```cpp
#include <iostream>
#include <string>
#include <iomanip>
#include <fstream>

using namespace std;

const int maxnum = 10000;
double numofcompares = 0;
double numofmoves = 0;


void printsort(ofstream &outf, string &sortname, double array[]){
float increment = 1000;
int j = 1;
outf <<  sortname << " " << " " << "N = " << maxnum << endl;
for (int i =0; i < maxnum; i+=increment) {
    outf << setw(13) << fixed << setprecision(3) << array[i];
    j++;
    if (j>5) {
        outf << endl;
        j = 1; } }
outf << endl <<  fixed << setprecision(0) << "Number of Compares = " <<
numofcompares << " ";
outf <<  fixed << setprecision(0) << " Number of Moves = " << numofmoves;
outf << endl << fixed << setprecision(4) << "Relative # of Compares = " <<
numofcompares/maxnum << " ";
outf << fixed << setprecision(4) << " Relative # of Moves = " << numofmoves/maxnum
<< endl;
outf << endl << endl;
}

void swapme(double &a, double &b) {
double temp = a;
a = b;
b = temp;
numofmoves+= 3;
}

void BubSort(double array[]) {
    for (int y = 0; y < maxnum-1; y++){
        for (int b = 0; b < maxnum - 1; b++) {
            numofcompares++;
            if (array[b] > array[b+1]) {
                swapme(array[b], array[b+1]); } } }
}

void SelSort(double array[]) {
    int min, k, j;
    for (k = 0; k < maxnum-1; k++) {
        min = k;
        for (j=k+1; j < maxnum; j++){
            numofcompares++;
            if (array[j] < array[min]) { min = j; } }
```

```
52                  swapme(array[k], array[min]);}
53  }
54
55  void InsSort(double array[]) {
56  double save;
57  int k, j;
58  for (k=maxnum-2; k >=0; k--) {
59      j=k+1;
60      save = array[k];
61      numofmoves++;
62      array[maxnum+1] = save;
63      numofmoves++;
64      numofcompares++;
65      while (save > array[j]) {
66          numofcompares++;
67          array[j-1] = array[j];
68          numofmoves++;
69          j = j+1; }
70      array[j-1]=save;
71      numofmoves++; }
72  }
73
74  void QuiSort(double array[],int left, int right) {
75  int j, k;
76  numofcompares++;
77  if (left < right) {
78      j= left;
79      k= right + 1;
80      do{
81          do{numofcompares++; j++;} while((j<k) && array[j] < array[left]);
82          do{numofcompares++; k--;} while((k>=0) && array[k] > array[left]);
83          numofcompares++;
84          if (j<k) { swapme(array[j],array[k]); }
85      } while(j<=k);
86      swapme(array[left],array[k]);
87      QuiSort(array,left,k-1);
88      QuiSort(array,k+1,right); }
89  }
90
91  int main() {
92  double RandNums[maxnum+1];
93  ofstream outf;
94  outf.open("outputfile.txt");
95  string sortname = "No Sort Routine Implemented";
96  for (int i =0; i < 4; i++) {
97      ifstream inf;
98      inf.open("infilerandnums.txt");
99      numofcompares=0;
100     numofmoves=0;
101     for (int k = 0; k < maxnum; k++) {
102         inf >> RandNums[k] >> ws; }
103     if (i==0) {
104         sortname = "Bubble Sort";
105         BubSort(RandNums); }
```

```
106         else if (i==1) {
107             sortname = "Selection Sort";
108             SelSort(RandNums); }
109         else if (i==2) {
110             sortname = "Insert Sort";
111             InsSort(RandNums); }
112         else {
113             sortname = "Quick Sort";
114             QuiSort(RandNums,0, maxnum); }
115         printsort(outf, sortname, RandNums);
116         }
117     system("pause");
118     }
119
120
```