E2 Quick Sort

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#define SIZE 15

int count=0;

void printing\_func(int Arr[],int value)

{

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

{

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

}

printf("\n");

}

void swaping\_func(int Arr[], int p, int q)

{

int temp\_val;

temp\_val = Arr[p];

Arr[p] = Arr[q];

Arr[q] = temp\_val;

}

void quick\_sort(int a[], int left\_ptr, int right\_ptr)

{

int left, right, pivot;

if(left\_ptr >= right\_ptr)

{

return;

}

left = left\_ptr;

right = right\_ptr;

// pivot selection

pivot = a[(left\_ptr + right\_ptr) /2];

// partition

while(left <= right)

{

while(a[left] < pivot)

{count++;

left++;

}

while(a[right] > pivot)

{count++;

right--;

}

if(left <= right)

{

swaping\_func(a,left,right);

left++; right--;

}

//printf("UNDER PROCESS : \n");

//print(a,right\_index);

}

// recursion

//printf("in process comp : %d\n",count);

quick\_sort(a,left\_ptr,right);

quick\_sort(a,left,right\_ptr);

}

int main()

{

int a[SIZE],i,k;

printf("ENTER TOTAL NO. OF ELEMENTS\n");

scanf("%d",&k);

printf("ENTER ELEMENTS\n");

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

{

printf("Enter element %d : ",i+1);

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

}

printf("\nGIVEN ORDER : \n");

printing\_func(a,k-1);

quick\_sort(a,0,k-1);

printf("\n\nFINAL ORDER : \n");

printing\_func(a,k-1);

printf("\n\ncomplexity is : %d\n",count);

}