#include <stdio.h>

#include <stdlib.h>

//Franklyn Gonzalez, last edited 02/28/2019

#define CLS system("cls")

#define pause system("pause")

//records date information

typedef struct {

int day;

int month;

int year;

} date;

//records payroll employee information

typedef struct {

char name[100];

int age;

float hrWage;

float hrsWorked;

float normPay;

float overtimePay;

float totalPay;

date payDate;

} payrollRecord;

//records used to keep track of multiple employee records, up to 100

payrollRecord records[100];

int recordCount = 0;

void printMenu(); //print menu

void addRecord(); //add employee payroll record

void searchRecords(); //identify employee record by date

void sumRecords(); //sum of all payrolls and hours equal to the total

main() {

char option;

//do-while used for menu

do {

printMenu();

option = getchar();

getchar();

if (option == 'A') {

CLS;

addRecord();

}

else if (option == 'B') {

CLS;

searchRecords();

}

else if (option == 'C') {

CLS;

sumRecords();

}

else if (option == 'D') {

break;

}

else {

CLS;

printf("Invalid option!\n\a");

}

} while (option != 'D');

}

//main menu to the user

void printMenu() {

printf("Payroll v1.0\n\n");

printf("Please select one of the options. \n\n");

printf("A) Enter an employee payroll record\n");

printf("B) Find employee's payroll with the date search engine\n");

printf("C) Calculate total hours and funds from all records\n");

printf("D) Exit\n\n");

printf("Your Input (i.e. 'A') : ");

}

//add to the user's payroll record for "records" array

void addRecord() {

//fetch employee information, age, salary

printf("Please, enter the employee's name: ");

gets(records[recordCount].name);

printf("\n\nNow, enter %s's age: ", records[recordCount].name);

scanf\_s(" %d", &records[recordCount].age);

printf("\n\nEnter their hourly salary: ");

scanf\_s(" %f", &records[recordCount].hrWage);

printf("\n\nEnter the amount of hours that %s worked during this pay period: ", records[recordCount].name);

scanf\_s(" %f", &records[recordCount].hrsWorked);

//verify dates

do {

printf("\n\nEnter the month of the pay date (between 1 and 12): ");

scanf\_s(" %d", &records[recordCount].payDate.month);

} while (records[recordCount].payDate.month > 12 || records[recordCount].payDate.month < 0);

do {

printf("\n\nEnter the day of the payroll date (between 1 and 31): ");

scanf\_s(" %d", &records[recordCount].payDate.day);

} while (records[recordCount].payDate.day > 31 || records[recordCount].payDate.day < 0);

do {

printf("\n\nEnter a year between 1990 and 2019: ");

scanf\_s(" %d", &records[recordCount].payDate.year);

getchar();

} while (records[recordCount].payDate.year > 2019 || records[recordCount].payDate.year < 1990);

if (records[recordCount].hrsWorked > 40) {

records[recordCount].overtimePay = (records[recordCount].hrsWorked - 40) \* records[recordCount].hrWage \* 1.5f;

records[recordCount].normPay = records[recordCount].hrWage \* 40;

}

else {

records[recordCount].overtimePay = 0;

records[recordCount].normPay = records[recordCount].hrsWorked \* records[recordCount].hrWage;

}

//The total pay is a sum of the normal pay plus the overtime pay.

records[recordCount].totalPay = records[recordCount].normPay + records[recordCount].overtimePay;

//for each new record, recordCount adds one

recordCount++;

CLS;

printf("Payroll record successfully added!\n\n\n");

}

void searchRecords() {

date userDate;

//fetch employee payroll date to verify if it is a real date

do {

printf("Enter the month of desired pay date: ");

scanf\_s("%d", &userDate.month);

} while (userDate.month > 12 || userDate.month < 0);

do {

printf("\n\nEnter the day of the month: ");

scanf\_s("%d", &userDate.day);

} while (userDate.day > 31 || userDate.day < 0);

do {

printf("\n\nEnter the year: ");

scanf\_s("%d", &userDate.year);

getchar();

} while (userDate.year > 2019 || userDate.year < 1990);

CLS;

printf("Searching for records from %d/%d/%d\n\n", userDate.month, userDate.day, userDate.year);

//matches payDate with employee's payroll

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

if (records[i].payDate.day == userDate.day && records[i].payDate.month == userDate.month

&& records[i].payDate.year == userDate.year) {

printf("Name: %s\n", records[i].name);

printf("Age: %d\n", records[i].age);

printf("Hourly wage: $%.2f/hr\n", records[i].hrWage);

printf("Hours worked: %.2f\n", records[i].hrsWorked);

printf("Regular pay: $%.2f\n", records[i].normPay);

printf("OT pay: $%.2f\n", records[i].overtimePay);

printf("Total pay: $%.2f\n\n", records[i].totalPay);

}

}

pause;

