Problem 3.2 : Selection Sort

B)

Loop Invariant:

Before the start of each loop, arr[minval] is always less than or equal to arr[x…..y].

Before first iteration :

Prior to the first iteration of the loop, y=x+1. So the subarray arr[x....y] is just arr[x]. Since the Selection Sort function sets the code sets minval = x, we have that minval indexes the smallest element (the only element) in subarray A[x…..y] and hence the loop invariant is true.

During iteration :

Before we get to y, we assume that minval indexes the smallest element in the subarray arr[x....y]. During iteration y we have two cases: either arr[y] < arr[minval] or arr[y] ≥ arr[minval]. In the second case, the if statement is not true and nothing is executed. But in the first case, minval indexes the smallest element of arr[x....y]. In the first case, the if statement assigns to minval index location y since it is the smallest. If minval indexes an element less than or equal to subarray arr[x…..y] and now arr[y] < arr[minval], then it must be the case that arr[y] is less than or equal to elements in subarray arr[x....y]. The code inside the if statement assigns to minval the index of this new location and hence after the inner loop iteration finishes, minval indexes the smallest element in subarray arr[x....y]. And during each iteration, since the if statement assigns y as the new position of minval after the comparison that is and since arr[y] < arr[minval], minval gets updated with each iteration by the if statement as being the index of the smallest element in subarray arr[x….y] which implies the loop invariant is true as arr[minval] is still less than or equal to arr[x…..y] for each iteration.

Termination of the loop :

At termination of the inner loop, minval indexes an element less than or equal to all elements in subarray A[x…..m] since y = m +1 upon termination which shows that the loop invariant also holds TRUE here. This finds the smallest element in this subarray and is used in the outer loop as a means to move that next smallest element into the correct location.

\*\*\*\* As a result of this analysis, the Selection Sort algorithm is CORRECT since the loop invariant holds TRUE for the entire lifetime of the algorithm. \*\*\*\*