Christiaan Cronje

Lab 2 Option A

COSC 2347 8AM

Due: March 31, 2017

Submitted: March 31, 2017

**controlbreak.c**

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

void NewPage(FILE \*p\_file, float \*p\_pgTotals[4], int \*p\_pgKnt, float \*p\_totals[4], int \*p\_lnKnt);

void PrintOpening(FILE \*p\_file, int \*p\_pgKnt);

void PrintClosing(FILE \*p\_file, float \*p\_pgTotals[3], int \*p\_pgKnt, float \*p\_totals[3], int numRecords);

void PrintDetailHeaders(FILE \*p\_file);

float CalculateGross(float p\_rate, float p\_hrs);

float CalculateFICA(float p\_fica, float p\_limit, float p\_ytd, float p\_gross);

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];

FILE \*fileFica;

char fileFicaName[15];

const int strw15 = 11, strw12 = 12, maxLines = 4;

float currFica = 0.071;

float ficaLimit = 110100;

// check if command line arguments were used for file names

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);

}

// get fica info file name and try to open it

printf("%s", "Enter the FICA Info file name: ");

scanf("%s", fileFicaName);

fileFica = fopen(fileFicaName, "r");

if( fileFica == NULL ){

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

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

}

// get the fica rate

fscanf(fileFica, "%f", &currFica);

// get the fica limit

fscanf(fileFica, "%f", &ficaLimit);

printf("\n");

// Program Initializations

int numRecords = 0;

int id;

char givenName[10]; //First name

char surname[10]; //Last name

char dept\_old[10]; // previous dept name

char dept[10]; //dept name

float ytd = 0, hoursWkd = 0, payrate = 0, gross = 0, fica = 0, net = 0;

int lineKt = 0; //Line counter.

int pageKt = 0; //Page counter.

float cumGross = 0, cumFica = 0, cumNet = 0; //cumulative totals

float pgGross = 0, pgFica = 0, pgNet = 0; // page totals

float deptGross = 0, deptFica = 0, deptNet = 0; // department totals

// init float arrays to easily keep track of totals when passing between functions

float \*pgTotals[] = { &pgGross, &pgFica, &pgNet };

float \*cumTotals[] = { &cumGross, &cumFica, &cumNet };

PrintOpening(fileOut, &pageKt);

// Obtain information from file and calculate the payroll.

while(!feof(fileIn))

{

//get all info from input file and add to running totals

fscanf(fileIn, "%d", &id);

fscanf(fileIn, "%s", givenName);

fscanf(fileIn, "%s", surname);

fscanf(fileIn, "%s", dept);

fscanf(fileIn, "%f", &ytd);

fscanf(fileIn, "%f", &payrate);

fscanf(fileIn, "%f", &hoursWkd);

// compare dept. names to find dept. end

if(strcmp(dept, dept\_old) && (deptGross != deptFica != deptNet != 0))

{

fprintf(fileOut, "\t\t\t%s\t $%12.2f $%12.2f $%12.2f\n\n", "Department Totals:", deptGross, deptFica, deptNet);

deptGross = deptFica = deptNet = 0;

}

strcpy(dept\_old, dept);

if(lineKt == maxLines)

{

NewPage(fileOut, pgTotals, &pageKt, cumTotals, &lineKt);

}

if(lineKt == 0)

PrintDetailHeaders(fileOut);

gross = CalculateGross(payrate, hoursWkd);

pgGross += gross;

deptGross += gross;

fica = CalculateFICA(currFica, ficaLimit, ytd, gross);

pgFica += fica;

deptFica += fica;

net = gross - fica;

pgNet += net;

deptNet += net;

// Print line to file.

fprintf(fileOut, "%-\*d", 5, id);

fprintf(fileOut, "%-\*s", strw15, givenName);

fprintf(fileOut, "%-\*s", strw15, surname);

fprintf(fileOut, "%-\*s", 7, dept);

fprintf(fileOut, " $%\*.2f", strw12, ytd);

fprintf(fileOut, " $%\*.2f", strw12, gross);

fprintf(fileOut, " $%\*.2f", strw12, fica);

fprintf(fileOut, " $%\*.2f\n", strw12, net);

lineKt++;

numRecords++;

}

fprintf(fileOut, "\t\t\t%s\t $%12.2f $%12.2f $%12.2f\n\n", "Department Totals:", deptGross, deptFica, deptNet);

PrintClosing(fileOut, pgTotals, &pageKt, cumTotals, numRecords);

// Close files

fclose(fileIn);

fclose(fileOut);

fclose(fileFica);

printf("%s\"%s\"\n", "Program has successfully completed and output results in ", fileOutName);

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

}

void NewPage(FILE \*p\_file, float \*p\_pgTotals[3], int \*p\_pgKnt, float \*p\_totals[3], int \*p\_lnKnt)

{

//add cumulative totals

for (int i = 0; i < 3; i++)

{

\*p\_totals[i] += \*p\_pgTotals[i];

}

//create page footer

fprintf(p\_file, "\n\t\t\t\t%s\t $%12.2f $%12.2f $%12.2f\n", "Page Totals:", \*p\_pgTotals[0], \*p\_pgTotals[1], \*p\_pgTotals[2]);

fprintf(p\_file, "\t\t\t\t\t\t\t\t\t\t\t%s%d", "Page ", \*p\_pgKnt);

fprintf(p\_file, "\f");

