TUGAS 2 STRUKTUR DATA DAN ALGORITMA

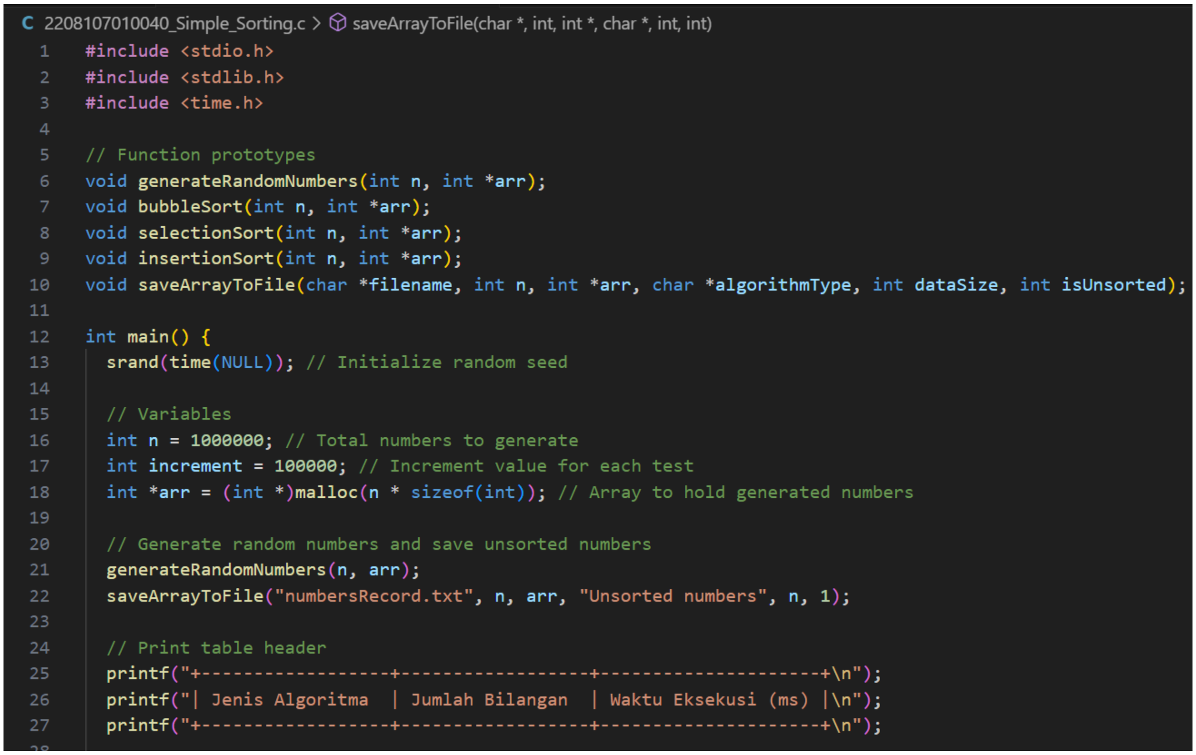
disusun untuk memenuhi tugas mata kuliah Struktur Data dan Algoritma

Oleh:

