DESIGN AND ANALYSIS OF ALGORITHMS LAB

NAME: Rahul Chalwadi

UID: 2021700012

BATCH: A

BRANCH: CSE DS

EXPT. NO.: 1B

AIM: Experiment on finding the running time of an algorithm.

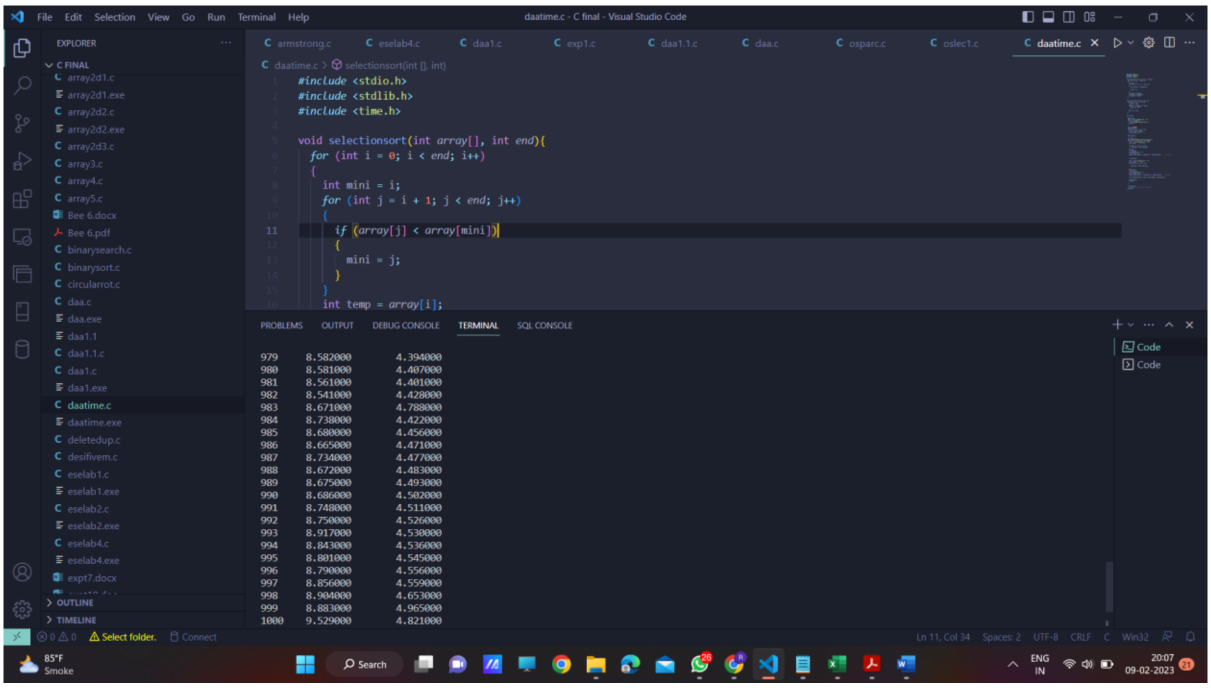
CODE:

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void selectionsort(int array[], int end){
  for (int i = 0; i < end; i++)
  {
    int mini = i;
    for (int j = i + 1; j < end; j++)
    {
      if (array[j] < array[mini])
      {
        mini = j;
      }
    }
    int temp = array[i];
    array[i] = array[mini];
    array[mini] = temp;
  }
}

void insertionsort(int a[], int n){
  for (int i = 1; i < n; i++){
    int key = a[i];
    int j = i - 1;
    while (j >= 0 && a[j] > key){
      a[j + 1] = a[j];
      j = j - 1;
    }
    a[j + 1] = key;
  }
}

int main(){
```

```c
FILE *fptr;
fptr = fopen("randomm.txt", "w");
if (fptr == NULL){
  printf("ERROR Creating File!");
  exit(1);
}

int n = 100000;
srand(time(0));
for (int i = 1; i <= n; i++){
  int r = rand() % 100;
  fprintf(fptr, "%d\n", r);
  // putw(r,fptr);
}
fclose(fptr);

int block = 1;
printf("Block\tSelection\tInsertion\n");
for (int i = 100; i <= n; i += 100){
  fptr = fopen("randomm.txt", "r");
  int arr[i];     // i size

  for (int j = 0; j < i; j++){
    fscanf(fptr, "%d", &arr[j]);
  }
  clock_t t;
  t = clock();
  selectionsort(arr, i);
  t = clock() - t;
  double time_takenss = ((double)t) / CLOCKS_PER_SEC;   // in seconds


  fclose(fptr);

  fptr = fopen("randomm.txt", "r");
  int arr2[i]; // i size
  for (int j = 0; j < i; j++){

    fscanf(fptr, "%d", &arr2[j]);

