**Nadine Mohamed Mostafa 20107088**

**Assignment #1**

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

void bubble\_sort(int arr[], int n) {

for (int i = 0; i < n-1; i++) {

for (int j = 0; j < n-i-1; j++) {

if (arr[j] > arr[j+1]) {

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

int main() {

int arr[1000];

int n = sizeof(arr)/sizeof(arr[0]);

// Fill the array with random integers between 1 and 1000

for (int i = 0; i < n; i++) {

arr[i] = rand() % 1000 + 1;

}

// Measure the time taken to sort the array

clock\_t start\_time = clock();

bubble\_sort(arr, n);

clock\_t end\_time = clock();

double time\_taken = (double)(end\_time - start\_time) / CLOCKS\_PER\_SEC;

// Print the sorted array and the time taken

printf("Sorted array:\n");

for (int i = 0; i < n; i++) {

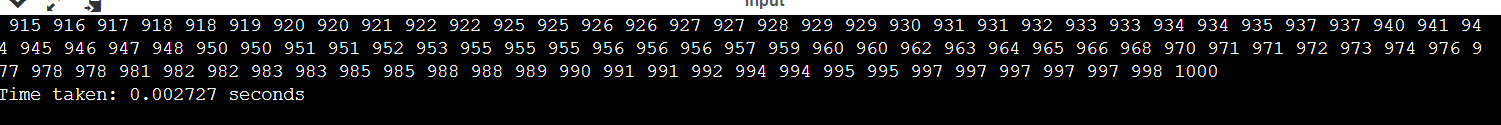
printf("%d ", arr[i]);

}

printf("\nTime taken: %f seconds\n", time\_taken);

return 0;

}



**IMPROVED ONE:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <time.h>**

**#include <stdbool.h>**

**void bubble\_sort\_improved(int arr[], int n) {**

**int i, j;**

**bool swapped;**

**for (i = 0; i < n-1; i++) {**

**swapped = false;**

**for (j = 0; j < n-i-1; j++) {**

**if (arr[j] > arr[j+1]) {**

**int temp = arr[j];**

**arr[j] = arr[j+1];**

**arr[j+1] = temp;**

**swapped = true;**

**}**

**}**

**if (!swapped) {**

**// If no swaps were made in the inner loop, the array is already sorted**

**break;**

**}**

**}**

**}**

**int main() {**

**int arr[1000];**

**int n = sizeof(arr)/sizeof(arr[0]);**

**// Fill the array with random integers between 1 and 1000**

**for (int i = 0; i < n; i++) {**

**arr[i] = rand() % 1000 + 1;**

**}**

**clock\_t start, end;**

**double cpu\_time\_used;**

**start = clock();**

**bubble\_sort\_improved(arr, n);**

**end = clock();**

**cpu\_time\_used = ((double) (end - start)) / CLOCKS\_PER\_SEC;**

**printf("Sorted array: ");**

**for (int i = 0; i < n; i++) {**

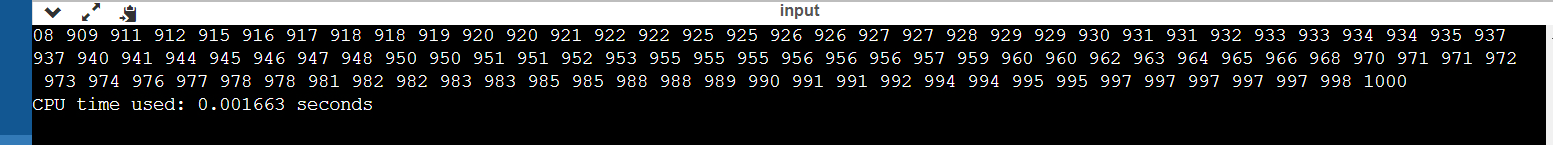
**printf("%d ", arr[i]);**

**}**

**printf("\n");**

**printf("CPU time used: %f seconds\n", cpu\_time\_used);**

**return 0;**

**}**

**Graph plotted between both functions using numpy**

**Chart, line chart

Description automatically generated**