ZAHRA ZAFIRA
2208107010040

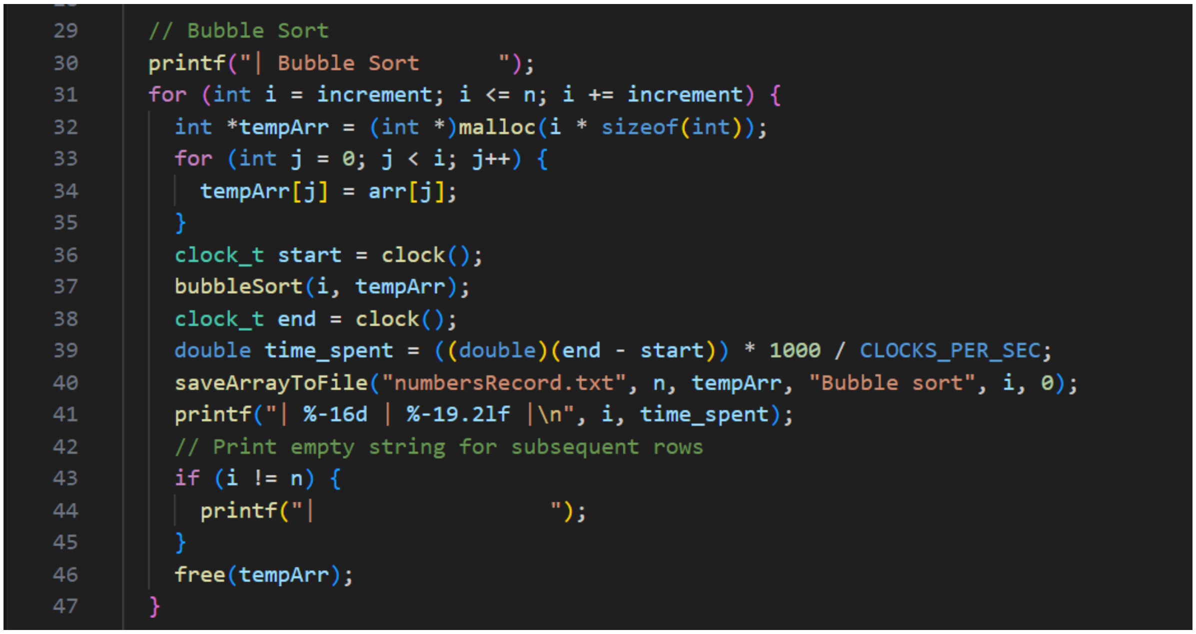
PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH
2024

1. Kode simple sorting menggunakan bahasa c

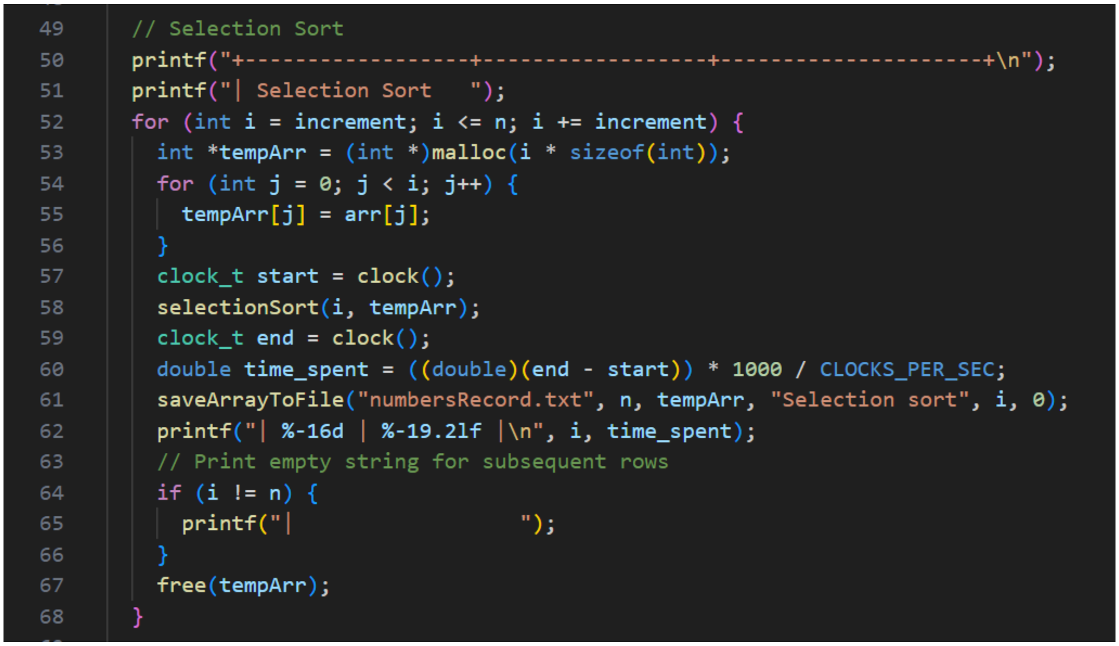


C 2208107010040_Simple_Sorting.c > saveArrayToFile(char *, int, int *, char *, int, int)