  }
  clock_t t2;
  t2 = clock();
  insertionsort(arr2, i);
  t2 = clock() - t2;
  double time_takenis = ((double)t2) / CLOCKS_PER_SEC; // in seconds

  printf("%d\t%f\t%f\n", block, time_takenss, time_takenis);
```

```c
    fclose(fptr);
    block++;

  }

  fclose(fptr);
  // insertion less time than selection.
  return 0;
}
```

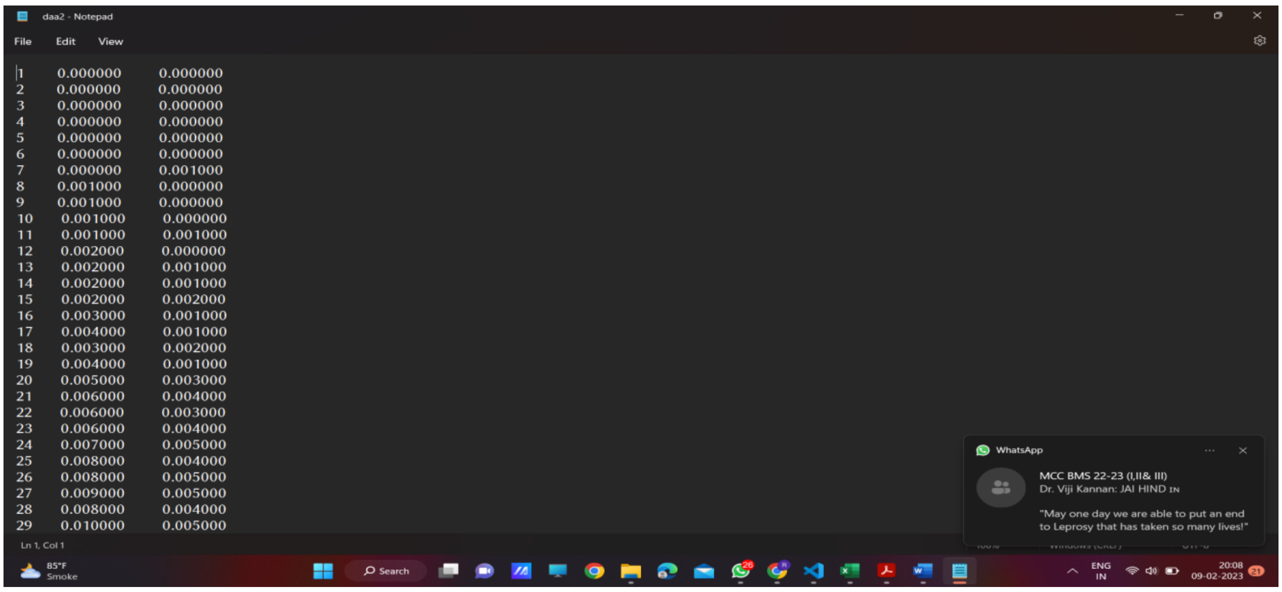
EXECUTION AND OUTPUT:



