**(1)**

/\* This program accepts positive integers from a user until 0 is entered.

After input terminates, the program displays the total number of even

integers recieved, the average value of the even integers, the total

number of odd integers, and the average value of the odd integers. It

then compares the two averages to display the winner (odd, even, or tie). \*/

// Written by: Sam Graham

// Date: 10/23/2018

// Program Name: Odd Or Even Game

#include <stdio.h>

#define QUIT 0

int main (void)

/\* This method is responsible for obtaining the user input. It cycles through

a loop confiming the user gives a positive integer each time. The totals

for odd and even are increased as it goes. The counters are used to provide

for the print statements and to compute averages when the loop ends.

The user ends the loop by pressing 0. \*/

{

int userInput;

int evenCounter = 0;

int oddCounter = 0;

double oddTotal = 0.0;

double evenTotal = 0.0;

printf("Enter your first positive integer (0 to quit): ");

scanf("%d", &userInput);

while (userInput != QUIT)

{

if (userInput < 0)

{ // checking for positive integer.

printf("%d is not a positive integer. Try again: ", userInput);

userInput = 1;

continue;

}

if (userInput % 2 == 0)

{

evenTotal += userInput;

evenCounter++;

printf ("Please give me another interger (0 to quit): ");

scanf ("%d", &userInput);

}

else

{

oddTotal += userInput;

oddCounter++;

printf ("Please give me another interger (0 to quit): ");

scanf ("%d", &userInput);

}

}

printf("%d odd numbers input and average %0.2lf \n", oddCounter, oddCounter == 0 ? 0.0 : oddTotal / oddCounter);

printf("%d even numbers input and average %0.2lf \n", evenCounter, evenCounter == 0 ? 0.0 : evenTotal / evenCounter);

if ((oddTotal / oddCounter) == (evenTotal / evenCounter))

printf("It's a Tie!");