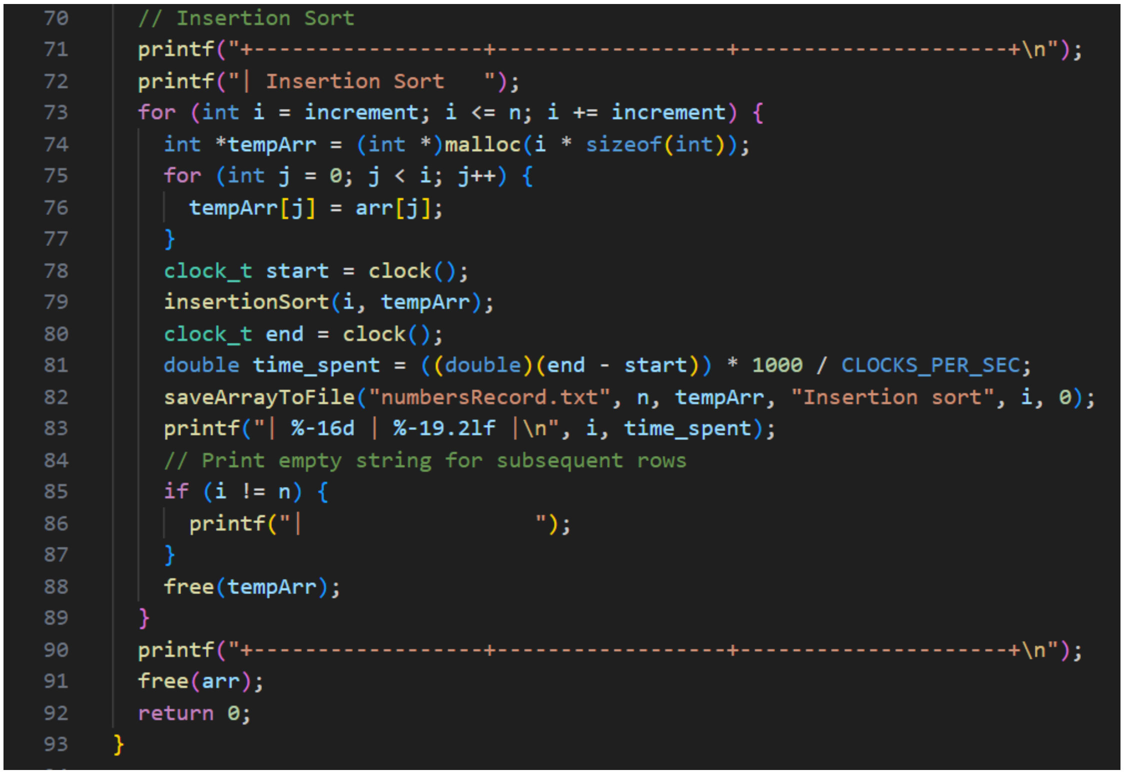
```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

// Function prototypes
void generateRandomNumbers(int n, int *arr);
void bubbleSort(int n, int *arr);
void selectionSort(int n, int *arr);
void insertionSort(int n, int *arr);
void saveArrayToFile(char *filename, int n, int *arr, char *algorithmType, int dataSize, int isUnsorted);

int main() {
    srand(time(NULL)); // Initialize random seed

    // Variables
    int n = 1000000; // Total numbers to generate
    int increment = 100000; // Increment value for each test
    int *arr = (int *)malloc(n * sizeof(int)); // Array to hold generated numbers

    // Generate random numbers and save unsorted numbers
    generateRandomNumbers(n, arr);
    saveArrayToFile("numbersRecord.txt", n, arr, "Unsorted numbers", n, 1);

    // Print table header
    printf("+------------------+------------------+---------------------+\n");
    printf("| Jenis Algoritma  | Jumlah Bilangan  | Waktu Eksekusi (ms) |\n");
    printf("+------------------+------------------+---------------------+\n");
```



