Question 5

This problem was very interesting. At first I had some difficulty understanding it and concept of the problem itself. I had to look up some videos which helped me understand “The longest increasing subsequence” term. As far as for the program, I used the same Ar\_Stack that my classmate constructed for one of the other homework problems. The Ar\_Stack privately contains a pointer to an array in the heap as well as a integer that stores the position that is on top of the stack and an integer that store the stacks size. The constructor passes in the size as its only parameter and creates the ar stack. The class contains all the basic functions of a stack such as push, pop, printStack, top, is\_empty, is\_full, and position. My classmate has explained these functions in his project question when creating the array based stack. This programs core is entirely in the “LISLength”. The fuction is a member function of the array stack and it returns the length of the longest increasing subsequence of a given sequence. The program begins by creating a dummy stack of the same size of the stack we are checking. We will push 1 onto each position in the dummy stack because every position has a LIS of at least 1. There are two for loops, one will go along the track until the number ahead of it is smaller than the current index. The second for loop will begin and do the same thing but it will compare itself to the i’th position. If the i’ith position is bigger than the j’th position then that means the j’th positon can fall under the LIS for i. We keep track of the LIS of each position with the dummy stack that is created. We only update the LIS of a position when the max of the two LIS’s is bigger. That is why I use the built in C++ max\_element function that returns the longest increasing subsequence. As the dummy stack holds the LIS for each position, it saves the program a lot of time by not having to do overlapping problems over and over again. The biggest difficulty in this problem was implementing the algorithm, there are many ways in doing it but it can get very complicated. The way I went about it was fairly complicated but still understandable.