**Report Project 2.**

There are two java files that I am attaching to the zip file containing this report.

1. **AXP133230\_Program1.java**.

This program does permutations or combinations based on the verbose input, which is the 3rd argument passed to the program.

The 3 arguments are n, k, v, which is nothing but selecting k values out of n available values.

When verbose = 0, the count of the permutations of n choose k, along with the running time is printed.

When verbose = 1, the count of combinations of n choose k, along with the running time is printed.

When verbose = 2, all permutations of n choose k is printed and then the output of verbose = 0 is printed.

When verbose = 3, all combinations of n choose k is printed and then the output of verbose = 1 is printed.

To run the program, first compile the code using

1. Javac AXP133230\_Program1.java
2. Then run using, java AXP133230\_Program1 n k v
3. **AXP133230\_Program2.java.**

This program does permutation of a sorted array which can contain duplicates and then prints the distinct permutations in a sorted fashion based on the verbose value.

This program takes 2 lines of input.

The first line of input takes 2 values, n and v. n is the length of the input array and v is the verbose value.

The second line of input takes n values which is the values of the input array.

If verbose = 0, it prints out just the number of permutations possible with the running time.

If verbose = 1, it prints out all the distinct permutations possible with the output of verbose = 0.

To run the program,

1. Compile the code using javac AXP133230\_Program2.java
2. Then run the code using java AXP133230\_Program2.java
3. Now it needs 2 lines of input.

n v

n values of the input array, with each element having at least a single space between them. Press enter.

It took me about 6-7 hours on the whole to complete the project. The second question was pretty straight forward, but the n choose k permutations was a little tricky as I had ideas for 2 or 3 algorithms but had to choose and implement what was better of those 3 algorithms.

**Arvind Krishna Parthasarathy – axp133230**