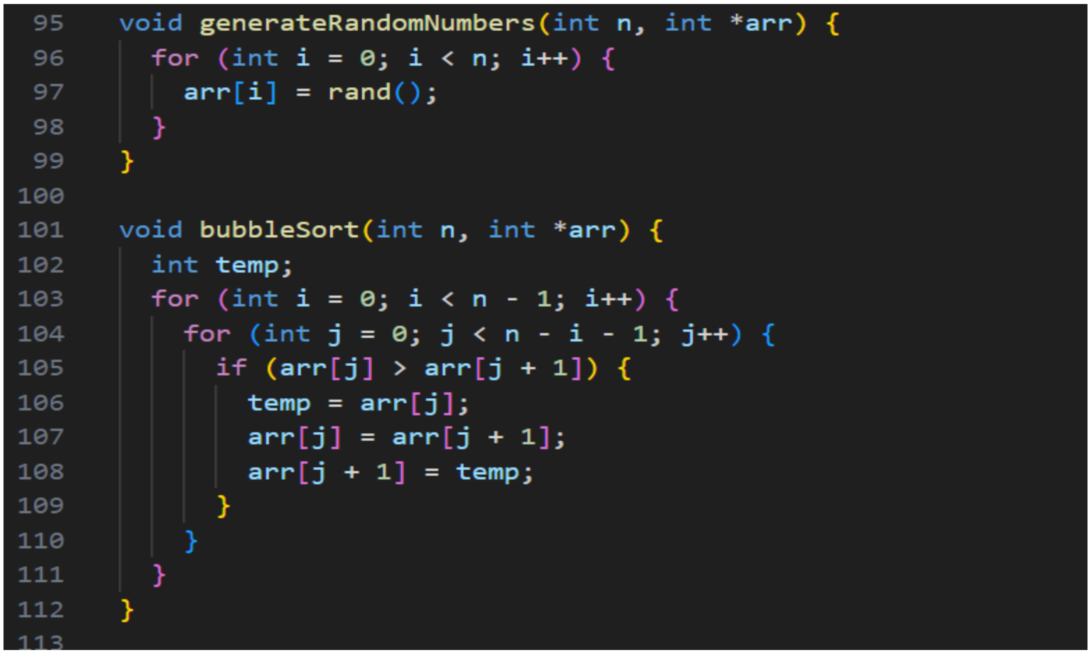
```c
    // Bubble Sort
    printf("| Bubble Sort      ");
    for (int i = increment; i <= n; i += increment) {
        int *tempArr = (int *)malloc(i * sizeof(int));
        for (int j = 0; j < i; j++) {
            tempArr[j] = arr[j];
        }
        clock_t start = clock();
        bubbleSort(i, tempArr);
        clock_t end = clock();
        double time_spent = ((double)(end - start)) * 1000 / CLOCKS_PER_SEC;
        saveArrayToFile("numbersRecord.txt", n, tempArr, "Bubble sort", i, 0);
        printf("| %-16d | %-19.2lf |\n", i, time_spent);
        // Print empty string for subsequent rows
        if (i != n) {
            printf("|                  ");
        }
        free(tempArr);
    }
```



