**Assignment 3 (due date: 10/14/2020)**

In this assignment you will ask to do coding in C using functions and iteration (compile and run). It will be an excellent prepration for your midterm exam. Please be descriptive and comprehensive and avoid short and ambiguous answers. Answers that are ambiguous will reduce your grades.

Please submit your assignment into Canvas. Please submit the codes as .c files (ready to compile).

1. Identify and correct the errors in each of the following statements:

a) **for** (a = **25**, a <= **1**, a--); {

printf(**"%d\n"**, a);

}

semicolon before the bracket must be removed, and the loop is never entered as a is never less than or equal to 1. So we change ‘<=’ to ‘>=’:

for(int a = 25; a >= 1; a--){

printf("%d\n", a);

}

b) The following code should print whether a given integer is odd or even:

**switch** (value) {

**case** (value % **2** == **0**):

puts(**"Even integer"**);

**case** (value % **2** != **0**):

puts(**"Odd integer"**);

}

In C cases must be compile time constants so we cannot use (value % 2 == 0) as that will have to be evaluated at run time. We must use an if else statement to accomplish this:

int value = 3;

if (value % 2 == 0)

puts("Even integer");

else if (value % 2 != 0)

puts("Odd integer");

c) The following code should calculate incremented salary after 10 years:

**for** (**int** year = **1**; year <= **10**; ++year) {

**double** salary += salary \* **0.05**;

}

printf(**"%4u%21.2f\n"**, year, salary);

salary and year should be declared outside of the for statement since they are being used in the printf statement later:

double salary = 30000;

int year;

for (year = 1; year <= 10; ++year) {

salary += salary \* 0.05;

}

printf("%4u%21.2f\n", year, salary);

d) **for** (**double** y = **7.11**; y != **7.20**; y += **.01**)

printf(**"%7.2f\n"**, y);

!= should be <

for (double y = 7.11; y < 7.20; y += .01)

printf("%7.2f\n", y);

e) The following code should output all multiples of 3 from 1 to 100:

**for** (**int** x = **3**; x <= **100**; x%**3** == **0**; x++ ) {

printf(**"%d\n"**, x);

}

We should put x % 3 == 0 in an if statement within the loop:

for (int x = 3; x <= 100; x++ ) {

if (x % 3 == 0)

printf("%d\n", x);

}

2: (Prime Numbers) Write a program to calculate and print a list of all prime numbers from 1 to 100.

See problem2.c attached

3: (Factorials) The factorial function is used frequently in probability problems. The factorial of a positive integer n (written n! and pronounced “n factorial”) is equal to the product of the positive integers from 1 to n. Write a program that evaluates the factorials of the integers from 1 to 5. Print the results in tabular format. What difficulty might prevent you from calculating the factorial of 20?

//Angel Avelar-Bonilla

//10/13/2020

//Prints 1 - 5 factorial

#include <stdio.h>

int factorial(int x)

{

if (x == 1) return 1;

return x \* factorial(x - 1);

}

int main()

{

for (int i = 1; i <= 5; i++){

printf("%d\t%d\n",i,factorial(i));

}

return 0;

}

Or see problem3.c attached

4: (Calculating Credit Limits) Collecting money becomes increasingly difficult during periods of recession, so companies may tighten their credit limits to prevent their accounts receivable (money owed to them) from becoming too large. In response to a prolonged recession, one company has cut its customers’ credit limits in half. Thus, if a particular customer had a credit limit of $2000, it’s now $1000. If a customer had a credit limit of $5000, it’s now $2500. Write a program that analyzes the credit status of three customers of this company. For each customer you’re given:

a) The customer’s account number.

b) The customer’s credit limit before the recession.

c) The customer’s current balance (i.e., the amount the customer owes the company).

Your program should calculate and print the new credit limit for each customer and should

determine (and print) which customers have current balances that exceed their new credit limits.

5: (Fibonacci) The Fibonacci series

0, 1, 1, 2, 3, 5, 8, 13, 21, …

begins with the terms 0 and 1 and has the property that each succeeding term is the sum of the two preceding terms. a) Write a non-recursive function fibonacci(n) that calculates the nth Fibonacci number. Use unsigned int for the function’s parameter and unsigned long long int for its return type. b) Determine the largest Fibonacci number that can be printed on your system.

6. (Sum of Digits) Write a function that takes an integer and returns the sum of its digits. For example, given the number 7631, the function should return 17.