**Assignment 2 (due date: 10/14/2021)**

In this assignment you will ask to do coding in C using function, iteration and logic (compile and run).

Please submit your assignment into Canvas. Please submit the codes as .c files or copy them all into a single word file but make sure that they are ready to be compiled. It is strongly recommended to write these codes in terms of functions! It will be an excellent practice for your midterm (if we write them in terms of functions).

1: (Diamond-Printing Program) Write a program that prints a diamond shape similar to the one shown below with 15 stars in the middle.



2: (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 to 3/4. Thus, if a particular customer had a credit limit of $2000, it’s now $1500. If a customer had a credit limit of $4000, it’s now $3000. 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.

3: ***(De Morgan’s Laws)*** In this chapter, we discussed the logical operators &&, ||, and !. De Morgan’s Laws can sometimes make it more convenient for us to express a logical expression. These laws state that the expression !(*condition1* && *condition2*) is logically equivalent to the expression (!*condition1* || !*condition2*). Also, the expression !(*condition1* || *condition2*) is logically equivalent to the expression (!*condition1* && !*condition2*). Use De Morgan’s Laws to write equivalent expressions for each of the following, and then write a program to show that both the original expression and the new expression in each case are equivalent.

a) !(x < **5**) && !(y >= **7**)

b) !(a == b) || !(g != **5**)

c) !((x <= **8**) && (y > **4**))

d) !((i > **4**) || (j <= **6**))

4. ***(Calculating the Value of*** π***)*** Calculate the value of π from the infinite series:



Print a table that shows the value of π approximated by one term of this series, by two terms, by three terms, and so on. How many terms of this series do you have to use before you first get 3.14?

3.141? 3.1415? 3.14159?

5: ***(Calculating Weekly Pay)*** A company pays its employees as managers (who receive a fixed weekly salary), hourly workers (who receive a fixed hourly wage for up to the first 40 hours they work and “time-and-a-half”—i.e., 1.5 times their hourly wage—for overtime hours worked), commission workers (who receive $350 plus 6.5% of their gross weekly sales), or pieceworkers (who receive a fixed amount of money for each of the items they produce—each pieceworker in this company works on only one type of item). Write a program to compute the weekly pay for each employee. You do not know the number of employees in advance. Each type of employee has its own pay code: Managers have pay code 1, hourly workers have code 2, commission workers have code 3 and pieceworkers have code 4. Use a switch (or nested if-else) to compute each employee’s pay based on that employee’s pay code. You should ask the user (i.e., the payroll clerk) to enter the appropriate facts your program needs to calculate each employee’s pay based on that employee’s pay code. [*Note:* You can input values of type double using the conversion specifier %lf with scanf.]