\*p\_pgKnt += 1;

\*p\_lnKnt = 0;

//reset page totals

for (int i = 0; i < 3; i++)

{

\*p\_pgTotals[i] = 0;

}

// new page with header info

// fprintf(p\_file, "%s", "\n\n\t\t\t\t\tACME Sports Inc.\n\n\n");

// fprintf(p\_file, "%s", "Emp# Given Surname Dept. YTD Gross FICA Net\n\n");

}

// title page and first detail headers

void PrintOpening(FILE \*p\_file, int \*p\_pgKnt)

{

fprintf(p\_file, "%s", "\n\n\n\n\t\t\t\t\tACME Sports Inc.\n");

fprintf(p\_file, "%s", "\n\n\t\t\t\"We are the best, you use the best!\"\n");

fprintf(p\_file, "\f");

\*p\_pgKnt += 1;

// fprintf(p\_file, "%s", "\n\n\t\t\t\t\tACME Sports Inc.\n\n\n");

// fprintf(p\_file, "%s", "Emp# Given Surname Dept. YTD Gross FICA Net\n\n");

}

void PrintClosing(FILE \*p\_file, float \*p\_pgTotals[3], int \*p\_pgKnt, float \*p\_totals[3], int p\_records)

{

if(\*p\_pgTotals[0] != \*p\_pgTotals[1] != \*p\_pgTotals[2] != 0)

{

// add totals

for (int i = 0; i < 3; i++)

{

\*p\_totals[i] += \*p\_pgTotals[i];

}

// print final page footer

fprintf(p\_file, "\n\t\t\t\t%s\t $%12.2f $%12.2f $%12.2f\n", "Page Totals:", \*p\_pgTotals[0], \*p\_pgTotals[1], \*p\_pgTotals[2]);

fprintf(p\_file, "\t\t\t\t\t\t\t\t\t\t\t%s%d", "Page ", \*p\_pgKnt);

}

fprintf(p\_file, "\f");

//reset page totals

for (int i = 0; i < 3; i++)

{

\*p\_pgTotals[i] = 0;

}

// new page and print summary text

\*p\_pgKnt += 1;

fprintf(p\_file, "%s", "\n\n\t\t\t\t\tReport Summary.\n\n\n");

fprintf(p\_file, "%s%d\n", "Records Processed: ", p\_records);

fprintf(p\_file, "%-15s$%12.2f\n", "Total Gross: ", \*p\_totals[0]);

fprintf(p\_file, "%-15s$%12.2f\n", "Total Net: ", \*p\_totals[1]);

fprintf(p\_file, "%-15s$%12.2f\n", "Total FICA: ", \*p\_totals[2]);

fprintf(p\_file, "\t\t\t\t\t\t\t\t\t\t\t%s%d", "Page ", \*p\_pgKnt);

}

void PrintDetailHeaders(FILE \*p\_file)

{

fprintf(p\_file, "%s", "\n\n\t\t\t\t\tACME Sports Inc.\n\n\n");

fprintf(p\_file, "%s", "Emp# Given Surname Dept. YTD Gross FICA Net\n\n");

}

float CalculateGross(float p\_rate, float p\_hrs)

{

float total = 0;

//pay for time

total += p\_rate \* p\_hrs;

// plus half

if(p\_hrs > 40)

total += (p\_hrs - 40) \* p\_rate \* 0.5;

return total;

}

float CalculateFICA(float p\_fica, float p\_limit, float p\_ytd, float p\_gross)

{

float owedTax = 0;

if (p\_ytd < p\_limit){

if (p\_ytd + p\_gross <= p\_limit){

//tax = total gross

owedTax = p\_gross \* p\_fica;

}

else{

//tax = 110100 - ytd

owedTax = (p\_limit - p\_ytd) \* p\_fica;

}

}

//else no tax

return owedTax;

}

**compile.sh (C option script)**

#!/bin/bash

gcc -o ControlBreak controlbreak.c

**Makefile (B option)**

default: controlbreak

controlbreak.o: controlbreak.c

gcc -c controlbreak.c -o controlbreak.o

controlbreak: controlbreak.o

gcc controlbreak.o -o ControlBreak

run: default

./ControlBreak

**input.txt**

6 Byrd Wesley Acct 110100.00 13.24 40.0

10 Cauble Blain Acct 66600.00 17.87 64.5

22 Cronje Chrisriaan Acct 36600.00 10.00 10.0

24 Darosa Mauro Acct 110090.00 41.3 36.7

13 Deal Ryan Acct 1358.34 16.22 53.51

5 SDiaz Ortiz DP 110090.00 10.00 1.0

3 Dominguez Michael Mgt 673478.34 187.56 40.0

1 Duhart Marcus Mgt 134234.34 67.42 56.25

4 Dunn Korbin Sales 11345.22 23.77 67.3

12 Evans Jeramy Sales 234.56 17.56 38.9

17 Friedman Micah Sales 2342.34 27.86 45.6

7 Gauss Addison Sales 67000.00 34.23 40.0

86 Giammaruti Alexander Inven 12345.78 13.24 40.0

66 Harrison Jeremy Inven 109998.75 57.87 64.5

51 Hedreen Robert Inven 206600.00 10.00 10.0

38 Jaiswar Rahul Inven 78342.00 51.3 36.75

**fica.txt**

0.071

110100