Rico Adrian

CS 361

Lab 01 Report

Program 1: percentile.cpp

Structure of program: The variables used are total number in file, an array of integers containing list of all numbers in the file except the first number. It also has some calculations of different percentiles (25^{th}, 50^{th}, and 75^{th} percentile). The other variables used are number to assign all the numbers in file and 3 percentile variables to calculate which index on the array is going to be the 25^{th}, 50^{th} and 75^{th} percentiles.

How it works: First, create an array using pointer and output the first number on data.txt into a variable. Then, create a while loop to assign all the numbers in the file into an array listOfNumbers. After that, sort all the numbers in the array. After sorting, call the function percentile to calculate the 25^{th}, 50^{th}, and 75^{th} percentile and convert them into an integer. That will round down all the calculation outputs. Lastly, there are if, else if and else statement to output the 25^{th}, 50^{th}, and 75^{th} percentile into the console. The first statement is in case the total numbers in file, that is, the first number in file data.txt ,is divisible by 4, the second case if the number is divisible by 2, the else statement is if the number is if the number is not divisible by 4 or 2. This will automatically output the index on an array without using cell function. For instance, if number is divisible by 4, then output the number with index-1 because array index starts at 0. (we do not have to round up the number if it's divisible by 4). If number is not divisible by 2 or 4, output the number with index 1 above convertedFirst, convertedSecond, and convertedThird. This is basically how I rounded up the calculation of percentiles. Inside those if statements, I also use cout to print/output the 25^{th}, 50^{th} and 75^{th} percentile.

Design choices of program: I design the code using only 2 functions. There is a function called percentile which takes percentage value as input and return the corresponding percentile. The second function is a main method which has all the calculations, if statements to output percentiles. The reason why I used the function percentile is to make it easier to calculate the percentiles since there are 3 things needed to calculate, which are the 25^{th},50^{th} and 75^{th} percentile.

Program 2: permutation.cpp

Structure of program: The variables used are n and k to input the value of n and k of permutations using command argument. N is the elements (0-9), k is the number of digit. There are also a string of numbers containing 0-9, a data structure set to output all possible permutations into the set later, stringstream to combine the strings together using str() method.

How it works: First, create an array of numbers containing numbers 0 to 9. Second, create a set of permutations. Then, use the do while loop to generate all the possible permutations using next_permutation algorithm. Inside the loop, it combines each permutations together (for instance, 0+1, 0+3) one by one using str() method. Inside the loop, I also insert all the possible combination of permutations into the set called permutation. Last, I use iterator to print all the possible permutations into the console using cout function.

Design choices of program: I design the code using only 1 function which is the main function which has all the loop statements to generate the permutations, an iterator to print all the possible permutations. The reason is because no other functions needed to design the code. The program is very short and it only needs couple of variables and loop statements since I used the next_permutation algorithm to generate the permutations.