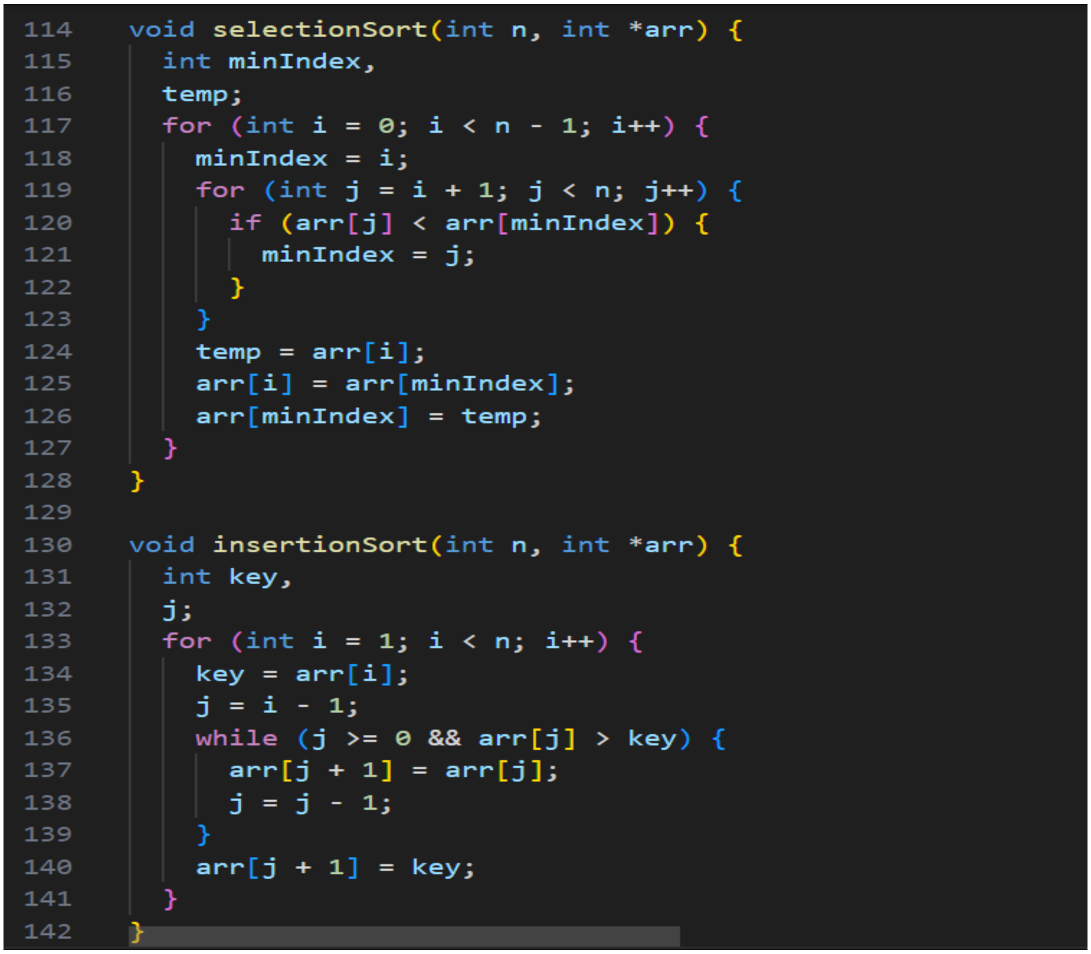
```c
  // Selection Sort
  printf("+------------------+------------------+---------------------+\n");
  printf("| Selection Sort   ");
  for (int i = increment; i <= n; i += increment) {
    int *tempArr = (int *)malloc(i * sizeof(int));
    for (int j = 0; j < i; j++) {
      tempArr[j] = arr[j];
    }
    clock_t start = clock();
    selectionSort(i, tempArr);
    clock_t end = clock();
    double time_spent = ((double)(end - start)) * 1000 / CLOCKS_PER_SEC;
    saveArrayToFile("numbersRecord.txt", n, tempArr, "Selection sort", i, 0);
    printf("| %-16d | %-19.2lf |\n", i, time_spent);
    // Print empty string for subsequent rows
    if (i != n) {
      printf("|                  ");
    }
    free(tempArr);
  }
```



