Knowing that I need to output some specific content, so I still prefer to use my adept mathematics software (MATLAB) to solve this problem.

1. Convert Coordinates to Index (2-dimensional grid)

The derivation of the formula in the 2-dimensional grid is obvious. These derivations will elaborate in the third part (In the d-dimensional grid).

For output data, please kindly find the attached file " output_index_7_1.txt ".
For code details, please kindly find the attached file " Coordinates_7_1.m " (Open with MATLAB).
For the convenience of review, I have copied all the code of the document as follows.

```
[delimiterOut] = importdata('input_coordinates_7_1.txt');
% [delimiterOut] is mean ro let the program detects the input data
coordinates = delimiterOut.data;
% What we need is in this structure array, the container named 'data'

total = size(coordinates,1); % Total number of coordinate groups
x1 = coordinates(1:end,1); % Matrix containing the x1 coordinates
x2 = coordinates(1:end,2); % Matrix containing the x2 coordinates

L1 = 50; L2 = 57; % The following sentence will be used

fileID = fopen('output_index_7_1.txt','wt'); % Control the output
fprintf(fileID,'%s\n','index'); % Control the output (title)

for i = 1:total
    index(i) = L1 * x2(i) + x1(i);
    % Equation that converts coordinates to the index
    fprintf(fileID,'%u\n',index(i)); % Write the results into the document
end

fclose(fileID);  % Finish the control
```

2. Convert Index to Coordinates (2-dimensional grid)

The derivation of the formula in the 2-dimensional grid is obvious. Here is a brief introduction. That is the division with no remainder and the modulus operation. These derivations will elaborate in the fourth part (In the d-dimensional grid).

For output data, please kindly find the attached file " output_coordinates_7_1.txt ".
For code details, please kindly find the attached file " Index_7_1.m " (Open with MATLAB). For the convenience of review, I have copied all the code of the document as follows.

```
[delimiterOut] = importdata('input_index_7_1.txt');
% [delimiterOut] is mean ro let the program detects the input data
index = delimiterOut.data;
% What we need is in this structure array, the container named 'data'

total = size(index,1); % Total number of coordinate groups
L1 = 50; L2 = 57; % The following sentence will be used

fileID = fopen('output_coordinates_7_1.txt','wt'); % Control the output
fprintf(fileID,'%s\t%s\t\n','x1','x2'); % Control the output (Title)

for i = 1:total
    x1(i) = mod(index(i),L1); % Equation that convert index to x1 coordinate
    x2(i) = fix(index(i)/L1); % Equation that convert index to x2 coordinate
    fprintf(fileID,'%u\t%u\t\n',x1(i),x2(i));
    % Write the results into the document
end

fclose(fileID); % Finish the control
```

3. Convert Coordinates to Index (d-dimensional grid)

In the 2-dimensional grid, the equation that converts coordinates to the index is shown below

$$L_1 \cdot x_2 + x_1 = I$$

In the 3-dimensional grid, the equation that converts coordinates to the index is shown below

$$L_1 \cdot L_2 \cdot x_3 + L_1 \cdot x_2 + x_1 = I$$

It is not difficult to find the law. Therefore, the equation for the question request in the 6-dimensional grid is shown below

$$\begin{aligned} & L_1 \cdot L_2 \cdot L_3 \cdot L_4 \cdot L_5 \cdot x_6 \cdots \\ & + L_1 \cdot L_2 \cdot L_3 \cdot L_4 \cdot x_5 \cdots \\ & + L_1 \cdot L_2 \cdot L_3 \cdot x_4 \cdots \\ & + L_1 \cdot L_2 \cdot x_3 \cdots \\ & + L_1 \cdot x_2 \cdots \\ & + x_1 \cdots \\ & = I \end{aligned}$$

Using the code to implement, we can get the following code

For output data, please kindly find the attached file " output_index_7_2.txt ".
For code details, please kindly find the attached file " Coordinates_7_2.m " (Open with MATLAB).
For the convenience of review, I have copied all the code of the document as follows.

```
[delimiterOut] = importdata('input_coordinates_7_2.txt');
% [delimiterOut] is mean ro let the program detects the input data
coordinates = delimiterOut.data;
% What we need is in this structure array, the container named 'data'

total = size(coordinates,1); % Total number of coordinate groups
x1 = coordinates(1:end,1); % Matrix containing the x1 coordinates
x2 = coordinates(1:end,2); % Matrix containing the x2 coordinates
x3 = coordinates(1:end,3); % Matrix containing the x3 coordinates
x4 = coordinates(1:end,4); % Matrix containing the x4 coordinates
x5 = coordinates(1:end,5); % Matrix containing the x5 coordinates
x6 = coordinates(1:end,6); % Matrix containing the x6 coordinates

L1 = 4; L2 = 8; L3 = 5; L4 = 9; L5 = 6; L6 = 7;
% The following sentence will be used

fileID = fopen('output_index_7_2.txt','wt');  % Control the output
fprintf(fileID,'%s\n','index'); % Control the output (Title)

for i = 1:total
    index(i) = L1 * L2 * L3 * L4 * L5 * x6(i) ...
        + L1 * L2 * L3 * L4 * x5(i) ...
        + L1 * L2 * L3 * x4(i) ...
        + L1 * L2 * x3(i) ...
        + L1 * x2(i) ...
        + x1(i);
    % Equation that converts coordinates to the index
    fprintf(fileID,'%u\n',index(i)); % Write the results into the document
end

fclose(fileID); % Finish the control
```

4. Convert Index to Coordinates (d-dimensional grid)

In the 2-dimensional grid, the equation that converts index to coordinates is shown below

$$x_2 = fix(I / L_1)$$

$$x_1 = mod(I / L_1)$$

$fix()$ means division with no remainder, and $mod()$ means the modulus operation.

