CIS 575. Introduction to Algorithm Analysis
Assignment #1, Spring 2019
Due Thursday, January 31, 11:59pm

You may if you so prefer work in groups of two in which case each name should be listed on your answer but only one of you should submit.

Problem. We want to write an algorithm that expects an array $A[1..n]$ and a number x, and returns a number q, and that implements the specification

precondition the array $A[1..n]$ is non-decreasing, and we know that x occurs in $A[1..n]$

postcondition $q \in 1..n$ and $A[q] = x$.

For example, if $A[1..6]$ is given by

1	2	3	4	5	6
12	15	17	17	26	30

then

- if $x = 26$ then $q = 5$ must be returned
- if $x = 17$ then either $q = 3$ or $q = 4$ must be returned (the specification is non-deterministic)
- if $x = 10$ then (as the precondition is not fulfilled) the algorithm could behave in any way; it may return 1, never terminate, or crash...

To implement this specification we shall use the binary search principle, and develop an iterative algorithm of the form

```
Find(x, A, n)
   P
   q ← (lo + hi) div 2
   while A[q] ≠ x
      if A[q] < x
         T
      else
         E
      q ← (lo + hi) div 2
   return q
```

which uses variables lo, hi, and q, and where the loop invariant Φ has various parts:

(0) A is non-decreasing
(1) $1 \leq lo \leq hi \leq n$
(2) $lo \leq q \leq hi$
(3) x occurs in $A[lo..hi]$

In this exercise, we shall develop P so as to complete the preamble, and develop T and E so as to complete the loop body.

1. Specification (6p) Translate the precondition into a formula in predicate logic (which must contain quantifiers, universal $\forall$ and/or existential $\exists$).

2. Preamble (8p) We shall find a suitable P.

1. (4p) Give an example that shows that letting P be $lo \leftarrow 1;\ hi \leftarrow n-1$ may *not* establish Φ.
2. (4p) Now define P (as a sequence of assignments) so that Φ will *always* be established (you don't need to argue for that).

3. Loop body (18p) We shall find suitable T and E.

1. (5p) First consider the case where T is given by $hi \leftarrow q$ and E is given by $lo \leftarrow q$.

   It is quite obvious that this choice will maintain parts (0)–(2) of Φ. But give an example that shows that this may *not* maintain part (3) of Φ.
2. Next consider the case where T is given by $lo \leftarrow q$ and E is given by $hi \leftarrow q$.
   (a) (4p) Argue that this choice will maintain part (3) of Φ.
   (b) (5p) Give an example that shows that with this choice, the loop may *not* terminate.
3. (4p) Write T and E such that Φ is maintained *and* termination is ensured (you do not need to argue for these properties).

4. Recursion (8p) Translate your completed iterative algorithm into a recursive algorithm that is equivalent, in that it always examines the *same* array elements.

Specifically, you must write a function FIND that calls itself (but contains no **while** loop) and which as argument takes at least lo and hi, and *perhaps* also q, but you *may* omit the "global" A and x.

Remember also to state how to call FIND at "top-level" so as to implement the specification.