Christiaan Cronje

Lab 2 Option A

COSC 2347 8AM

Due: April 21, 2017

Submitted: April 21, 2017

**emp\_struct.h**

struct employee{

char surname[20];

char given[20];

char dept[20];

float payRate;

char eyeColor[20];

};

**SortRecords.c**

void SortRecords(struct employee emp[50], int count)

{

for (int j = 1; j < count; j++)

{

int i = j - 1;

struct employee temp = emp[j];

while(1)

{

if(strcmp(temp.dept, emp[i].dept) >= 0)

{

emp[i+1] = temp;

break;

}

else

{

emp[i+1] = emp[i];

i--;

if (i < 0)

{

emp[0] = temp;

break;

}

}

}

}

return;

}

**PtrSortRecordsC.c**

void PtrSortRecordsC(struct employee \*emp[50], int count)

{

for (int j = 1; j < count; j++)

{

int i = j - 1;

struct employee temp = (\*emp)[j];

while(1)

{

if(strcmp((temp).dept, (\*emp)[i].dept) >= 0)

{

(\*emp)[i+1] = temp;

break;

}

else

{

(\*emp)[i+1] = (\*emp)[i];

i--;

if (i < 0)

{

(\*emp)[0] = temp;

break;

}

}

}

}

return;

}

**PtrSortRecordsB.c**

#include <string.h>

#include "emp\_struct.h"

void PtrSortRecordsB(struct employee \*emp[50], int count)

{

for (int j = 1; j < count; j++)

{

int i = j - 1;

struct employee temp = \*(\*emp + j);

while(1)

{

if(strcmp((temp).dept, (\*(\*emp + i)).dept) >= 0)

{

\*(\*emp + i + 1) = temp;

break;

}

else

{

\*(\*emp+i+1) = \*(\*emp+i);

i--;

if (i < 0)

{

\*(\*emp) = temp;

break;

}

}

}

}

return;

}

**TemplateSortRecords.cpp**

int cmpN(struct employee a, struct employee b)

{

return strcmp(a.surname, b.surname);

}

int cmpD(struct employee a, struct employee b)

{

return strcmp(a.dept, b.dept);

}

template<class T>

void TemplateSortRecords(T emp[50], int count, compare cmp)

{

for (int j = 1; j < count; j++)

{

int i = j - 1;

T temp = emp[j];

while(1)

{

if((\*cmp)(temp, emp[i]) >= 0)

{

emp[i+1] = temp;

break;

}

else

{

emp[i+1] = emp[i];

i--;

if (i < 0)

{

emp[0] = temp;

break;

}

}

}

}

return;

}

**ArraySorting.cpp**

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

typedef int(\*compare) (struct employee, struct employee);

#include "PtrSortRecordsB.c"

#include "SortRecords.c"

#include "PtrSortRecordsC.c"

#include "TemplateSortRecords.cpp"

void PrintRecords(struct employee \*emp, int count, FILE\* output);

int main( int argc, char \*argv[] )

{

FILE \*fileIn; // Logical output file name in program.

char fileInName[15]; // Actual path and file name on disk.

FILE \*fileOut;

char fileOutName[20];

if(argc >= 3)

{

strcpy(fileInName, argv[1]);

strcpy(fileOutName, argv[2]);

}

else // else get input for file names

{

printf("%s", "Enter the name of the input file [max 15 characters]: ");

scanf("%s", fileInName);

printf("%s", "Enter the name of the file to hold the results [max 20 characters]: ");

scanf("%s", fileOutName);

}

//try to open input file

fileIn = fopen(fileInName, "r");

if( fileIn == NULL ){

printf("%s\n", "Input file does not exist! Program terminating.");

exit(1); // exit(0) indicates program success, exit(1) failure.

}

//try to open output file

fileOut = fopen(fileOutName, "w+");

if( fileOut == NULL ){

printf("%s\n", "Could not create the output file! Program terminating.");

exit(1);

}

// Program Initializations

struct employee records[50];

int c = 0;

// Obtain information from file and calculate the payroll.

while(!feof(fileIn))

{

//get all info from input file

fscanf(fileIn, "%s", records[c].surname);

fscanf(fileIn, "%s", records[c].given);

fscanf(fileIn, "%s", records[c].dept);

fscanf(fileIn, "%f", &records[c].payRate);

fscanf(fileIn, "%s", records[c].eyeColor);

c++;

}

int recordCnt = c;

// Create data sets to be processed by functions

// this will not override original copy

struct employee dataD[50];

memcpy(dataD, records, sizeof(records));

struct employee dataC[50];

memcpy(dataC, records, sizeof(records));

struct employee dataB[50];

memcpy(dataB, records, sizeof(records));

struct employee dataA1[50];

memcpy(dataA1, records, sizeof(records));

struct employee dataA2[50];

memcpy(dataA2, records, sizeof(records));

