EXPERIMENT [18](quick sort)

CODE:

#include <stdio.h>

// Function to swap two elements

void swap(int\* a, int\* b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

// Partition function

int partition(int arr[], int low, int high) {

int pivot = arr[high]; // pivot element

int i = low - 1;

for (int j = low; j < high; j++) {

if (arr[j] < pivot) {

i++;

swap(&arr[i], &arr[j]);

}

}

swap(&arr[i + 1], &arr[high]);

return (i + 1); // return partition index

}

// Quick Sort function (recursive)

void quickSort(int arr[], int low, int high) {

if (low < high) {

int pi = partition(arr, low, high); // partition index

quickSort(arr, low, pi - 1); // sort left part

quickSort(arr, pi + 1, high); // sort right part

}

}

// Function to print array

void printArray(int arr[], int size) {

for (int i = 0; i < size; i++)

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

printf("\n");

}

// Main function

int main() {

int arr[100], n, i;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

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

scanf("%d", &arr[i]);

printf("Original array: ");

printArray(arr, n);

quickSort(arr, 0, n - 1);

printf("Sorted array using Quick Sort: ");

printArray(arr, n);

return 0;

}

OUTPUT:

