Loop Invariants

4. Give a useful loop invariant of each loop. Indicate the places in the code where the invariant holds.

   (a) For this problem, assume that $A[0] \ldots A[n-1]$ is an array of integers, where n is a positive integer. Loop Invariant: $A[i] = \min\{A[k] : 0 \leq k < j\}$

   ```
   int i = 0;
   int j = 0;
                  // loop invariant holds here
   while(j < n-1){
                  // loop invariant holds here
     if(A[j] < A[i]) i = j;
     j++;
                  // loop invariant holds here
     }
                  // loop invariant holds here
   ```

   (b) For this problem, assume that $A[0] \ldots A[n-1]$ is a sorted array of integers, where n is a positive integer, and that B is an integer. Loop Invariant: $A[k] < B$ for all $0 \leq k <$ `lo`, and $A[k] \geq B$ for all `hi` $\leq k < n$.

   ```
   int lo = 0;
   int hi = n;
                  // loop invariant holds here
   while(lo < hi){
                  // loop invariant holds here
     int mid = (lo+hi)/2; // truncated division, as in C++
     if(A[mid] < B) lo = mid+1;
     else hi = mid;
                  // loop invariant holds here
     }
                  // loop invariant holds here
   if (           ) cout << "Yes" << endl;  // I need to insert a condition here!
   else cout << "No" << endl;
   ```

   It should be clear to you what the purpose of this code is. What do you think the condition of the if statement should be?

   The purpose of this code is to decide whether there is some entry of A which is equal to B. The condition is: `A[hi] == B`. You could write `A[lo] == B`, since `lo = hi`.

(c) For this problem, assume that $X[0] \dots X[n-1]$ is an array of real numbers, where n is a positive integer. Loop Invariant: `sumPositive` is the sum of the all the positive entries in the array of index less than `i`. That is, all the positive entries of the prefix $X[0] \dots X[i-1]$.

```
real sumPositive = 0;
int i = 0;
             // loop invariant holds here
while (i < n){
             // loop invariant holds here
   if (X[i] > 0)
    sumPositive += X[i];
             // loop invariant DOES NOT HOLD here
   i++;
             // loop invariant holds here
  }
             // loop invariant holds here
cout << sumPositive << endl;
```