```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void selectionsort(int array[], int end){
  for (int i = 0; i < end; i++)
  {
    int mini = i;
    for (int j = i + 1; j < end; j++)
    {
      if (array[j] < array[mini])
      {
        mini = j;
      }
    }
    int temp = array[i];
```

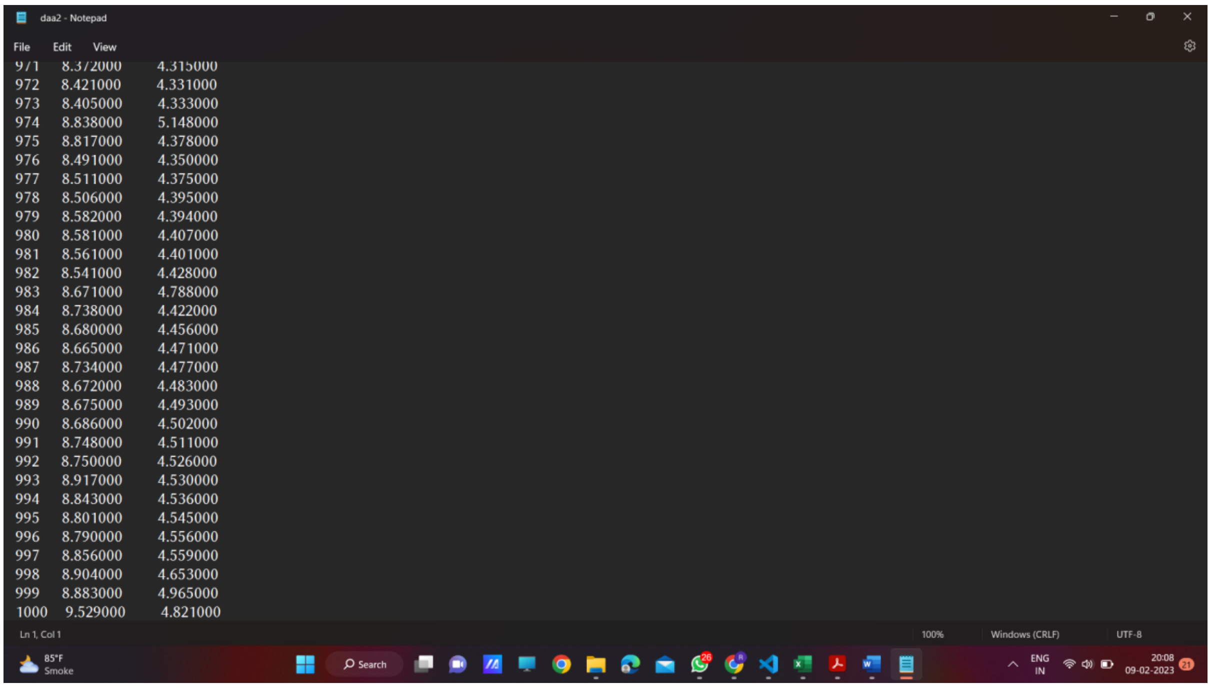
```
979     8.582000        4.394000
980     8.581000        4.407000
981     8.561000        4.401000
982     8.541000        4.428000
983     8.671000        4.788000
984     8.738000        4.422000
985     8.680000        4.456000
986     8.665000        4.471000
987     8.734000        4.477000
988     8.672000        4.483000
989     8.675000        4.493000
990     8.686000        4.502000
991     8.748000        4.511000
992     8.750000        4.526000
993     8.917000        4.530000
994     8.843000        4.536000
995     8.801000        4.545000
996     8.790000        4.556000
997     8.856000        4.559000
998     8.904000        4.653000
999     8.883000        4.965000
1000    9.529000        4.821000
```

output.txt file (first and last 20 rows):



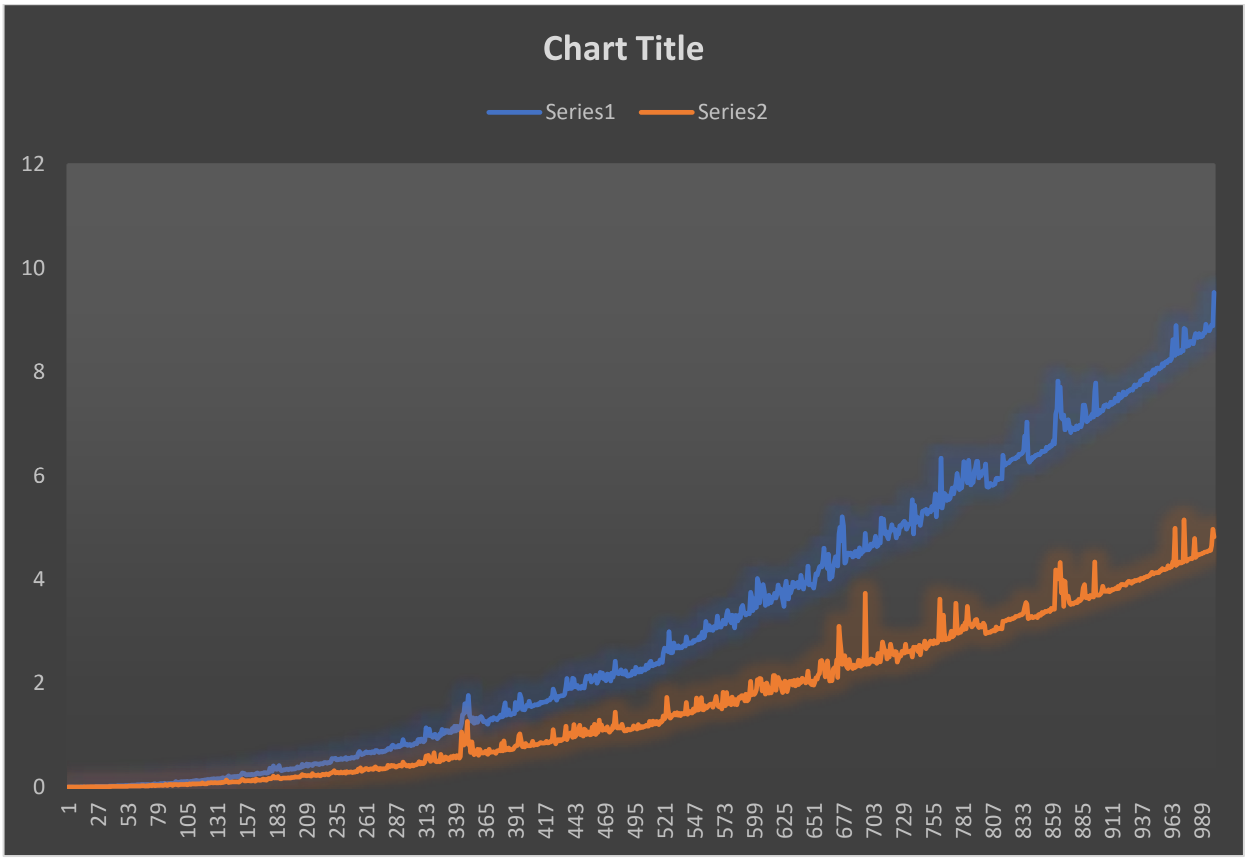
```
1       0.000000        0.000000
2       0.000000        0.000000
3       0.000000        0.000000
4       0.000000        0.000000
5       0.000000        0.000000
6       0.000000        0.000000
7       0.000000        0.001000
8       0.001000        0.000000
9       0.001000        0.000000
10      0.001000        0.000000
11      0.001000        0.001000
12      0.002000        0.000000
13      0.002000        0.001000
14      0.002000        0.001000
