WAP for quick sort using recursion

import java.util.Scanner;

public class Runner {

static Scanner *s* = new Scanner(System.*in*);

public static int[] takeInput(){

int size = *s*.nextInt();

int[] input = new int[size];

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

input[i] = *s*.nextInt();

}

return input;

}

public static void main(String[] args) {

int[] input = *takeInput*();

*quickSort*(input);

for(int i = 0; i < input.length; i++) {

System.*out*.print(input[i] + " ");

}

}

private static void quickSort(int[] input) {

// TODO Auto-generated method stub

*quickSort*(input,0,input.length-1);

}

private static void quickSort(int[] input, int si, int ei) {

// TODO Auto-generated method stub

if(si>=ei) {

return ;

}

int pivotpos=*partition*(input,si,ei);

*quickSort*(input,si,pivotpos-1);

*quickSort*(input,pivotpos+1,ei);

}

private static int partition(int[] input, int si, int ei) {

int pivot = input[ei];

int i = (si-1);

for (int j=si; j<ei; j++)

{

if (input[j] <= pivot)

{

i++;

int temp = input[i];

input[i] = input[j];

input[j] = temp;

}

}

int temp = input[i+1];

input[i+1] = input[ei];

input[ei] = temp;

return i+1;

}

}