In the 3-dimensional grid, the equation that converts index to coordinates is shown below

$$x_3 = fix(I^{(0)} / L_1 \cdot L_2)$$

$$x_2 = fix(I^{(1)} / L_1), I^{(1)} = I^{(0)} - L_1 \cdot L_2 \cdot x_3$$

$$x_1 = mod(I^{(1)} / L_1), I^{(1)} = I^{(0)} - L_1 \cdot L_2 \cdot x_3$$

It is not difficult to find the law. Therefore, the equation for the question request in the 6-dimensional grid is shown below

$$x_6 = fix(I^{(0)} / L_1 \cdot L_2 \cdot L_3 \cdot L_4 \cdot L_5)$$

$$x_5 = fix(I^{(1)} / L_1 \cdot L_2 \cdot L_3 \cdot L_4), I^{(1)} = I^{(0)} - L_1 \cdot L_2 \cdot L_3 \cdot L_4 \cdot L_5 \cdot x_6$$

$$x_4 = fix(I^{(2)} / L_1 \cdot L_2 \cdot L_3), I^{(2)} = I^{(1)} - L_1 \cdot L_2 \cdot L_3 \cdot L_4 \cdot x_5$$

$$x_3 = fix(I^{(3)} / L_1 \cdot L_2), I^{(3)} = I^{(2)} - L_1 \cdot L_2 \cdot L_3 \cdot x_4$$

$$x_2 = fix(I^{(4)} / L_1), I^{(4)} = I^{(3)} - L_1 \cdot L_2 \cdot x_3$$

$$x_1 = mod(I^{(4)} / L_1), I^{(4)} = I^{(3)} - L_1 \cdot L_2 \cdot x_3$$

It seems very simple, but it needs to pay attention to one detail when you put this code into the computer. That is, it cannot calculate all coordinates at once. It can only estimate the maximum coordinate first and then put the result into the next coordinate calculation. Then, it quate these two maximum coordinates and put these two results into the following coordinate calculation, and so on.

Using the code to implement, we can get the following code

For output data, please kindly find the attached file " output_coordinates_7_2.txt ".
For code details, please kindly find the attached file " Index_7_2.m " (Open with MATLAB). For the convenience of review, I have copied all the code of the document as follows.

```
[delimiterOut] = importdata('input_index_7_2.txt');
% [delimiterOut] is mean ro let the program detects the input data
index = delimiterOut.data;
% What we need is in this structure array, the container named 'data'

total = size(index,1); % Total number of coordinate groups
L1 = 4; L2 = 8; L3 = 5; L4 = 9; L5 = 6; L6 = 7;
% The following sentence will be used

fileID = fopen('output_coordinates_7_2.txt','wt'); % Control the output
fprintf(fileID,'%s\t%s\t%s\t%s\t%s\t%s\t\n','x1','x2','x3','x4','x5','x6');
% Control the output (Title)
```

```
for i = 1:total

    x6_eq = index(i) / (L1 * L2 * L3 * L4 * L5);
    x6(i) = fix(x6_eq);
    % Equation that converts index to x6 coordinate

    x5_eq = (index(i) - L1 * L2 * L3 * L4 * L5 * x6(i)) / (L1 * L2 * L3 * L4);
    x5(i) = fix(x5_eq);
    % Equation that converts index to x6 coordinate
    % Quote the calculation of x6

    x4_eq = (index(i) - L1 * L2 * L3 * L4 * L5 * x6(i) ...
        - L1 * L2 * L3 * L4 * x5(i)) / (L1 * L2 * L3);
    x4(i) = fix(x4_eq);
    % Equation that converts index to x4 coordinate
    % Quote the calculation of x6 and x5

    x3_eq = (index(i) - L1 * L2 * L3 * L4 * L5 * x6(i) ...
        - L1 * L2 * L3 * L4 * x5(i) - L1 * L2 * L3 * x4(i)) / (L1 * L2);
    x3(i) = fix(x3_eq);
    % Equation that converts index to x3 coordinate
    % Quote the calculation of x6, x5, and x4

    x2_eq = (index(i) - L1 * L2 * L3 * L4 * L5 * x6(i) ...
        - L1 * L2 * L3 * L4 * x5(i) - L1 * L2 * L3 * x4(i) ...
        - L1 * L2 * x3(i)) / L1;
    x2(i) = fix(x2_eq);
    % Equation that converts index to x2 coordinate
    % Quote the calculation of x6, x5, x4, and x3

    x1_eq = (index(i) - L1 * L2 * L3 * L4 * L5 * x6(i) ...
        - L1 * L2 * L3 * L4 * x5(i) - L1 * L2 * L3 * x4(i) ...
        - L1 * L2 * x3(i) - L1 * x2(i)); ...
    x1(i) = mod(x1_eq,L1);
    % Equation that converts index to x2 coordinate
    % Quote the calculation of x6, x5, x4, x3, and x2


fprintf(fileID,'%u\t%u\t%u\t%u\t%u\t%u\t\n',x1(i),x2(i),x3(i),x4(i),x5(i),x6(i));
     % Write the results into the document

end

fclose(fileID); % Finish the control
```