**ENGR 476 ENGINEERING Lab**

**Lab - 1**

**Student name: Anish Kumaramangalam**

**Section: 01**

**Date: Feb. 08, 2016**

**ENGR 476 : computer Communications Networks**

**Lab: C Program**

Code:

/////////////////////////////////////////////////////////////////////////////////

//

// Title: Asign\_1.c

// Problem: this program take a file with student info and tell the user:

// 1) checks who gets a diploma.

// 2) sorts subject A in ascending order.

// 3) Calculate the average and standard deviation of each subject.

// 4) copies to an external file.

// Class: ENGR 476

// Date: 01/29/2016

// Author: Anish Kumaramagalam

//

/////////////////////////////////////////////////////////////////////////////////

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

/\* function init \*/

double getAverage(double sub[], int size);

double getSd(double sub[], double avg, int size);

void extract\_once(char file\_name[]);

int process\_once();

void sort(int row);

void print\_once(int row, int passCounter, double avgA, double avgB, char passed[][50]);

/\* global var \*/

const int limit = 8;// WARNING! this program will only host limit number of students + 1 (heading).

char data[limit][50]; // stores file locally

char passed[limit][50]; // people who passed

double subA[limit-1]; // has all Subject A grades.

double subB[limit-1]; // has all Subject B grades.

int passCounter = 1; // counts people who pass

/\* main \*/

int main() {

char file\_name[20]; // file name

int row; // counts the number of rows

double avgA, avgB; // average values

char choce[1]; //

while (1) {

/\* clear old data \*/

memset(file\_name, 0, sizeof file\_name);

memset(subA, 0, sizeof subA);

memset(subB, 0, sizeof subB);

passCounter = 1;

row = 0, avgA = 0, avgB = 0;

for (int i = 0; i < limit; i++) { // clears all old cash

memset(passed[i], 0, sizeof passed[i]);

memset(data[i], 0, sizeof data[i]);

}

printf("\nPlease enter the student info file you wish to evaluate?\n");

scanf("%s", file\_name); // takes file name

extract\_once(file\_name); // will extract file validity

row = process\_once(); // split people who passed and sub A and B into separate double array

sort(row); // will sort the data sheet

avgA = getAverage(subA, row); // finds average

avgB = getAverage(subB, row);

print\_once(row, passCounter, avgA, avgB, passed);

/\* options \*/

printf("do you want to run another file type 'y' else 'n' to exit\n");

scanf("%s", choce);

if (choce[0] == 'n') {

break;

}

}

return 0;

} // end main

/\* average computer \*/

double getAverage(double sub[], int size) {

int i;

double avg = 0.0;

double sum = 0.0;

for (i = 1; i < size; ++i)

sum += sub[i];

avg = sum / (size - 1);// because array has students plus one correction.

return avg;

} //end avg

/\* standard deviation computer \*/

double getSd(double sub[], double avg, int size) {

double sd = 0.0;

int i;

for(i = 1; i < size; ++i)

sd += ( (sub[i] - avg) \* (sub[i] - avg) );

return sqrt(sd / size - 2);// because array has students plus one correction.

} //end sd

/\* extracting file function \*/

void extract\_once(char file\_name[]) {

char ch;

int row = 0; // counter

int col = 0; // counter

FILE \*fp; // file

fp = fopen(file\_name,"r"); // opens file read mode

if( fp == NULL ) { // checks if fill exists and has anything

perror("Error while opening the file.\n");

exit(EXIT\_FAILURE);

}

printf("The contents of %s file are :\n\n", file\_name); //test

while(1) { //store data locally at run time by char

ch = fgetc(fp);

if (ch == '\n') {

data[row][col] = ch;

row++;

col = 0;

if (row == limit) break; // brakes out of loop when limit is exceeded

} else {

if (ch == ' ') {

data[row][col] = '\t';

if (row > 0) {

col++;

data[row][col] = '\t';

}

} else {

data[row][col] = ch;

}

col++;

}

if (ch == EOF ) break; // brakes out of loop when their is nothing more to process

}

fclose(fp); // closing file

}// end of extracing\_once

/\* spalit people who passed and sub A and B into separate double array \*/

int process\_once() {

int row = 0, col = 0; // counters

char tempA[5], tempB[5]; // stores tempruary grade.

int count = 0; // counts the number of ' '.

int st = 0; // count the digits of the number.

