# Project

## Project Description

It is required to develop a program to solve linear equations. The programs reads first an integer n which is the number of equations.

Then the program reads n lines containing the equations.

For example the input to the program is like:

3
2x1+3x2+4x3=16
1x1+2x2+1x3=8
3x1+1x2+2x3=13

**Equation proper form**: Any operation should first convert every equation to the proper form. The equation proper should have the following properties

1. variables are ordered alphabetically from left to right

   3x2+2x1+4x3=16
   Should be
   2x1+3x2+4x3=16

2. Any variable should appears only one time

   4x1+3x2-2x1+4x3=16
   Should be
   2x1+3x2+4x3=16

3. Only one constant term should appear in the equation and it should be on the right hand side

2x1+3x2+5+4x3-11=10
Should be
2x1+3x2+4x3=16

4. Coefficient when equals to one or -1 the digit 1 is optional

1x1+3x2-1x3=10
Can be input as be
x1+3x2-x3=10

In the program, after reading the equations the program reads one or more commands from user and program should print the results of each operations and read another operation till the program read the operation quit which ends the program. **Any result printed should be based on the proper form of equations.**

| Level | operation | Meaning | Action Required from the Program |
|---|---|---|---|
| 1 | num_vars | Print the number of variables | Print the number of variables in the equations. |
| | equation i | Print equation number i (i is integer) | Print equation of the equation number i |
| | column x2 | Print the column of coefficients of variable (x2) | Print the vector of coefficients of the given variable name. (x2 can be replaced by any variable name) |
| 2 | add 1 3 | Add equation 1 and equation 3 and print the result equation | Print the result equation from adding equation 1 and equation 3. (1 and 3 can be any integer numbers from 1 to n). |

| | | | |
|---|---|---|---|
| | subtract 1 3 | Subtract equation 1 minus equation 3 and print the result equation | Print the result equation from subtracting equation 1 and equation 3. (1 and 3 can be any integer numbers from 1 to n). |
| | substitute x2 1 3 | Substitutes the variable x2 in equation 1 by its equation in equation 3 and print the result equation | Remove the variable x2 from equation 1 by substituting it with equation 3 and print the result equation |
| 3 | D | Cramer's Coefficient matrix | Prints the Cramer's coefficients matrix. |
| | D x1 | Cramer's Coefficient matrix of variable x1 | Prints the Cramer's coefficients matrix with answer-column values in x1 columns values |
| | D_value | Cramer's coefficient matrix determinant value | Prints the value of the determinant of the Coefficient matrix. |
| | solve | Print the solution of the equations | Prints the value of each variable that solves the solution<br>I no solution, the program should print "No Solution" |

**General Constraints:**

1- User can input equations can be up to 100 equations.

2- All values are of float data types in general case.
3- Do not prompt user to enter anything , just read the input directly.
4- Output should not include any extra white spaces or any extra texts more than the results.

**Hints:**

1. While parsing the equations, you will need to convert from string to float and from float to string. Use the following example as a guide:

```
#include<iostream>
#include<string>
#include<cstdlib>
#include<sstream>

using namespace std;

int main() {
// To convert from string to float use atof function
// atof needs #include<cstdlib>

string s = "1.5";
float v;
v = atof(s.c_str());


// To convert float-to string use ostringstream variable
// ostringstream needs #include<sstream>
// ostringstream variables are used in a similar way to cout
// Then get the string from the ostringstream using .str

float d = 1.55;
string str;
ostringstream ss;
ss << d;
str = ss.str();


return 0;
}
```

**Input / Output Samples**

User Input in Blue, Program Output in Red (Just for Clarification, not required in the program)

**Example 1 (num_vars operation)**

```
3
2x1+3x2+4x3=16
1x1+2x2+1x3=8
3x1+1x2+2x3+x4=13
num_vars
4
```

**Example 2 (equation i operation)**

```
3
2x1+3x2+4x3=16
1x1+2x2+1x3=8
3x1+1x2+2x3+x4=13
equation 2
1x1+2x2+1
```

**Example 3 (column operation)**

```
3
2x1+3x2+4x3=16
1x1+2x2+1x3=8
3x1+1x2+2x3+x4=13
column x2
3
2
1
```

**Example 4 (Add operation)**

```
3
2x1+3x2+4x3=16
1x1+2x2+1x3=8
3x1+1x2+2x3+x4=13
add 1 3
4x1+4x2+6x3+x4=29
```

**Example 5 (subtract operation)**

```
3
2x1+3x2+4x3=16
1x1+2x2+1x3=8
3x1+1x2+2x3+x4=13
subtract 3 1
2x1-1x2+1x3+x4=5
```

**Example 6 (substitute operation)**

**Example 7 (D operation)**

```
3
2x1+3x2+4x3=16
1x1+2x2+1x3=8
3x1+1x2+2x3=13
D
2     3     4
1     2     1
3     1     2
```

**Example 7 (D x2 operation)**

```
3
2x1+3x2+4x3=16
1x1+2x2+1x3=8
3x1+1x2+2x3=13
D x2
2     16     4
1     8      1
3     13     2
```

**Example 8 (D_value operation)**



```
3
2x1+3x2+4x3=16
1x1+2x2+1x3=8
3x1+1x2+2x3=13
D_value
-11
```

**Example 9 (solve operation) note that the solution of xi = (D xi)value/D_value**

```
3
2x1+3x2+4x3=16
1x1+2x2+1x3=8
3x1+1x2+2x3=13
solve
x1=3
x2=2
x3=1
```