Assignment No. 2
EECS 210
Discrete Structures
Due: 11:59 PM, Thursday, September 14, 2023
Submit deliverables in a single zip file to Canvas
Files in other formats (e.g., .tar) will not be graded
Name of the zip file: FirstnameLastname_Assignment2 (with your first and last name)
Name of the Assignment folder within the zip file: FirstnameLastname_Assignment2

Deliverables:

1. Copy of Rubric2.docx with your name and ID filled out (do not submit a PDF)
2. Source code.
3. Screen print showing the successful execution of your code or copy and paste the output from a console screen to a Word document and PDF it.

Assignment:

- You may use any language you want, but if you want help from me or one of the SIs, you should probably use C++ or Python.
- Create a program for the following:
    1. Prove (true) or disprove (false) the following assertions for the domain of {0,1,2,3,4,5,6,7,8,9,10}. For the universal quantifier, show at least one of the numbers in the domain that disproves the assertion. For the existential quantifier, show at least one number in the domain that proves the assertion:
        a) $\exists x\, P(x)$, where $P(x)$ is the statement "$x<2$"
        b) $\forall x\, P(x)$, where $P(x)$ is the statement "$x<2$"
        c) $\exists x\, (P(x) \vee Q(x))$ where $P(x)$ is the statement "$x<2$" and where $Q(x)$ is the statement "$x>7$"
        d) $\forall x\, (P(x) \vee Q(x))$ where $P(x)$ is the statement "$x<2$" and where $Q(x)$ is the statement "$x>7$"
        e) Prove De Morgan's Law for the Existential Quantifier where $P(x)$ is the statement "$x<5$"
        f) Prove De Morgan's Law for the Universal Quantifier where $P(x)$ is the statement "$x<5$"
    2. Find the following truth values for $P(x,y)$: $x \cdot y = 1$ where the domain of x and y is {1,2,4,5,10,0.5,0.25,0.2,0.1}. Show the values in the domain that either make the assertions true or false.
        a) $\forall x \forall y P(x,y)$
        b) $\forall x \exists y P(x,y)$
        c) $\forall y \exists x P(x,y)$
        d) $\exists x \forall y P(x,y)$
        e) $\exists y \forall x P(x,y)$
        f) $\exists x \exists y P(x,y)$
- Print out a line between each of the above indicating which number your program is answering, (e.g., 1b).
- Remember:

    - To prove ∀ requires showing it is true for ALL values in the domain; to prove ∃ requires showing it is true for only ONE value in the domain; to disprove ∀ requires showing it is false for ONE value in the domain; to disprove ∃ requires showing it is false for ALL values in the domain.
    - To prove Laws, requires showing both sides are the same for the domain (either both TRUE or both FALSE); and given P(x), you have to figure out what ¬P(x) is, e.g., if P(x) = x > y, then ¬P(x) = x ≤ y.
- Provide comments that explain what each line of code is doing. See rubric below.

Rubric for Program Comments		
Exceeds Expectations (90-100%)	**Meets Expectations (80-89%)**	**Unsatisfactory (0-79%)**
Software is adequately commented with prologue comments, comments summarizing major blocks of code, and comments on every line.	Prologue comments are present but missing some items or some major blocks of code are not commented or there are inadequate comments on each line.	Prologue comments are missing all together or there are no comments on major blocks of code or there are very few comments on each line.

Adequate Prologue Comments:

- Name of program contained in the file (e.g., EECS 210 Assignment 3)
- Brief description of the program, e.g.:
    - Python code for demonstrating operations on relations and properties of relations.
- Inputs (e.g., none, for a function, it would be the parameters passed to it)
- Output, e.g.,
    - Print out of the name of each exercise, followed by the exercise's output.
- All collaborators
- Other sources for the code ChatGPT, stackOverflow, etc.
- Author's full name
- Creation date: The date you first create the file, i.e., the date you write this comment

Adequate comments summarizing major blocks of code and comments on every line:

- Provide comments that explain what each line of code is doing.
- You may comment each line of code (e.g., using //) and/or provide a multi-line comment (e.g., using /* and */) that explains what a group of lines does.
- Multi-line comments should be detailed enough that it is clear what each line of code is doing.
- Each block of code must indicate whether you authored the code, you obtained it from one of the sources listed in the prolog, or one of your collaborators authored the code, or if it was a combination of all of these.

Collaboration and other sources for code:

- When you collaborate with other students or use other sources for the code (e.g., ChatGPT, stackOverflow):
    - Your comments must be significantly different from your collaborators.
    - More scrutiny will be applied to grading your comments in particular explaining the code “in your own words”, not the source’s comments (e.g., ChatGPT’s comments).
- Failure to identify collaborators or other sources of code will not only result in a 0 on the assignment but will be considered an act of Academic Misconduct.
- Students who violate conduct policies will be subject to severe penalties, up through and including dismissal from the School of Engineering.