<QuickSort library 함수 사용>

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

typedef struct node

{

double x;

double y;

double val;

}node;

int comND(const void \*a, const void \*b)

{

node \*c = (node \*)a;

node \*d = (node \*)b;

if (c->val < d->val) return -1;

else if (c->val > d->val) return 1;

else return 0;

}

int main()

{

node \*v;

FILE \*ofp = fopen("data.txt", "w+");

v = (node \*)calloc(10, sizeof(node));

fprintf(ofp, "Before Sorting\n");

for (int i = 0; i < 10; i++)

{

v[i].x = rand() / (double)RAND\_MAX \* 100;

v[i].y = rand() / (double)RAND\_MAX \* 100;

v[i].val = sqrt(pow(50-v[i].x, 2) + pow(50-v[i].y, 2));

fprintf(ofp, "%6.2lf%6.2lf%6.2lf\n", v[i].x, v[i].y, v[i].val);

}

fprintf(ofp,"\n\n");

qsort(v, 10, sizeof(node), comND);

fprintf(ofp, "After Sorting\n");

for (int i = 0; i < 10; i++)

{

fprintf(ofp, "%6.2lf%6.2lf%6.2lf\n", v[i].x, v[i].y, v[i].val);

}

fprintf(ofp, "\n");

return 0;

}

<QuickSort library 함수 미사용>

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

typedef struct node

{

double x;

double y;

double val;

} node;

void swap(node \*a, node \*b)

{

node temp = \*a;

\*a = \*b;

\*b = temp;

}

int Partition(node \*arr, int left, int right)

{

double pivot = arr[left].val;

int low = left + 1;

int high = right;

while (low <= high)

{

while (pivot > arr[low].val)

low++;

while (pivot < arr[high].val)

high--;

if (low <= high)

swap(&arr[low], &arr[high]);

}

swap(&arr[left], &arr[high]);

return high;

}

void QuickSort(node \*arr, int left, int right)

{

if (left <= right)

{

int pivot = Partition(arr, left, right);

QuickSort(arr, left, pivot - 1);

QuickSort(arr, pivot + 1, right);

}

}

int main()

{

node \*arr;

FILE \*ofp = fopen("datadata.txt", "w+");

arr = (node \*)calloc(10, sizeof(node));

int arr\_size = 10;

fprintf(ofp, "Before Sorting\n");

for (int i = 0; i < arr\_size; i++)

{

arr[i].x = rand() / (double)RAND\_MAX \* 100;

arr[i].y = rand() / (double)RAND\_MAX \* 100;

arr[i].val = sqrt(pow(50 - arr[i].x, 2) + pow(50 - arr[i].y, 2));

fprintf(ofp, "%8.2lf%8.2lf%8.2lf\n", arr[i].x, arr[i].y, arr[i].val);

}

QuickSort(arr, 0, arr\_size - 1);

fprintf(ofp, "After Sorting\n");

for (int i = 0; i < arr\_size; i++)

{

fprintf(ofp, "%8.2lf%8.2lf%8.2lf\n", arr[i].x, arr[i].y, arr[i].val);

}

}

<결과>

