1. You can use arrays, pointers, vectors, list, NumPy arrays etc.

2. You can’t use any inbuilt linear algebra or linear programming algorithms (not even addition of matrices). You can use other inbuilt algorithms if you need.

3. Do not use platform dependent libraries such as conio.h.

4. Any code that you reuse; make a function for it such as reading or printing arrays and matrices {if you have a print matrix function you can very easily check output at different steps}

Some mistakes from lab1:

Don’t mix GS and combination generator functions. Keep them separate.

1. For every iterative algorithm you have to have an abs error bound epsilon. No of iterations( with sufficient large value atleast 1000 ; not 20 or 50) can be a secondary check for divergence.
2. The error bound has to be checked for each component of x.   
   err[i]=x1[i]-x0[i] and then check if abs(err[i])>epsilon for any i=1,2,...n if false : convergence else: iterate
3. Write comments.
4. The submatrices will be generated by combination ( more specifically by combination of column indices; so it is just combination of sequence 0,1,2,...n ) not permutations.
5. Avoid type conversions.
6. Be very careful with commas in python ex: a=[],b=c
7. Variable length arrays are not supported in C++, some compilers may provide some partial functionalities but don’t use them. Use fixed size arrays or dynamic memory allocation (pointers and vectors) .

What to do in lab 2:

1. fix mistakes from lab1

2. write additional code for lab 2

(optional but still do it) 3. Implement Gauss elimination with partial pivoting (just the forward elimination step). Instead of back substitution pass your matrix to Gauss Seidel. [ to generate row echelon form of your submatrices]

Functions that you may need to write to better organize your code:

1. a GE function (which will create a temp [m][m+1] aux matrix and call forward elimination )

2. utility functions to swap rows and max in a row (may not need depending on your language)

3. write a function to do forward substitution ( this can also test whether the system is solvable or not) ( solvable system can also diverge in Gauss –Seidel)