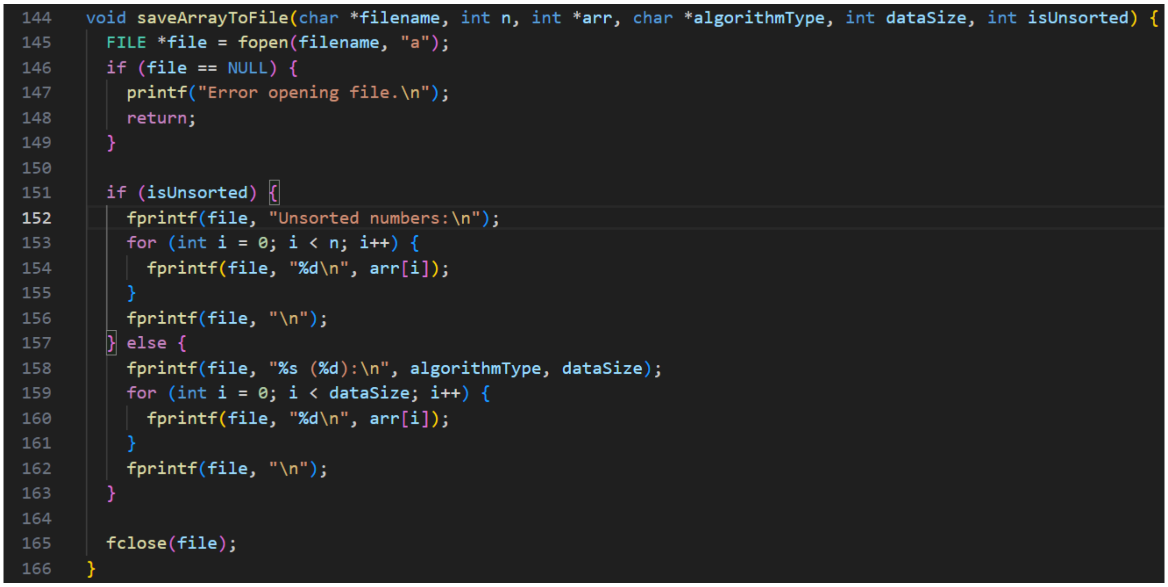
```c
  // Insertion Sort
  printf("+------------------+------------------+---------------------+\n");
  printf("| Insertion Sort   ");
  for (int i = increment; i <= n; i += increment) {
    int *tempArr = (int *)malloc(i * sizeof(int));
    for (int j = 0; j < i; j++) {
      tempArr[j] = arr[j];
    }
    clock_t start = clock();
    insertionSort(i, tempArr);
    clock_t end = clock();
    double time_spent = ((double)(end - start)) * 1000 / CLOCKS_PER_SEC;
    saveArrayToFile("numbersRecord.txt", n, tempArr, "Insertion sort", i, 0);
    printf("| %-16d | %-19.2lf |\n", i, time_spent);
    // Print empty string for subsequent rows
    if (i != n) {
      printf("|                  ");
    }
    free(tempArr);
  }
  printf("+------------------+------------------+---------------------+\n");
  free(arr);
  return 0;
}
```



```c
void generateRandomNumbers(int n, int *arr) {
  for (int i = 0; i < n; i++) {
    arr[i] = rand();
  }
}

void bubbleSort(int n, int *arr) {
  int temp;
  for (int i = 0; i < n - 1; i++) {
    for (int j = 0; j < n - i - 1; j++) {
      if (arr[j] > arr[j + 1]) {
        temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
      }
    }
  }
}
```