CLS;

}

//runs through 'for' loop to calculate employee records

void sumRecords() {

float totalHrsWorked = 0;

float totalNormPay = 0;

float totalOtPay = 0;

float totalTotalPay = 0;

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

totalHrsWorked += records[i].hrsWorked;

totalNormPay += records[i].normPay;

totalOtPay += records[i].overtimePay;

totalTotalPay += records[i].totalPay;

}

printf("Total payroll figures:\n\n");

printf("Hours worked: %.2f\n", totalHrsWorked);

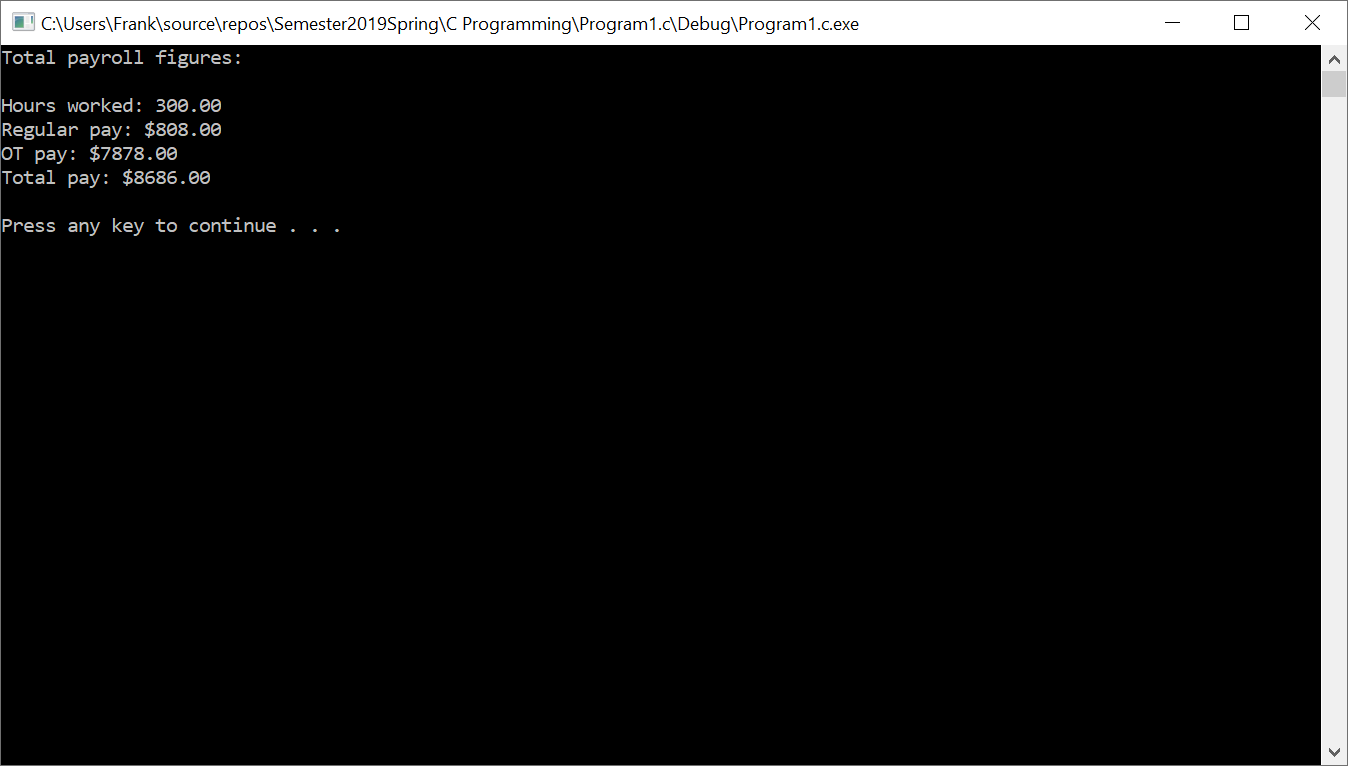
printf("Regular pay: $%.2f\n", totalNormPay);

