CS2263 Lab 7

Stephen Cole

3553803

Exercise One:

Compile

A picture containing ball, monitor, orange, cat

Description automatically generated

Successful program run

A black sign with white text

Description automatically generated

sortTest.c

#include <stdio.h>

#include <stdlib.h>

#define LENGTH 5

void sortArray(int\* arr);

int main(void)

{

int arr[] = {3,4,5,1,2};

printf("Array pre-sort: %d, %d, %d, %d, %d\n", arr[0], arr[1], arr[2], arr[3], arr[4]);

sortArray(arr);

printf("Array post-sort: %d, %d, %d, %d, %d\n", arr[0], arr[1], arr[2], arr[3], arr[4]);

return 1;

}

void sortArray(int\* arr)

{

int i,j,a;

int n = LENGTH;

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

{

for (j = i + 1; j < n; ++j)

{

if (arr[i] > arr[j])

{

a = arr[i];

arr[i] = arr[j];

arr[j] = a;

}

}

}

}

Exercise Two:

A picture containing object, monitor, television, cat

Description automatically generated

A picture containing bottle, photo, black, sign

Description automatically generated

sortTest.c

#include <stdio.h>

#include <stdlib.h>

#define LENGTH 5

void sortArray(int\* arr, int (\*comp)(int, int));

int comp(int a, int b);

int main(void)

{

int arr[] = {3,4,5,1,2};

printf("Array pre-sort: %d, %d, %d, %d, %d\n", arr[0], arr[1], arr[2], arr[3], arr[4]);

sortArray(arr, comp);

printf("Array post-sort: %d, %d, %d, %d, %d\n", arr[0], arr[1], arr[2], arr[3], arr[4]);

return 1;

}

int comp(int a, int b)

{

if(a > b)

return 1;

return 0;

}

void sortArray(int\* arr, int (\*comp)(int, int))

{

int i,j,a;

int n = LENGTH;

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

{

for (j = i + 1; j < n; ++j)

{

if (comp(arr[i], arr[j]))

{

a = arr[i];

arr[i] = arr[j];

arr[j] = a;

}

}

}

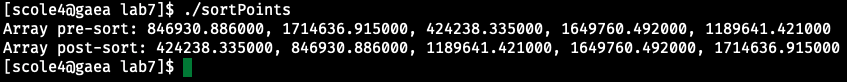
}

Exercise Three:

Compile



Run



sortPoint2D.c

#include <stdio.h>

#include <stdlib.h>

#include "Point2D.h"

#include <math.h>

#define LENGTH 5

void sortArray(Point2D\* arr[], int (\*comp)(Point2D\*, Point2D\*));

int comp(Point2D\* a, Point2D\* b);

int main(void)

{

Point2D\* arr[5];

int i;

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

{

Point2D\* pt = createPoint2D((double)rand()/1000, (double)rand()/1000);

arr[i] = pt;

}

printf("Array pre-sort: %lf, %lf, %lf, %lf, %lf\n", arr[0]->x, arr[1]->x, arr[2]->x, arr[3]->x, arr[4]->x);

sortArray(arr, comp);

printf("Array post-sort: %lf, %lf, %lf, %lf, %lf\n", arr[0]->x, arr[1]->x, arr[2]->x, arr[3]->x, arr[4]->x);

return 1;

}

int comp(Point2D\* a, Point2D\* b)

{

if(a->x > b->x)

return 1;

return 0;

}

void sortArray(Point2D\* arr[], int (\*comp)(Point2D\*, Point2D\*))

{

int i,j;

Point2D\* a = mallocPoint2D();

int n = LENGTH;

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

{

for (j = i + 1; j < n; ++j)

{

if (comp(arr[i], arr[j]))

{

a = arr[i];

arr[i] = arr[j];

arr[j] = a;

}

}

}

}