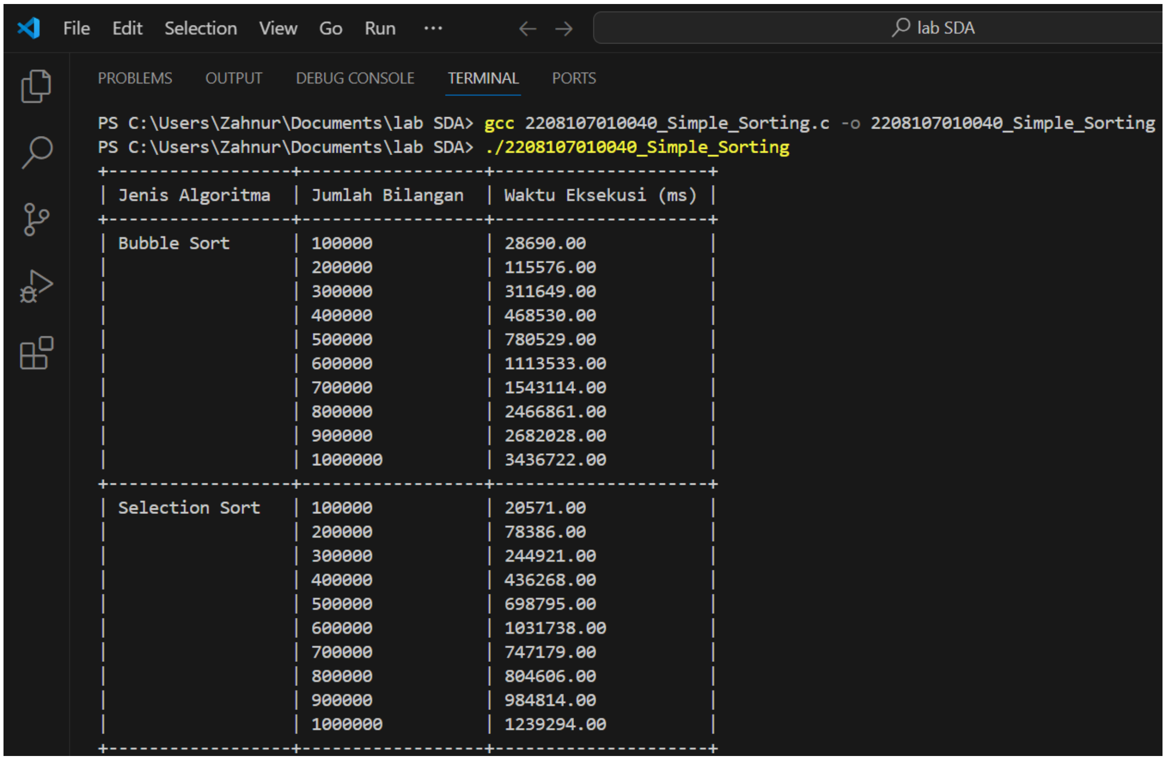


```c
void selectionSort(int n, int *arr) {
  int minIndex,
  temp;
  for (int i = 0; i < n - 1; i++) {
    minIndex = i;
    for (int j = i + 1; j < n; j++) {
      if (arr[j] < arr[minIndex]) {
        minIndex = j;
      }
    }
    temp = arr[i];
    arr[i] = arr[minIndex];
    arr[minIndex] = temp;
  }
}

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

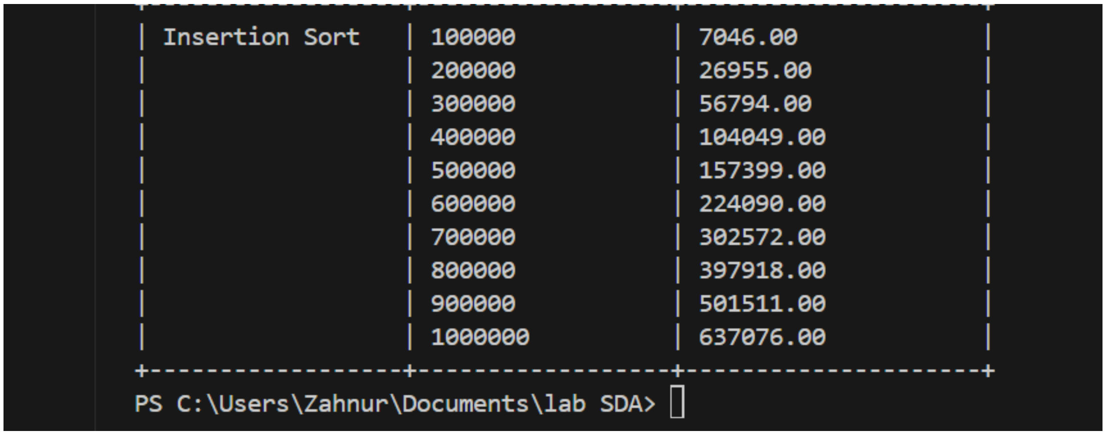


