Objective: In this project we will empirically evaluate two sorting algorithms and one of which is radix sort to sort strings.

Description: Use arrays of sizes from 10,000 to 50,000 (increments of 5,000) all of them are strings of variable lengths not exceeding 10. You need to write a function that will generate random strings of length not exceeding a particular length.

For each size (say 10000) generate 10 different sets of random strings. Execute the two sorting algorithms and find the average time it takes to execute by each of the algorithms (count the executing time with QueryPerformanceCounter below). Finally, you are to draw a bar graph with the sizes and algorithms on the x-axis and average time on the y-axis using Excel. The output of your program should be unsorted and sorted set for each sorting method (print only the first 10 elements).

```
#include <iostream>
using namespace std;
#include <windows.h>

int X[] = {10000, 15000, 20000, 25000, ....};

//Write your sorting programs

void main()
{
   //Create a seed
   //A is an array of random strings of length not exceeding 10.
   //Make a copy of A into B.
   //Call your first sorting method on B
   //Print the sorted array B.
   //Make a copy of A into B.
   //Call your Radix sorting method on B.
   //Print the sorted Array B.

   delete [] A;

   for (int k =0; k < 10; k ++)
   {
      for (int q = 0; q < 10; q++)
      {
           //fill up A with x[k] random strings
           //copy contents of A into B.
           //start time
           //perform sorting 1
           //end time
           //copy contents of A into B.
           //start time
           //perform sorting 2
           //end time

           //Housekeeping to calculate the average
      }
      //find the average
      delete [] A;
      delete [] B;
   }
}
```

In order to evaluate the number of ticks you may use a routine similar to the one below.

```
#include <iostream>
using namespace std;
#include <windows.h>

void main( void )
{
        int i;
        int j;
        LARGE_INTEGER start, stop;

        for (int k = 0; k < 10; k++) {
                cout << "Enter a integer"  <<endl;
                cin >> j;

                QueryPerformanceCounter(&start);//start the tick counter

                for (i =0; i < j; i++); //Perform the operation; A dummy for loop

                QueryPerformanceCounter(&stop);//end the tick counter

                cout << stop.QuadPart - start.QuadPart << endl;
        }
        cin >> i;
}
```

You could find usage of QueryPerformanceCounter from the following link:
https://msdn.microsoft.com/en-us/library/windows/desktop/ms644904%28v=vs.85%29.aspx

You are not allowed to communicate in any form or action about this project with anyone else in the class. This is an individual project. Good Luck!