**PROJECT 6 ( Do A, B, C, and D) (5+30+15+10 points)**

**Hash Function, Hash Table and Sequential Search**

**(Correction in Color Font )**

**(A) Create an Array of 4-digits student ID’s of Size 500 using the random generator function. Display the array.**

**(B) Using division method for a hash function H and a hash table of size m, a prime number, store your data of 4-digits student ID’s (n=500) to a hash table which is an array of linked list for collision resolution.**

**H (k) = k mod m**

**m : The size of hash table. A prime number larger than and not too larger than 500 / 3 and 2\*\*7 < m < 2\*\*8 and not too close to the power of 2.**

**\*\* means power of 2.**

**(1) Create 500 4-digit student ID’s using random number generator; And mapped to hash table where each cell is chained to a link list.**

**(2) Print out hash table with contents of each cell (addresses) which are head pointers pointing to a chain of linked list of IDs.**

**(3) Do 20 searches – 17 with ID found and the other three with ID not found.**

**(4) Use the counter during the searches to compute the line of statements**

**each searching goes thru which are roughly representing the computation time.**

**(5) Display ID to be searched and the result for each search with the computation time by counting the number of times the search function has gone thru.**

**(C) Use sequential search to search a student-ID.**

**For each search, print the ID to be searched and count the number of times the sequential search function has gone thru. And print the count for each search.**

**Do same as B(3), B(4) and B (5).**

**(D) Now compare the computation complexity of Sequential Search and Hashing in the tabular form or in a graphical form.**

**Your program output must show proper information to be understood well**

**by the reader/viewer. Note: Binary Search will only work on the sorted array.**