//output raw data

printf("\nUnsorted:\n");

fprintf(fileOut, "Unsorted:\n");

PrintRecords(records, recordCnt, fileOut);

//sort data for D option

SortRecords(dataD, recordCnt);

// print sorted data

printf("\nSorted D:\n");

fprintf(fileOut, "\nSorted D:\n");

PrintRecords(dataD, recordCnt, fileOut);

// Create pointer array to sort using C option

struct employee \*ptrsC[50];

for (int i = 0; i < recordCnt; ++i){ ptrsC[i] = &dataC[i]; }

PtrSortRecordsC(ptrsC, recordCnt);

printf("\nSorted C:\n");

fprintf(fileOut, "\nSorted C:\n");

PrintRecords(dataC, recordCnt, fileOut);

// Create pointer array to sort using B option

struct employee \*ptrsB[50];

for (int i = 0; i < recordCnt; ++i){ ptrsB[i] = &dataB[i]; }

PtrSortRecordsC(ptrsB, recordCnt);

printf("\nSorted B:\n");

fprintf(fileOut, "\nSorted B:\n");

PrintRecords(dataB, recordCnt, fileOut);

// Template of type struct employee sorting A option by Surname. Uses cmpN to compare

TemplateSortRecords<struct employee>(dataA1, recordCnt, cmpN);

printf("\nTemplate Sorted A by Surname:\n");

fprintf(fileOut, "\nTemplate Sorted A by Surname:\n");

PrintRecords(dataA1, recordCnt, fileOut);

// Template of type struct employee sorting A option by Department. Uses cmpD to compare

TemplateSortRecords<struct employee>(dataA2, recordCnt, cmpD);

printf("\nTemplate Sorted A by Department:\n");

fprintf(fileOut, "\nTemplate Sorted A by Department:\n");

PrintRecords(dataA2, recordCnt, fileOut);

// Close files

fclose(fileIn);

fclose(fileOut);

return 0; // Indicate success to the runtime environment.

}

void PrintRecords(struct employee \*emp, int count, FILE\* output)

{

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

{

printf("%-20s%-20s%-20s%-8.2f%-20s\n", emp[i].surname, emp[i].given, emp[i].dept, emp[i].payRate, emp[i].eyeColor);

fprintf(output, "%-20s%-20s%-20s%-8.2f%-20s\n", emp[i].surname, emp[i].given, emp[i].dept, emp[i].payRate, emp[i].eyeColor);

}

return;

}

**Makefile**

FILENAME=ArraysSorting

all:

gcc -c -Wall PtrSortRecordsB.c

ar -cru libPtrSort.a PtrSortRecordsB.o

gcc -o ./$(FILENAME) ArraysSorting.cpp -L -lPtrSort

clean:

rm -rf \*.o \*.a $(FILENAME)

run: all

./$(FILENAME)

runIO: all

./$(FILENAME) input.txt output.txt

**input.txt**

Nienberg Dylan Computer\_Services 18.24 Blue

Fitzgerald Matthew Accounting 8.96 Green

DeLuna-Solis Eduardo Sales 9.57 Hazel

McGowen Andrew Accounting 11.31 Blue

Moreno

Issac

Sales

16.27

Hazel

Welch

Wade Computer\_Services 22.01 Blue

Fruci Jordan Sales 19.26 Blue

Gazeau Valentin Management 6.72 Steel

Silva Julian Management 17.35 Gray

Brook Tyler Hacker 45.44 Green

Al-Husain

Thamer

Manufacturing

12.22

Blue

Hester Travis Manufacturing 1.34 Blue

Howell Brandon Sales 30.00 Hazel

Feuling Matthew Computer\_Services 15.21 Green

Jacobson Jake Water\_Sports 53.34 Brown

Zhang Zhaohe Accounting 8.96 Green

Pennington Jeremy Customer\_Service 78.93 Blue

Oliver Logan Sales 16.27 Hazel

Pike Jeremy Computer\_Services 22.01 Blue

Celaya Esteban Accounting 11.11 Blue

Horn John Sales 19.26 Blue

Brosh Domenic Management 6.72 Steel

Kavanaugh Sarah Shipping 3.22 Steel

Dominguez Jared Water\_Sports 53.34

Brown

Harding Andrew

Sales 16.37 Gray

Fruci

Jordan

Sales 19.26 Blue

Pike Jeremy Computer\_Services 22.01 Blue

Curtis Harrison Hacker 78.33 Hazel

Unsorted:

