Homework Assignment #1
Due: January 16, 11:30 a.m.

Before working on this assignment, read the policy on academic honesty on the course web page. When you submit this assignment, you must also submit the signed declaration that you have read this policy.

1. Bob Fnord is the mayor of a large city that is building a subway, which will be n metres long. One day, he was looking at the subway plans in his office. The plans include a list of numbers that give the elevation (in metres above sea level) for each location along the track. Bob types those numbers into an array $A[0..n]$ so that $A[i]$ gives the elevation of the point on the track that is a horizontal distance of i metres from the southern end. (Thus, $A[0]$ is the elevation of the south end of the track and $A[n]$ is the elevation of the north end of the track.) Assume that the track rises or falls along a straight line between the elevations of $A[i-1]$ and $A[i]$ along the ith metre of the track.

Bob also has the contract from the manufacturer of the subway trains. It says that the subway trains can never travel on a track whose slope is greater than 0.1 at any point. Furthermore, it says that the trains cannot travel up a slope that is continuously greater than 0.05 for more than a (horizontal) distance of 25 metres at a time. These are the only constraints the manufacturer puts on the track.

Bob writes the following algorithm to determine whether the trains will be able to travel from the southern end of the track to the northern end. (Bob is clever enough to see that he can check the reverse direction by reversing the array and running the algorithm again.)

```
CHECK(A[0..n])
    Precondition: n ≥ 1 and A[i] is a real number for 0 ≤ i ≤ n
    Postcondition: outputs yes if a train can travel from the south end to the north end,
                   and no otherwise.
    c = 0
    d = 0
    ok = true
    for i = 1..n
        // Invariant: fill this in later
        if A[i] > A[i − 1] + 0.1 then ok = false
        else if A[i] > A[i − 1] + 0.05 then c = c + 1
        else c = 0
        end if
        d = max(d, c)
    end for
    // finish this later
end CHECK
```

As you can see, Bob did not finish writing the code. (He had to go coach a foosball game.)

(a) Write the code that should come after the loop to produce the output. Do not change the code inside the loop. The code you add should take $O(1)$ time to run.

(b) Fill in the loop invariant. The invariant you choose should be strong enough that you can complete part (c), below.

(c) Prove that your completed algorithm is correct.