```c
void saveArrayToFile(char *filename, int n, int *arr, char *algorithmType, int dataSize, int isUnsorted) {
  FILE *file = fopen(filename, "a");
  if (file == NULL) {
    printf("Error opening file.\n");
    return;
  }

  if (isUnsorted) {
    fprintf(file, "Unsorted numbers:\n");
    for (int i = 0; i < n; i++) {
      fprintf(file, "%d\n", arr[i]);
    }
    fprintf(file, "\n");
  } else {
    fprintf(file, "%s (%d):\n", algorithmType, dataSize);
    for (int i = 0; i < dataSize; i++) {
      fprintf(file, "%d\n", arr[i]);
    }
    fprintf(file, "\n");
  }

  fclose(file);
}
```

2. Outputnya:



```
PS C:\Users\Zahnur\Documents\lab SDA> gcc 2208107010040_Simple_Sorting.c -o 2208107010040_Simple_Sorting
PS C:\Users\Zahnur\Documents\lab SDA> ./2208107010040_Simple_Sorting
+------------------+------------------+---------------------+
| Jenis Algoritma  | Jumlah Bilangan  | Waktu Eksekusi (ms) |
+------------------+------------------+---------------------+
| Bubble Sort      | 100000           | 28690.00            |
|                  | 200000           | 115576.00           |
|                  | 300000           | 311649.00           |
|                  | 400000           | 468530.00           |
|                  | 500000           | 780529.00           |
|                  | 600000           | 1113533.00          |
|                  | 700000           | 1543114.00          |
|                  | 800000           | 2466861.00          |
|                  | 900000           | 2682028.00          |
|                  | 1000000          | 3436722.00          |
+------------------+------------------+---------------------+
| Selection Sort   | 100000           | 20571.00            |
|                  | 200000           | 78386.00            |
|                  | 300000           | 244921.00           |
|                  | 400000           | 436268.00           |
|                  | 500000           | 698795.00           |
|                  | 600000           | 1031738.00          |
|                  | 700000           | 747179.00           |
|                  | 800000           | 804606.00           |
|                  | 900000           | 984814.00           |
|                  | 1000000          | 1239294.00          |
+------------------+------------------+---------------------+
```



```
| Insertion Sort   | 100000          | 7046.00             |
|                  | 200000          | 26955.00            |
|                  | 300000          | 56794.00            |
|                  | 400000          | 104049.00           |
|                  | 500000          | 157399.00           |
|                  | 600000          | 224090.00           |
|                  | 700000          | 302572.00           |
|                  | 800000          | 397918.00           |
|                  | 900000          | 501511.00           |
|                  | 1000000         | 637076.00           |
+------------------+-----------------+---------------------+
PS C:\Users\Zahnur\Documents\lab SDA> 
```

3. Spesifikasi laptop:
   Device name :LAPTOP-TU6F7B66
   Processor :Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz
   Installed RAM:8,00 GB (7,88 GB usable)
   Device ID :FC59968F-251D-4421-A6CF-CBCE54FB5D3B
   Product ID :00327-35840-06943-AAOEM
   System type :64-bit operating system, x64-based processor