Assignment No. 5
EECS 210
Discrete Structures
Due: 11:59 PM, Thursday, October 26, 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_Assignment5 (with your first and last name)
Name of the Assignment folder within the zip file: FirstnameLastname_Assignment5

Deliverables:

1. Copy of Rubric5.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.
- Part 1:
    - Recall that a function f from A to B, is a relation $f \subseteq A \times B$ such that each element of A is assigned to exactly one element of B.
    - Thus, we can represent a function as a relation of order pairs, (a,b), where $a \in A$, $b \in B$, and $b = f(a)$.
    - Write a program that takes inputs of A, B, and a relation f and a) determines if the relation is a function or not, b) and if it is a function, determines if it is injective, surjective, or bijective, c) and if it is bijective, determines the inverse function.
    - Your program should print out the inputs using the notation you learned in class, i.e., A = {…}, B = {…}, f = {(a,b), …}, whether the relation is a function or not, b) and if it is a function, whether it is injective, surjective, or bijective, c) and if it is bijective, what is the inverse function. Test your program with the following inputs:
        a. A = {a,b,c,d}, B = {v,w,x,y,z}, f = {(a,z),(b,y),(c,x),(d,w)}
        b. A = {a,b,c,d}, B = {x,y,z}, f = {(a,z),(b,y),(c,x),(d,z)}
        c. A = {a,b,c,d}, B = {w,x,y,z}, f = {(a,z),(b,y),(c,x),(d,w)}
        d. A = {a,b,c,d}, B = {1,2,3,4,5}, f = {(a,4),(b,5),(c,1),(d,3)}
        e. A = {a,b,c}, B = {1,2,3,4}, f = {(a,3),(b,4),(c,1)}
        f. A = {a,b,c,d}, B = {1,2,3}, f = {(a,2),(b,1),(c,3),(d,2)}
        g. A = {a,b,c,d}, B = {1,2,3,4}, f = {(a,4),(b,1),(c,3),(d,2)}
        h. A = {a,b,c,d}, B = {1,2,3,4}, f = {(a,2),(b,1),(c,2),(d,3)}
        i. A = {a,b,c}, B = {1,2,3,4}, f = {(a,2),(b,1),(a,4),(c,3)}
- Part 2:
    - Write a program to calculate the greatest common divisor of two integers, gcd(a,b) using the Euclidean Algorithm.

- Your program should show each step of the Euclidean Algorithm using the "remainder" format. For example, the first step for gcd(414, 662) would be: 662/414 = 1 R 248
- Your program should show the final results as follows: gcd(a,b) = n
- Test your program with the following inputs:
    a. gcd(414, 662)
    b. gcd(6, 14)
    c. gcd(24, 36)
    d. gcd(12, 42)
    e. gcd(252, 198)

- Part 3:
    - Write a program to express gcd(a, b) as a linear combination: gcd(a, b) = sa + tb, where s and t are Bézout coefficients of a and b, using the Euclidean Algorithm (Method 1).
    - Your program should show each step of the Euclidean Algorithm using the "product & sum" format. For example, the first step for gcd(252, 198) would be: 252 = 198 * 1 + 54
    - Then, your program should show each step of working backwards through the steps of the Euclidean algorithm in the same way. For example, the "backward" steps for the gcd(252, 198) would be:
        18 = 54 − 1 * 36
        18 = 54 − 1 * (198 − 3 * 54)
        18 = 4 * 54 − 1 * 198
        18 = 4 * (252 − 1 * 198) − 1 * 198
        18 = 4 * 252 − 5 * 198
    - Your program should show the final results as follows:
        gcd(a, b) = s*a + t*b
    - Test your program with the following inputs:
        a. gcd(414, 662)
        b. gcd(6, 14)
        c. gcd(24, 36)
        d. gcd(12, 42)
        e. gcd(252, 198)

- Part 4:
    - Write a program to express gcd(a, b) as a linear combination: gcd(a, b) = sa + tb, where s and t are Bézout coefficients of a and b, using the **extended** Euclidean Algorithm (Method 2).
    - Your program should show the quotients q1 through qj and then the calculations for sj and tj. For example for gcd(252, 198):
        q1 = 1, q2 = 3, q3 = 1, and q4 = 2
        s0 = 1, s1 = 0, s2 = s0 − s1*q1 = 1 − 0 * 1 = 1, etc.
        t0 = 0, t1 = 1, t2 = t0 − t1*q1 = 0 − 1 * 1 = −1, etc.
    - Your program should show the final results as follows:
        gcd(a, b) = s*a + t*b
    - Test your program with the following inputs:

a. gcd(414, 662)
b. gcd(6, 14)
c. gcd(24, 36)
d. gcd(12, 42)
e. gcd(252, 198)

- 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.