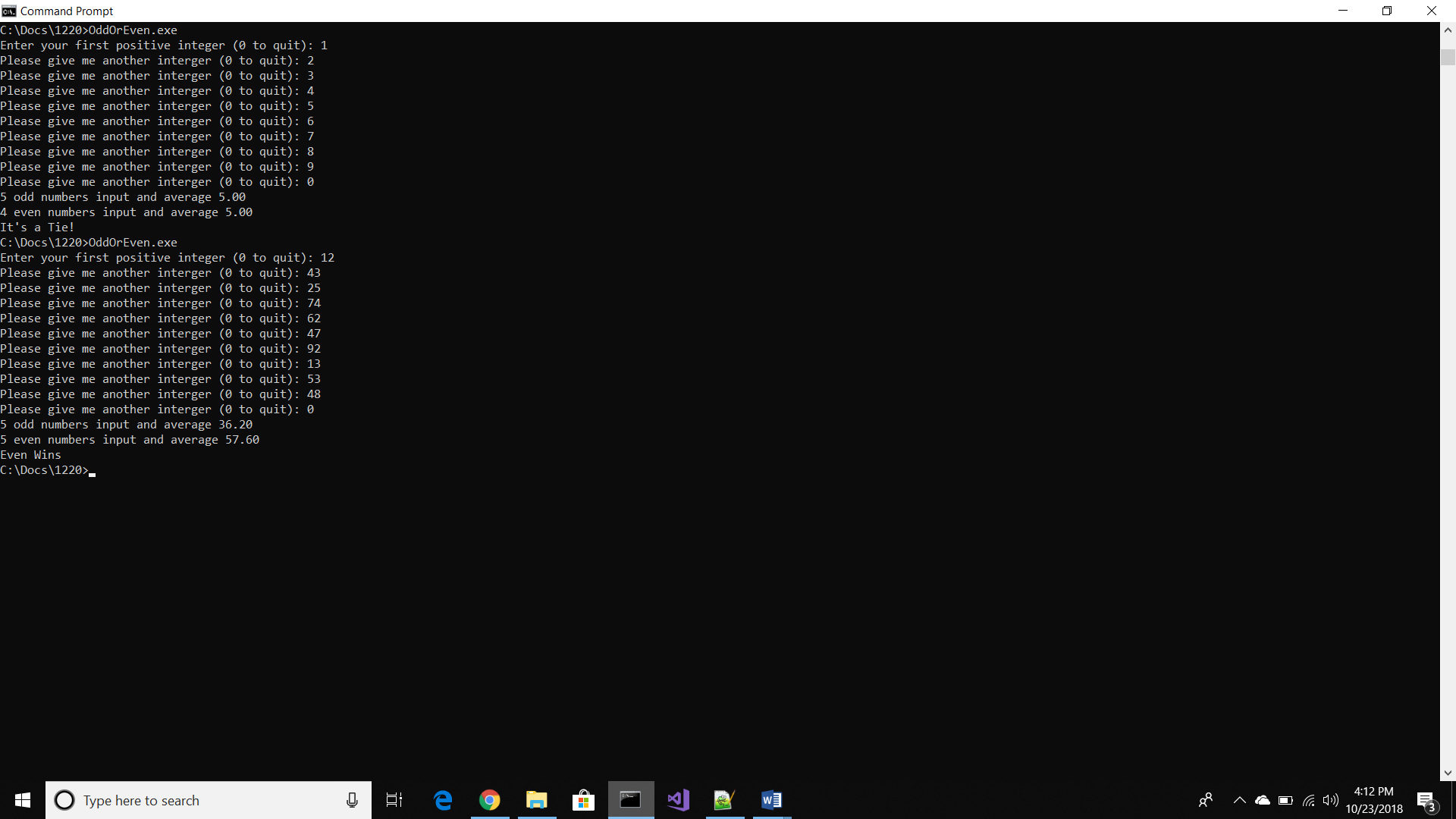
else

(oddTotal / oddCounter) > (evenTotal / evenCounter) ? printf("Odd Wins") : printf("Even Wins");

return 0;

}

**Output:**



**(2)**

/\* This program requests the hours worked in a week then diplays

the gross pay, taxes, and net pay. \*/

// Written By: Sam Graham

// Date: 10/23/2018

// Program Name: Pay Calculator

#include <stdio.h>

#define TAXRATE1 0.18 // rate for first $300

#define TAXRATE2 0.20 // rate for next $150

#define TAXRATE3 0.23 // rate for over the rest

#define TAXBREAK1 300.0

#define TAXBREAK2 (TAXBREAK1 + 150.0)

#define BASEPAY 14.00

#define OVERTIMERATE (BASEPAY \* 1.5)

#define NORMALWEEK 40.0

int main(void)

/\* This method asks the user to input the number of hours worked. It

calculates gross pay, adjust tax withholdings, and displays net pay.

Tax rate for the first $300 is 18%, the next $150 is 20%, then 23%

for the rest \*/

{

float hoursWorked;

float grossPay;

float netPay;

float taxWithHolding;

printf("Input number of hours worked this week: ");

scanf("%f", &hoursWorked);

float overTimePay = (hoursWorked - NORMALWEEK) \* OVERTIMERATE;

if (hoursWorked > NORMALWEEK)

{

grossPay = (BASEPAY \* NORMALWEEK) + overTimePay;

}

else

{

grossPay = BASEPAY \* hoursWorked;

}

if (grossPay <= TAXBREAK1)

netPay = grossPay \* TAXRATE1; // $300 or less

else if (grossPay <= TAXBREAK2)

netPay = grossPay - ((TAXRATE1 \* TAXBREAK1) + (TAXRATE2 \* (grossPay-TAXBREAK1))); // $450 or less

else

netPay = grossPay - ((TAXRATE1 \* TAXBREAK1) + (TAXRATE2 \* TAXBREAK2) + (TAXRATE3 \* (grossPay - TAXBREAK2))); // above $450

taxWithHolding = grossPay - netPay;

printf("The gross pay for %.2f hours is $%.2f. The net pay after $%.2f in taxes is $%.2f", hoursWorked, grossPay, taxWithHolding, netPay);

return 0;

}

**Output:**