15      0.002000        0.002000
16      0.003000        0.001000
17      0.004000        0.001000
18      0.003000        0.002000
19      0.004000        0.001000
20      0.005000        0.003000
21      0.006000        0.004000
22      0.006000        0.003000
23      0.006000        0.004000
24      0.007000        0.005000
25      0.008000        0.004000
26      0.008000        0.005000
27      0.009000        0.005000
28      0.008000        0.004000
29      0.010000        0.005000
```

WhatsApp
MCC BMS 22-23 (I,II& III)
Dr. Viji Kannan: JAI HIND IN
"May one day we are able to put an end to Leprosy that has taken so many lives!"



```
971   8.372000    4.315000
972   8.421000    4.331000
973   8.405000    4.333000
974   8.838000    5.148000
975   8.817000    4.378000
976   8.491000    4.350000
977   8.511000    4.375000
978   8.506000    4.395000
979   8.582000    4.394000
980   8.581000    4.407000
981   8.561000    4.401000
982   8.541000    4.428000
983   8.671000    4.788000
984   8.738000    4.422000
985   8.680000    4.456000
986   8.665000    4.471000
987   8.734000    4.477000
988   8.672000    4.483000
989   8.675000    4.493000
990   8.686000    4.502000
991   8.748000    4.511000
992   8.750000    4.526000
993   8.917000    4.530000
994   8.843000    4.536000
995   8.801000    4.545000
996   8.790000    4.556000
997   8.856000    4.559000
998   8.904000    4.653000
999   8.883000    4.965000
1000   9.529000    4.821000
```

PLOTTING THIS DATA IN EXCEL:



Conclusion:

BY performing this experiment, I understood about the concept of time complexity and verified it with the example by implementing the code in C.