TOPIC NO.: 16

TITLE: Quick Sort

Program Code:

#include <stdio.h>

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

int t = \*a;

\*a = \*b;

\*b = t;

}

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

int pivot = array[high];

int i = (low - 1);

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

if (array[j] <= pivot) {

i++;

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

}

}

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

return (i + 1);

}

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

if (low < high) {

int pi = partition(array, low, high);

quickSort(array, low, pi - 1);

quickSort(array, pi + 1, high);

}

}

Output:

Write the size of array : 4

element at 0 = 5

element at 1 = 1

element at 2 = 9

element at 3 = 4

Unsorted Array : 5 1 9 4

Sorted array in ascending order : 1 4 5 9

DATE: 03.05.24

void printArray(int array[], int size)

{

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

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

}

printf("\n");

}

int main() {

int i, r;

printf("Write the size of array : ");

scanf("%d", &r);

int arr[r];

for (i = 0; i < r; i++) {

printf("element at %d = ", i);

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

}

printf("Unsorted Array : \n");

printArray(arr, r);

quickSort(arr, 0, r - 1);

printf("Sorted array in ascending order : \n");

printArray(arr, r);

return 0;

}