int tabCheck = 0; // counts tabs

strcpy(passed[0], data[0]); // copying header

while (1) { // local storage processing

if (data[row][col] == '\n') {

row++;

if (row > 1) { // resat, load grades and pass people

subA[row-1] = atof (tempA); // coverts string to float

subB[row-1] = atof (tempB);

if ( subA[row-1] >= 50 && subB[row-1] >= 50 ) {// critria to pass

strcpy(passed[passCounter], data[row-1]); // copying people who pass

passCounter++; // counts passing people

}

/\* clearing temp data \*/

memset(tempA, 0, sizeof tempA);

memset(tempB, 0, sizeof tempB);

}

if (row == limit) break; // brakes out of loop when limit is exeaded

col = 0;

count = 0;

} else {

if (data[row][col] == '\t') { // tab tokens

if (tabCheck == 0) { // this is to add tabs

tabCheck++;

count++;

st = 0;

} else {

tabCheck--;

}

} else {

if (row > 0) { // spliting grades of subject A and B

if (count == 2) { // subject A

tempA[st] = data[row][col];

st++;

} else if (count == 3) { // subject B

tempB[st] = data[row][col];

st++;

}

}

}

col++;

}

if (data[row][col] == EOF ) break; // brakes out of loop when thier is nothing more to process

}

return row;

} //end of void extract\_file

/\* Select sort \*/

void sort(int row) {

double tempNum = 0; // temp

char tempStr[50]; // temp

for(int i = 1; i < row-1; i++)

for(int j = i+1; j < row ; j++)

if( subA[i] > subA[j] ) { // sorts according to subject A in ascending order

memset(tempStr, 0, sizeof tempStr);

tempNum = subA[i];

strcpy(tempStr, data[i]);

memset(data[i], 0, sizeof data[i]);

subA[i] = subA[j];

strcpy(data[i], data[j]);

memset(data[j], 0, sizeof data[j]);

subA[j] = tempNum;

strcpy(data[j], tempStr);

}

}// end of select sort

/\* prints output on file and terminal \*/

void print\_once(int row, int passCounter, double avgA, double avgB, char passed[][50]) {

FILE \*fp;

fp = fopen( "output.txt", "w"); // writing to file output.txt

if (fp == NULL) {

printf("Error opening file!\n");

exit(1);

}

printf("\npeople below you get a diploma :\n\n");

fprintf(fp,"\npeople below you get a diploma :\n\n");

for (int i = 0; i <= passCounter; i++) {

printf("%s", passed[i]);

fprintf(fp,"%s", passed[i]);

}

printf("\nThe original file sorted in ascending order according to subject A below :\n\n");

fprintf(fp,"\nThe original file sorted in ascending order according to subject A below :\n\n");

for (int i = 0; i <= row-1; i++) {

printf("%s", data[i]);

fprintf(fp,"%s", data[i]);

}

printf("\navrage of subject A is %f\n", avgA);

printf("avrage of subject B is %f\n\n", avgB);

printf("\nSD of subject A is %f\n", getSd(subA, avgA, row));

printf("SD of subject B is %f\n\n", getSd(subB, avgB, row));

fprintf(fp,"\naverage of subject A is %f\n", avgA);

fprintf(fp,"average of subject B is %f\n\n", avgB);

fprintf(fp,"\nSD of subject A is %f\n", getSd(subA, avgA, row));

fprintf(fp,"SD of subject B is %f\n\n", getSd(subB, avgB, row));

printf("everything above has been copied to output.txt\n\n");

fclose(fp); // closing file

}

Output:

sfs-wifi-dhcp-10-143-0-218:Desktop anishkumaramangalam$ ./a

Please enter the student info file you wish to evaluate?

std.data

The contents of std.data file are :

people below you get a diploma :

STUDENT\_NAME STUDENT\_NO. SUBJECT\_A SUBJECT\_B

JOAN 1 70.5 85

TOM 3 53 54

SUSAN 5 89 90

KATHY 6 99 55

The original file sorted in ascending order according to subject A below :

STUDENT\_NAME STUDENT\_NO. SUBJECT\_A SUBJECT\_B

RAYMOND 7 22.5 75

TANIA 2 49 75

TOM 3 53 54

JOAN 1 70.5 85

JEFF 4 80 49.5

SUSAN 5 89 90

KATHY 6 99 55

average of subject A is 66.142857

average of subject B is 69.071429

SD of subject A is 22.826402

SD of subject B is 13.938321

everything above has been copied to output.txt

do you want to run another file type 'y' else 'n' to exit