# Programming Guidelines

1. Use comments to explain your codes
   1.1 Header comments at the beginning of a file
   1.2 Global variables and function declarations need to have comments
   1.3 Key operations must be clearly documented.
   1.4 Spelling must be correct.
   1.5 Comments should also be properly indented and with space char inserted.
2. Indentation to group statements at the same block level
   2.1 Use `<tab>` for indentation.
3. Blank lines to separate
   3.1 directives and functions
   3.2 declarations and statements
   3.3 All declarations must precede statements in a function.
4. Space character to separate tokens
   4.1 The same way as in English sentences
5. Variable name should be descriptive.
   5.1 `i, j, k` for integral local variables
   5.2 `x, y, z` for `floating point` local variables
   5.3 `p, q, r` for local `pointers`
   5.4 `All-capital` tokens for symbolic constants
6. Each line of source code should not have more than 80 characters.

# Example

```
// EE4070 Lab01. Checking eigenvalue and eigenvector
// ID, 姓名
// 2020/03/09

#include <cstdio>
#include <stdlib.h>
#include <math.h>
#include "VEC.h"
#include "MAT.h"

int main(void)
{
    int n;                            // system dimension
    int i, j;                         // i, j: loop indices
    double lambda;                    // eigenvalue
    double z;                         // |A x - lambda x|

    scanf("%d", &n);                  // dimension of the linear system
    MAT A(n);                         // matrix A, uninitialized
    VEC X(n), V(n);                   // X: eigenvector, V: residue

    for (i = 0; i < n; i++) {         // read matrix A from stdin
        for (j = 0; j < n; j++) {
            scanf("%lf", &A[i][j]);
        }
    }
```

# Example, II

```
    for (i = 0; i < n; i++) {         // read vector X from stdin
        scanf("%lf", &X[i]);
    }
    scanf("%lf", &lambda);            // read eigenvalue lambda from stdin

    V = A * X - lambda * X;           // residue equation

    z = 0;
    for (i = 0; i < n; i++) {         // to calculate |V|
        z += V[i] * V[i];
    }
    z = sqrt(z);

    printf("|A x - lambda x| = %g\n", z);

    return 0;
}
```