Use *Gaussian Elimination* and *Jacobi Iteration* to solve a linear system $Ax = b$. First input the number of equations and unknowns n, then input the augmented matrix in the form of $[A\ b]$ and judge whether there exists a unique solution. If there is a unique solution, let users choose a method between *Gaussian Elimination* and *Jacobi Iteration* to solve linear equations. Otherwise quit the program and prompt "No unique solution exists!". If there are any invalid inputs during the process, for example, non-numerical values, quit the program and prompt "Invalid input!".

Example 1:

$$\begin{aligned} 10x_1 - x_2 + 2x_3 &= 6 \\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25 \\ 2x_1 - x_2 + 10x_3 - x_4 &= -11 \\ 3x_2 - x_3 + 8x_4 &= 15 \end{aligned}$$

The number of equations and unknowns is 4 and the augmented matrix is

$$\tilde{A} = [A\ b] = \begin{bmatrix} 10 & -1 & 2 & 0 & 6 \\ -1 & 11 & -1 & 3 & 25 \\ 2 & -1 & 10 & -1 & -11 \\ 0 & 3 & -1 & 8 & 15 \end{bmatrix}$$

The interaction process of your program should be

- Input the number of equations and unknowns n: 4
- Input the augmented matrix of Ax=b as [A b]:
  10 -1 2 0 6
  -1 11 -1 3 25
  2 -1 10 -1 -11
  0 3 -1 8 15
- [0] Jacobi Iteration [1] Gaussian Elimination
  Choose a method: 1
- Results is:
  1 2 -1 1

Specifically, when using *Jacobi Iteration*, we set the maximum number of iterations equal to 10000 and the stopping criterion is

$$||X^{(k)} - X^{(k-1)}||_\infty < 0.001$$

Let users input the initial approximation $X^{(0)}$ randomly. If linear equations can't be solved within 10000 iterations, prompt "Maximum number of iterations exceeded!". The figures below are two examples.

```
Input the number of equations and unknowns n: 4

Input the augmented matrix of Ax=b as [A b]:
10 -1 2 0 6
-1 11 -1 3 25
2 -1 10 -1 -11
0 3 -1 8 15

[0] Jacobi Iteration  [1] Gaussian Elimination
Choose a method: 1

Results is:
1 2 -1 1
```

Figure 1: Gaussian Elimination

```
Input the number of equations and unknowns n: 4

Input the augmented matrix of Ax=b as [A b]:
10 -1 2 0 6
-1 11 -1 3 25
2 -1 10 -1 -11
0 3 -1 8 15

[0] Jacobi Iteration  [1] Gaussian Elimination
Choose a method: 0

Input the initial approximation x(0):
0 1 2 3

Results is:
0.999845 2.00004 -1.00008 1.00032
```

Figure 2: Jacobi Iteration

Example 2:

$$
\begin{aligned}
x_1 + x_2 + x_3 &= 4 \\
2x_1 + 2x_2 + x_3 &= 6 \\
x_1 + x_2 + 2x_3 &= 6
\end{aligned}
$$

- Input the number of equations and unknowns n: 3
- Input the augmented matrix of Ax=b as [A b]:
  1 1 1 4
  2 2 1 6
  1 1 2 6
- No unique solution exists!

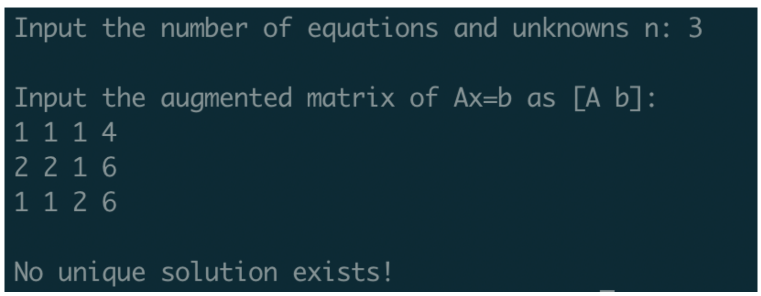

Figure 3: No unique solution exists

Example 3:

- Input the number of equations and unknowns n: q
- Invalid input!

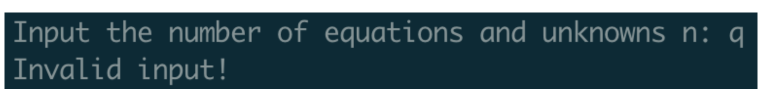

Figure 4: Invalid input

Important Notes:

- Gaussian Elimination is covered by *NumericalAnalysis* in Chapter 6.1
- Jacobi Iteration is covered by *NumericalAnalysis* in Chapter 7.3
- Remember to submit your makefile!
- Due: 2019/10/10 11:59pm
- Happy National day! :)