Nienberg Dylan Computer\_Services 18.24 Blue

Fitzgerald Matthew Accounting 8.96 Green

DeLuna-Solis Eduardo Sales 9.57 Hazel

McGowen Andrew Accounting 11.31 Blue

Moreno Issac Sales 16.27 Hazel

Welch Wade Computer\_Services 22.01 Blue

Fruci Jordan Sales 19.26 Blue

Gazeau Valentin Management 6.72 Steel

Silva Julian Management 17.35 Gray

Brook Tyler Hacker 45.44 Green

Al-Husain Thamer Manufacturing 12.22 Blue

Hester Travis Manufacturing 1.34 Blue

Howell Brandon Sales 30.00 Hazel

Feuling Matthew Computer\_Services 15.21 Green

Jacobson Jake Water\_Sports 53.34 Brown

Zhang Zhaohe Accounting 8.96 Green

Pennington Jeremy Customer\_Service 78.93 Blue

Oliver Logan Sales 16.27 Hazel

Pike Jeremy Computer\_Services 22.01 Blue

Celaya Esteban Accounting 11.11 Blue

Horn John Sales 19.26 Blue

Brosh Domenic Management 6.72 Steel

Kavanaugh Sarah Shipping 3.22 Steel

Dominguez Jared Water\_Sports 53.34 Brown

Harding Andrew Sales 16.37 Gray

Fruci Jordan Sales 19.26 Blue

Pike Jeremy Computer\_Services 22.01 Blue

Curtis Harrison Hacker 78.33 Hazel

Sorted D:

Fitzgerald Matthew Accounting 8.96 Green

McGowen Andrew Accounting 11.31 Blue

Zhang Zhaohe Accounting 8.96 Green

Celaya Esteban Accounting 11.11 Blue

Nienberg Dylan Computer\_Services 18.24 Blue

Welch Wade Computer\_Services 22.01 Blue

Feuling Matthew Computer\_Services 15.21 Green

Pike Jeremy Computer\_Services 22.01 Blue

Pike Jeremy Computer\_Services 22.01 Blue

Pennington Jeremy Customer\_Service 78.93 Blue

Brook Tyler Hacker 45.44 Green

Curtis Harrison Hacker 78.33 Hazel

Gazeau Valentin Management 6.72 Steel

Silva Julian Management 17.35 Gray

Brosh Domenic Management 6.72 Steel

Al-Husain Thamer Manufacturing 12.22 Blue

Hester Travis Manufacturing 1.34 Blue

DeLuna-Solis Eduardo Sales 9.57 Hazel

Moreno Issac Sales 16.27 Hazel

Fruci Jordan Sales 19.26 Blue

Howell Brandon Sales 30.00 Hazel

Oliver Logan Sales 16.27 Hazel

Horn John Sales 19.26 Blue

Harding Andrew Sales 16.37 Gray

Fruci Jordan Sales 19.26 Blue

Kavanaugh Sarah Shipping 3.22 Steel

Jacobson Jake Water\_Sports 53.34 Brown

Dominguez Jared Water\_Sports 53.34 Brown

Sorted C:

Fitzgerald Matthew Accounting 8.96 Green

McGowen Andrew Accounting 11.31 Blue

Zhang Zhaohe Accounting 8.96 Green

Celaya Esteban Accounting 11.11 Blue

Nienberg Dylan Computer\_Services 18.24 Blue

Welch Wade Computer\_Services 22.01 Blue

Feuling Matthew Computer\_Services 15.21 Green

Pike Jeremy Computer\_Services 22.01 Blue

Pike Jeremy Computer\_Services 22.01 Blue

Pennington Jeremy Customer\_Service 78.93 Blue

Brook Tyler Hacker 45.44 Green

Curtis Harrison Hacker 78.33 Hazel

Gazeau Valentin Management 6.72 Steel

Silva Julian Management 17.35 Gray

Brosh Domenic Management 6.72 Steel

Al-Husain Thamer Manufacturing 12.22 Blue

Hester Travis Manufacturing 1.34 Blue

DeLuna-Solis Eduardo Sales 9.57 Hazel

Moreno Issac Sales 16.27 Hazel

Fruci Jordan Sales 19.26 Blue

Howell Brandon Sales 30.00 Hazel

Oliver Logan Sales 16.27 Hazel

Horn John Sales 19.26 Blue

Harding Andrew Sales 16.37 Gray

Fruci Jordan Sales 19.26 Blue

Kavanaugh Sarah Shipping 3.22 Steel

Jacobson Jake Water\_Sports 53.34 Brown

Dominguez Jared Water\_Sports 53.34 Brown

Sorted B:

