Assignment 3 (Project) – Due January 27

**Algorithms**

**Description:** The assignment can be done **individually** or in teams of up to **three**. Submit one assignment per team via Omnivox and NOT MIO. Assignments sent via MIO will be deducted marks. Assignments must be done alone or in groups and collaboration between individuals or groups is strictly forbidden. Submissions that are late will receive a 10% penalty per day, and after four days of no submission a mark of zero will be given.

**General Guidelines When Writing Your Programs:**

Include the following in comments at the top of your program:

// -----------------------------------------------------

// Assignment (include number)

// Written by: (include your name(s) and student ID(s))

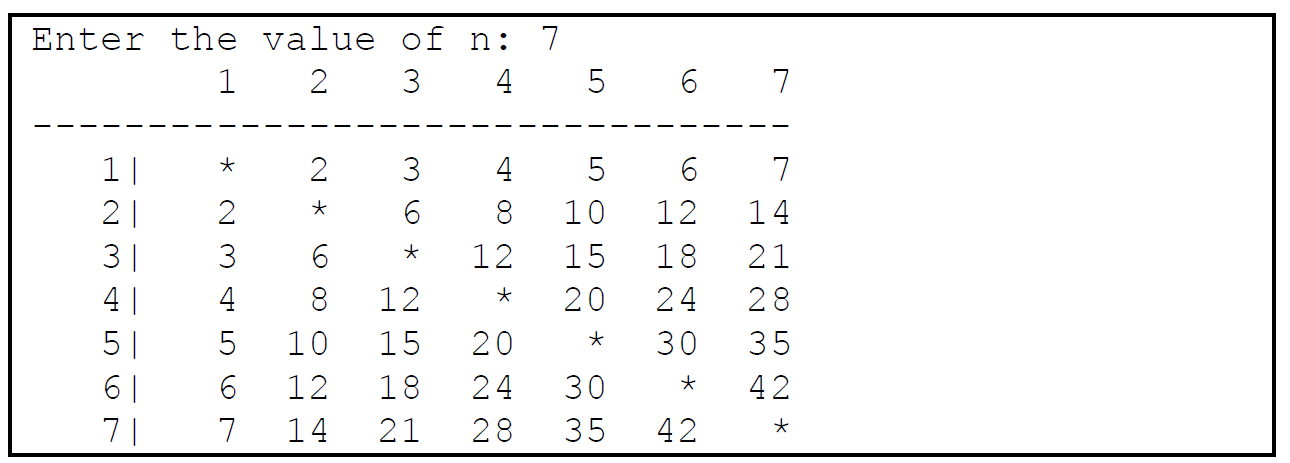
// Short Description of your project/code and how you designed it.

// -----------------------------------------------------

Throughout your program, include comments in your program describing the main steps where necessary.

**QUESTION ONE: Multiplication Table (3 points)**

Write a program to display on the screen a multiplication table from 1 to n; where n is an integer entered by the user. The main diagonal of the table must be filled with the character ‘\*’. For example, if the user enters 7, your program should display:



**QUESTION TWO: Time Algorithm (2 points)**

Write a program that outputs and prints the number of hours, minutes, and seconds that corresponds to 50391 total seconds. The output should be 13 hours, 59 minutes, and 51 seconds. Solve the problem using arithmetic operators only.

**QUESTION THREE: Password Checker (5 points)**

Write a program that incorporates an algorithm with a function that will check whether or not a string is in a valid password format with the following rules:

A password must have at least ten characters.  
A password consists of only letters, digits and symbol(s).  
A password must contain at least two digits.

A password must contain at least one uppercase letter

A password must contain at least one special symbol

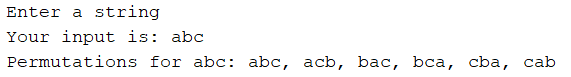
Your program should continue to prompt the user until he/she enters a valid password.

**QUESTION FOUR: Common Substring Algorithm (10 points)**

Write a program that will find the longest common substring from a series of two to many words given to you by the user. For example, your program will ask the user how many words he/she wants to compare, with two being the minimum input. The longest common substring of the strings “Q**WER**TY”, “TE**WER**U”, “Y**WER**QT” is **WER**. Your algorithm should work for any number of inputs (words), if there is no match of common substrings, indicate to the user with a message. A common algorithm to follow when making this program is first to prompt the user for how many words he/she wants to compare (check for valid input of a number 2 to ‘n’) and then prompt the user for the strings (words) and following this return the common substring if one is present.

**QUESTION FIVE: String Permutation (5 points)**

Write a program that uses a recursive method/algorithm to compute all permutations of a string that is unique taken from the user. Permutation is defined as a way or several possibilities in which a number or things can be organized. For example:



**QUESTION SIX: Modified Fibonacci Recursion (10 points)**

Using recursion, write a program that takes an input ‘n’ (a number) from a user to calculate and print out the Fibonacci using the following modified definition:

F(N) = 1 **if n = 1 or n = 2**

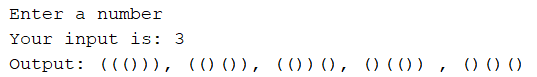
= F((n+1)/2)2 + F((n-1/2)2 **if n is odd**

= F(n/2 + 1)2 – F(n/2 – 1)2 **if n is even**

Your solution must implement recursion to receive points for this question.

**QUESTION Seven: Printing Parenthesis (5 points)**

Write a program that uses a recursive algorithm to print all the valid (properly closed and open brackets) combinations of a number of parentheses taken from the user. For example your program should behave similar to the below:



Submission:

You are allowed to work on a team of 3 students at most (including yourself!). Any teams of 4 or more students will result in 0 marks for all team members. If your work on a team, ONLY one copy of the assignment is to be submitted for both members. Zip together the source codes. (Please use WINZIP or WINRAR).

The zip file should be called a#\_studentID, where # is the number of the assignment studentID is your student ID(s) number. For example, for the first assignment, student 1234567 would submit a zip file named a1\_12345678.zip. If you work on a team and your IDs are 12345678 and 34567890, you would submit a zip file named a1\_1234567\_4567890.zip

Submit only ONE version of an assignment. If more than one version is submitted, the latest (most recent) version submitted will be the only one considered.

**Grade Weight: 40% of your final grade.**