1) Design and implement C/C++ Program to sort a given set of n integer elements using Selection Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

// Online C compiler to run C program online

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

// Function to perform Selection Sort

void selectionSort(int arr[], int n) {

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

int minIndex = i;

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

if (arr[j] < arr[minIndex]) {

minIndex = j;

}

}

if (minIndex != i) {

// Swap arr[i] and arr[minIndex]

int temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

}

}

}

// Function to generate random array of size n

void generateRandomArray(int arr[], int n) {

srand(time(NULL)); // Seed for random number generator

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

arr[i] = rand();

}

}

int main() {

// Vary n from 5000 to 10000

for (int n = 5000; n <= 10000; n += 1000) {

// Dynamically allocate memory for array

int \*arr = (int \*)malloc(n \* sizeof(int));

// Generate random array

generateRandomArray(arr, n);

// Measure time taken for sorting

clock\_t start = clock();

selectionSort(arr, n);

clock\_t end = clock();

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

// Output n and time taken

printf("n = %d, Time taken: %lf seconds\n", n, time\_taken);

// Free dynamically allocated memory

free(arr);

}

return 0;

}

OUTPUT:

n = 5000, Time taken: 0.017971 seconds

n = 6000, Time taken: 0.028647 seconds

n = 7000, Time taken: 0.034901 seconds

n = 8000, Time taken: 0.056108 seconds

n = 9000, Time taken: 0.060766 seconds

n = 10000, Time taken: 0.072657 seconds