**Data Structure\_2071035 Lee Somin**

**Technical Report – radix\_sort.cpp**

*Theorical Explanation of Functions in ‘radix\_sort.cpp’*

**#define MAX\_DATA**

This definition represents the max number of data to be generated.

**#define MAX\_BIT**

This definition represents the number of bits used to save a number.

**#define RADIX\_BIT**

This definition represents the number of bits used for each digit.

**int raidx**

The variable is initialized as the 2^(RADIX\_BIT) and it stands for the base number of radix sort.

**int digit**

The variable is initialized as MAX\_BIT/RADIX\_BIT and it stands for the max number of digits a number can contain.

**random**

Inputs: int n

Return: int (rand()\*rand())%n

This function is made for generating a randomly selected number from 0 to n by returning the remainder of (rand()\*rand()) divided by n. Since the range of function ‘rand()’ is not big enough, rand()\*rand() was put.

**CountingSort**

Inputs: int\* A, int\* B

Return: non

This function is for sorting the array ‘A’ based on the given reference ‘B’. First, it allocates memory to the histogram array ‘int \*hist’. Next, it initializes the array with 0. In following ‘for’ statement, it builds histogram by adding 1 to the number that appeared in array ‘B’. Then, the function makes the histogram as cumulative by adding a element with its preceding element. Then in the next ‘for’ loop, the index of the data in ‘B’ is saved at the index which is equal to its frequency in histogram. Then the data of index with result[t] is copied to result[t] from A. Finally, the data is copied back to A in sorted order.

**RadixSort**

Inputs: int\* A, int d

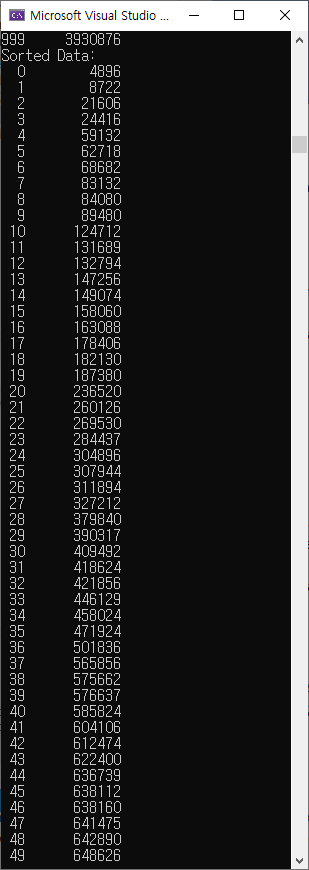
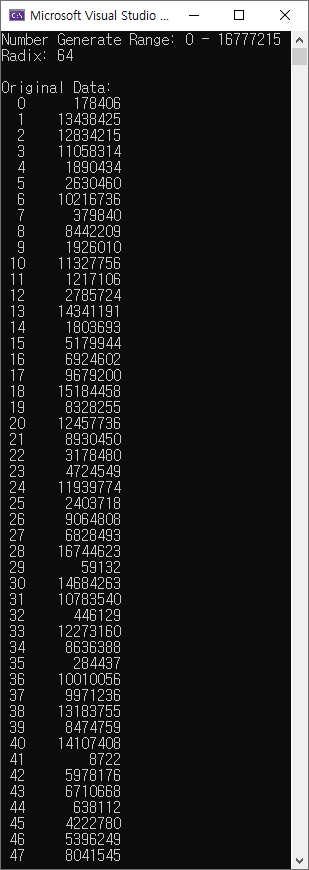
Return: non

This function is for executing the Radix Sort to the given data array A with numbers of d digit. For each digit, the function calculates the number at digit by getting the remainder of C(A divided by 2^current digit-1) divided by the value of radix. Then it calls 'CountingSort()' to sort the data based on the digit. The sorting is executed from least significant digit to most significant digit.

**main**

In main function, the random numbers are assigned to the each index of the array, 'input\_data' and the array is printed. Then, it calles the function 'RadixSort()' to sort the generated data. Finally, it prints out the sorted data.

(Result Screen Shot Continued)

**Result:** **(part of result omitted)**