

```
1  #include <iostream>
2  #include <ctime>
3  #include <vector>
4  #include <stdint.h>
5  #include <stdio.h>
6  #include <iostream>
7  #include <string>
8  #include <chrono>
9
10
11 using namespace std;
12
13 void insertionSort(vector<uint16_t> &A);
14
15 int main() {
16     int arrSize[11] = {100, 500, 1000, 5000, 10000, 50000, 100000, 500000,
17         1000000, 5000000, 10000000};
18     string tmp;
19     getline(cin, tmp);
20     printf("Insertion Sort\n");
21     for(int i = 0; i < 11; ++i) {
22         // Create the Array to be sorted with random data
23         srand(clock());
24         vector<uint16_t> A(arrSize[i]);
25         for(int j = 0; j < arrSize[i]; ++j) {
26             A[j] = rand();
27         }
28         // Get the start time
29         auto init = chrono::high_resolution_clock::now();
30         // Run the algorithm
31         insertionSort(A);
32         // Get the end time
33         auto end = chrono::high_resolution_clock::now();
34         // Calculate the elapsed time
35         auto duration2 = end - init;
36         int sec2 = chrono::duration_cast<chrono::seconds>(duration2).count();
37         int nano2 = chrono::duration_cast<chrono::nanoseconds>(duration2).count()
38             % 1000000000;
39
40         printf("%i, %i.%09i\n", arrSize[i], sec2, nano2);
41         // Make sure the output was sorted
42         for (int j = 1; j < A.size(); j++) {
43             if (A[j] < A[j - 1]) {
44                 cout << "WRONG " << j;
45             }
46         }
47         // Wait for user input.
48         //string tmp;
49         getline(cin, tmp);
50     }
```

```
51 // Based on sudo-code on slide
52 void insertionSort(vector<uint16_t> &A) {
53     for(int i = 1; i < A.size(); ++i) {
54         int key = A[i];
55         int j = i - 1;
56         while(j >= 0 and A[j] > key) {
57             A[j + 1] = A[j];
58             j--;
59         }
60         A[j + 1] = key;
61     }
62 }
63
```