictKeys that matches the input key, it will always return that keys corresponding value.

Problem 2
Precondition: $low = L\ \ high = H \land mid = M \land comparison = C$
Postcondition 1: $low = L \land high = H \land mid = M \land comparison = C \land C \ge 0$
Postcondition 2: $low = L \land high = H \land mid = M \land comparison = C \land C \ge 0 \land C \le 0$
Postcondition 3: $low = L \land high = H \land mid = M \land comparison = C \land C \ge 0 \land C \le 0 \land C \neq 0$
There is no assignment to C that makes this true
Comparable values must always be one of less than, equal to, or greater than each other

Problem 3
Precondition: $prevIndex = M \land prevprevIndex = L \land currentIndex = C$
$\land\ \forall\ i, 0 \le L < M < C < n$
Postconditon: Number of times $(a[C]<a[L]\ge a[M] \lor a[C]>a[L]\le a[M])$ occurs in a[].

Problem 4
If there are more shared variables than locks, it is impossible to determine which variable and thread has each lock at any point in time without contextual analysis.

Problem 5
Having the actual values in the states would be more precise, but take up a lot of space since each state would be represented in the set of states. Since you have each possible state you would be able to get your answer quickly. If you just have the variables and not the concrete values, then you are going to save a lot of space, but you have to formulate the representation of the state that you need.

Problem 6
(1)
Pre-condition: top = T ∧ psums[ ] = P ∧ counts[ ] = C ∧ alloc_count = A ∧ cur_count = CC
Post-condition: ∀ i, C[i] = 0 ∧ (the number of times each client appears in the output is equal to its weight) ∧ (they are in weighted round robin ordering.)

(2)
In hw5p6.c
And test_results#.txt

Test1 : Input : 5, 1 2 1 3 4 Expected Output : 5 4 3 2 5 4 1 5 2 4 5
Test2 : Input : 3, 1 1 1 Expected output : 3 2 1
Test3 : Input : 0, Expected output :
Test4 : Input: 8, 1 2 1 2 1 2 1 2 Expected Output : 8 7 6 5 4 3 2 1 8 6 4 2
Test5 : Input : -1, 1 : Expected Output: Segmentation Fault

(3) C[i] -> (P[i] = C[i] + C[i-1] + … + C[0])

(4) ∀ x in X, P[x] = C[x] + C[x-1] + … + C[0]