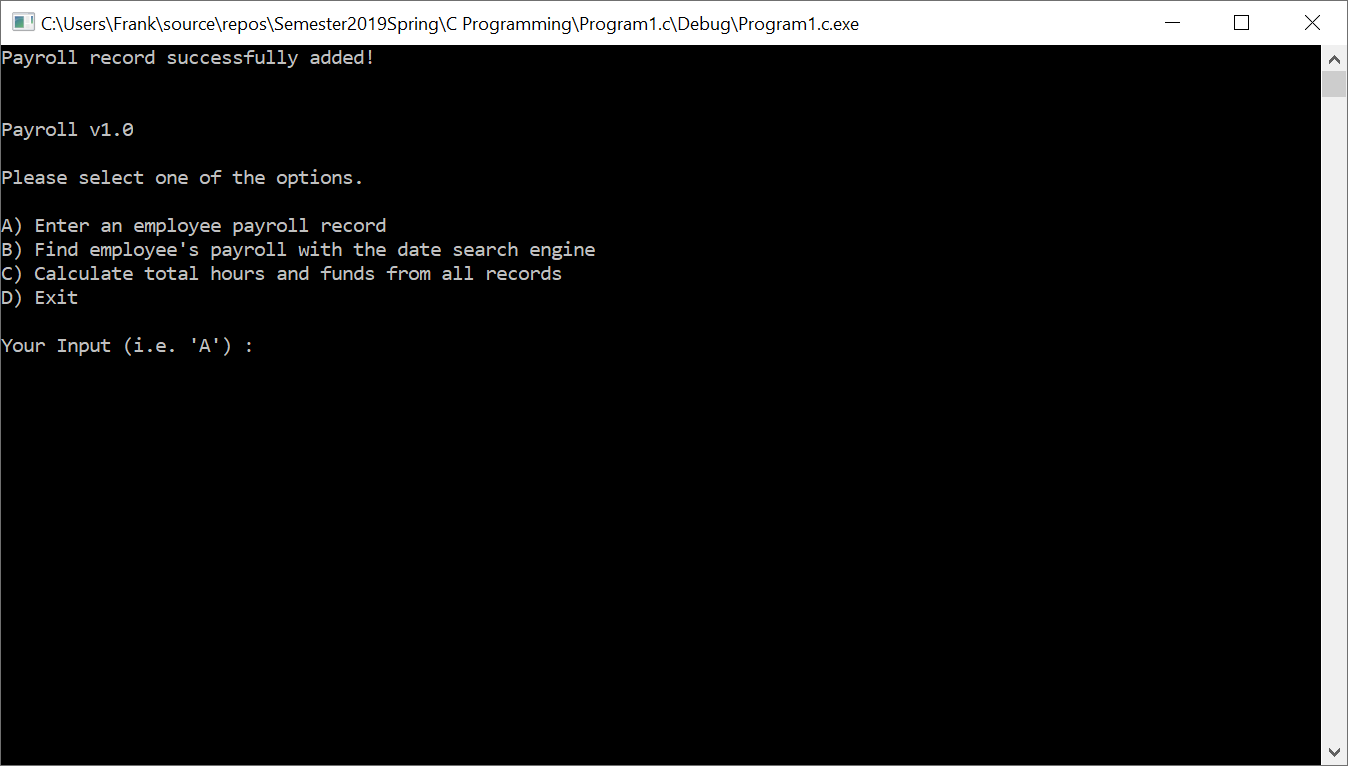
printf("OT pay: $%.2f\n", totalOtPay);

printf("Total pay: $%.2f\n\n", totalTotalPay);

pause;

CLS;

}



|  |  |  |
| --- | --- | --- |
| **Input**  Day  Month  Year  Name[100]  Age  hrWage  hrsWorked  normPay  overtimePay  totalPay  payDate  printmenu  addRecord  searchRecords  sumRecords  option  recordCount  userDate  totalHrsWorked  totalNormPay  totalOtPay  totalTotalPay  records[100] | **Processing**  🡪 takes user input for days in the month🡪  🡪takes user input for months in a year🡪  🡪takes user input for a value from 1990-2019🡪  🡪takes employee’s name🡪  🡪takes the age of the employee🡪  🡪takes the hourly wage that an employee makes🡪  🡪takes the hours that an employee has worked🡪  🡪takes the normal wage of an employee🡪  🡪takes the overtime pay of an employee🡪  🡪takes the total pay that an employee has earned🡪  🡪takes the payroll date from the employee🡪  🡪calls function to printmenu for Payroll v1.0🡪  🡪calls function adds record for employee payroll🡪  🡪calls function to search records of employee🡪  🡪calls function to add the payroll information, such as pay, hours, and overtime for every employee in database  🡪takes option from user🡪  🡪counts amount of records that have been recorded by the program🡪  🡪takes the employee date to verify if date is valid🡪  🡪adds all hours worked by every employee🡪  🡪adds all normal pay by every employee🡪  🡪adds all overtime pay by every employee🡪  🡪adds all pay in total by every employee🡪  🡪adds to database of records🡪 | **Output**  Day of the employee’s payroll  Month of the employee’s payroll  Year of the employee’s payroll  Name of the employee  Age of the employee  Hourly wage by the employee  Hours worked by employee  Normal wage by employee  Overtime pay by employee  Total pay by employee  Payroll date format  Prints menu options  Requests user input for employee information  Runs through all records recorded by program  Adds information by all employees  Displays one of four menu options  Records records by program  Checks to verify employee date  Displays all hours in every record  Displays all normal pay in every record  Displays all overtime pay in every record  Displays total pay  Displays employee payroll record |