Fitzgerald Matthew Accounting 8.96 Green

McGowen Andrew Accounting 11.31 Blue

Zhang Zhaohe Accounting 8.96 Green

Celaya Esteban Accounting 11.11 Blue

Nienberg Dylan Computer\_Services 18.24 Blue

Welch Wade Computer\_Services 22.01 Blue

Feuling Matthew Computer\_Services 15.21 Green

Pike Jeremy Computer\_Services 22.01 Blue

Pike Jeremy Computer\_Services 22.01 Blue

Pennington Jeremy Customer\_Service 78.93 Blue

Brook Tyler Hacker 45.44 Green

Curtis Harrison Hacker 78.33 Hazel

Gazeau Valentin Management 6.72 Steel

Silva Julian Management 17.35 Gray

Brosh Domenic Management 6.72 Steel

Al-Husain Thamer Manufacturing 12.22 Blue

Hester Travis Manufacturing 1.34 Blue

DeLuna-Solis Eduardo Sales 9.57 Hazel

Moreno Issac Sales 16.27 Hazel

Fruci Jordan Sales 19.26 Blue

Howell Brandon Sales 30.00 Hazel

Oliver Logan Sales 16.27 Hazel

Horn John Sales 19.26 Blue

Harding Andrew Sales 16.37 Gray

Fruci Jordan Sales 19.26 Blue

Kavanaugh Sarah Shipping 3.22 Steel

Jacobson Jake Water\_Sports 53.34 Brown

Dominguez Jared Water\_Sports 53.34 Brown

Template Sorted A by Surname:

Al-Husain Thamer Manufacturing 12.22 Blue

Brook Tyler Hacker 45.44 Green

Brosh Domenic Management 6.72 Steel

Celaya Esteban Accounting 11.11 Blue

Curtis Harrison Hacker 78.33 Hazel

DeLuna-Solis Eduardo Sales 9.57 Hazel

Dominguez Jared Water\_Sports 53.34 Brown

Feuling Matthew Computer\_Services 15.21 Green

Fitzgerald Matthew Accounting 8.96 Green

Fruci Jordan Sales 19.26 Blue

Fruci Jordan Sales 19.26 Blue

Gazeau Valentin Management 6.72 Steel

Harding Andrew Sales 16.37 Gray

Hester Travis Manufacturing 1.34 Blue

Horn John Sales 19.26 Blue

Howell Brandon Sales 30.00 Hazel

Jacobson Jake Water\_Sports 53.34 Brown

Kavanaugh Sarah Shipping 3.22 Steel

McGowen Andrew Accounting 11.31 Blue

Moreno Issac Sales 16.27 Hazel

Nienberg Dylan Computer\_Services 18.24 Blue

Oliver Logan Sales 16.27 Hazel

Pennington Jeremy Customer\_Service 78.93 Blue

Pike Jeremy Computer\_Services 22.01 Blue

Pike Jeremy Computer\_Services 22.01 Blue

Silva Julian Management 17.35 Gray

Welch Wade Computer\_Services 22.01 Blue

Zhang Zhaohe Accounting 8.96 Green

Template Sorted A by Department:

Fitzgerald Matthew Accounting 8.96 Green

McGowen Andrew Accounting 11.31 Blue

Zhang Zhaohe Accounting 8.96 Green

Celaya Esteban Accounting 11.11 Blue

Nienberg Dylan Computer\_Services 18.24 Blue

Welch Wade Computer\_Services 22.01 Blue

Feuling Matthew Computer\_Services 15.21 Green

Pike Jeremy Computer\_Services 22.01 Blue

Pike Jeremy Computer\_Services 22.01 Blue

Pennington Jeremy Customer\_Service 78.93 Blue

Brook Tyler Hacker 45.44 Green

Curtis Harrison Hacker 78.33 Hazel

Gazeau Valentin Management 6.72 Steel

Silva Julian Management 17.35 Gray

Brosh Domenic Management 6.72 Steel

Al-Husain Thamer Manufacturing 12.22 Blue

Hester Travis Manufacturing 1.34 Blue

DeLuna-Solis Eduardo Sales 9.57 Hazel

Moreno Issac Sales 16.27 Hazel

Fruci Jordan Sales 19.26 Blue

Howell Brandon Sales 30.00 Hazel

Oliver Logan Sales 16.27 Hazel

Horn John Sales 19.26 Blue

Harding Andrew Sales 16.37 Gray

Fruci Jordan Sales 19.26 Blue

Kavanaugh Sarah Shipping 3.22 Steel

Jacobson Jake Water\_Sports 53.34 Brown

Dominguez Jared Water\_Sports 53.34 Brown