#include <stdio.h>

#include <malloc.h>

#include "RandomPermutation.c"

void QuickSort (int\* Arr, int Start, int End);

void DispArr (int\* Arr, int Size);

int main (void) {

int \*Arr = malloc (sizeof(int) \* 100);

GenRandArr(Arr, 100);

DispArr(Arr, 100);

QuickSort(Arr, 0, 99);

DispArr(Arr, 100);

return 0;

}

void QuickSort (int\* Arr, int Start, int End) {

int FirstVal = Start;

int LastVal = End;

int PIndex = Start;

int Pivot = Arr [Start];

int Temp = 0;

if (Start < End) {

while (Start <= End) {

if (Arr[Start] == Pivot) {

Start++;

}

else if (Arr[Start] < Pivot) {

// Swap Arr[Start] and Arr[PIndex]

Temp = Arr[Start];

Arr[Start] = Arr[PIndex];

Arr[PIndex] = Temp;

PIndex++;

Start++;

}

else if (Arr[Start] > Pivot) {

// Swap Arr[Start] with Arr[End]

Temp = Arr[Start];

Arr[Start] = Arr[End];

Arr[End] = Temp;

End--;

}

}

// Call QuickSort() for left part. i.e from FirstVal to PIndex - 1.

QuickSort(Arr, FirstVal, PIndex - 1);

// Call QuickSort() for right part. i.e from Start to LastVal.

QuickSort(Arr, Start, LastVal);

}

}

void DispArr (int\* Arr, int Size) {

int i = 0;

printf ("\n");

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

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

}

printf ("\n");

}