```
Joint PMF of X and Y
   y: 0 1 2 3 4 5 6 7
x ---------------------------------------------------------------
0 | 0.0000
1 | 0.0000 0.0000
2 | 0.0000 0.0004 0.0019
3 | 0.0001 0.0006 0.0048 0.0185
4 | 0.0001 0.0011 0.0077 0.0288 0.0651
5 | 0.0000 0.0010 0.0063 0.0254 0.0602 0.0841
6 | 0.0000 0.0006 0.0036 0.0125 0.0292 0.0414 0.0330
7 | 0.0000 0.0001 0.0007 0.0028 0.0067 0.0090 0.0066 0.0022

Conditional PMF of X given Y
   y: 0 1 2 3 4 5 6 7
x ---------------------------------------------------------------
0 | 0.0000
1 | 0.0000 0.0129
2 | 0.1071 0.1085 0.0750
3 | 0.2143 0.1473 0.1917 0.2108
4 | 0.4286 0.2920 0.3077 0.3274 0.4040
5 | 0.1786 0.2584 0.2507 0.2882 0.3735 0.6250
6 | 0.0714 0.1499 0.1460 0.1420 0.1811 0.3081 0.8344
7 | 0.0000 0.0310 0.0289 0.0316 0.0415 0.0669 0.1656 1.0000

Conditional PMF of Y given X
   y: 0 1 2 3 4 5 6 7
x ---------------------------------------------------------------
0 | 0.0000
1 | 0.0000 1.0000
2 | 0.0129 0.1810 0.8060
3 | 0.0025 0.0238 0.1996 0.7741
4 | 0.0012 0.0110 0.0746 0.3274 0.6333
5 | 0.0003 0.0057 0.0353 0.1433 0.3403 0.4752
6 | 0.0002 0.0048 0.0302 0.1038 0.2425 0.3443 0.2742
7 | 0.0000 0.0043 0.0257 0.0992 0.2388 0.3212 0.2338 0.0771
```

I completed this lab in C++, utilizing arrays of arrays and proper parsing to stimulate the situation. I used the following code:

```
#include <iostream>
#include <stdio.h>      /* NULL */
#include <stdlib.h>     /* srand */
#include <time.h> /* time */
#include <iomanip> /* fixed and setprecision */
```

```
void lab(){
  srand (time(NULL));
  int x=0, y = 0;
  float arr [8][8], pmfxly[8][8], pmfylx[8][8], pmfx[8], pmfy[8];
  for (int i = 0 ; i < 8; i++) {
    for (int a = 0; a<8; a++) {
      arr[i][a]=0.0;
      pmfxly[i][a]=0.0;
      pmfylx[i][a]=0.0;
    }
    pmfx[i]=0.0;
    pmfy[i]=0.0;
  }
  for (int a = 0 ; a < 100000; a++){
    for (int i = 0; i<7; i++){
      if (rand() % 10 + 1 <= 6) x++;
      if (rand() % 10 + 1 <= 7) y++;
    }
    arr[x][y]++;
    x = 0;
    y = 0;
  }
  for (int i = 0; i<8; i++) for (int a=0; a<8 && a<=i;a++) arr[i][a]/=100000.0;
  for (int i = 0 ; i < 8; i++) for (int a = 0; a<8 && a<=i; a++) pmfx[i] += arr[i][a];
  for (int i = 0 ; i < 8; i++) for (int a = i; a<8; a++) pmfy[i] += arr[a][i];
  for (int i = 0 ; i < 8; i++) for (int a = 0; a<8 && a<=i; a++){
    if (pmfx[i]!=0.0) pmfylx[i][a] = arr[i][a] / pmfx[i];
  }
  for (int i = 0 ; i < 8; i++) for (int a = i; a<8; a++){
    if (pmfy[i]!=0.0) pmfxly[a][i] = arr[a][i] / pmfy[i];
  }
  std::cout << "Joint PMF of X and Y\n\ty: 0 1 2 3 4 5 6 7\nx
-----------------------------------------------------------------\n0 | " << std::fixed << std::setprecision(4) <<
arr[0][0] << "\n1 | " << arr[1][0] << " " << arr[1][1] << "\n2 | " << arr[2][0] << " " << arr[2][1] << " "
<< arr[2][2] << "\n3 | " << arr[3][0] << " " << arr[3][1] << " " << arr[3][2] << " " << arr[3][3] << "\n4 |
" << arr[4][0] << " " << arr[4][1] << " " << arr[4][2] << " " << arr[4][3] << " " << arr[4][4] << "\n5 | "
<< arr[5][0] << " " << arr[5][1] << " " << arr[5][2] << " "<< arr[5][3] << " " << arr[5][4] << " " <<
arr[5][5] << "\n6 | " << arr[6][0] << " " << arr[6][1] << " " << arr[6][2] << " " << arr[6][3] << " " <<
arr[6][4] << " " << arr[6][5] << " " << arr[6][6] << "\n7 | " << arr[7][0] << " " << arr[7][1] << " " <<
arr[7][2] << " " << arr[7][3] << " " << arr[7][4] << " " << arr[7][5] << " " << arr[7][6] << " " <<
arr[7][7] << "\nConditional PMF of X given Y\n\ty: 0 1 2 3 4 5 6 7\nx
-----------------------------------------------------------------\n0 | " << pmfxly[0][0] << "\n1 | " <<
pmfxly[1][0] << " " << pmfxly[1][1] << "\n2 | " << pmfxly[2][0] << " " << pmfxly[2][1] << " " <<
pmfxly[2][2] << "\n3 | " << pmfxly[3][0] << " " << pmfxly[3][1] << " " << pmfxly[3][2] << " " <<
pmfxly[3][3] << "\n4 | " << pmfxly[4][0] << " " << pmfxly[4][1] << " " << pmfxly[4][2] << " " <<
pmfxly[4][3] << " " << pmfxly[4][4] << "\n5 | " << pmfxly[5][0] << " " << pmfxly[5][1] << " " <<
pmfxly[5][2] << " " << pmfxly[5][3] << " " << pmfxly[5][4] << " " << pmfxly[5][5] << "\n6 | " <<
pmfxly[6][0] << " " << pmfxly[6][1] << " " << pmfxly[6][2] << " " << pmfxly[6][3] << " "<<
pmfxly[6][4] <<" "<< pmfxly[6][5] << " " << pmfxly[6][6] << "\n7 | " << pmfxly[7][0] << " " <<
pmfxly[7][1] << " " << pmfxly[7][2] << " " << pmfxly[7][3] << " " << pmfxly[7][4] << " " <<
pmfxly[7][5] << " " << pmfxly[7][6] << " " << pmfxly[7][7] << "\nConditional PMF of Y given X\n\ty:
0 1 2 3 4 5 6 7\nx ---------------------------------------------------------------\n0 | " << pmfylx[0][0] <<
"\n1 | " << pmfylx[1][0] << " " << pmfylx[1][1] << "\n2 | " << pmfylx[2][0] << " " << pmfylx[2][1] << "
" << pmfylx[2][2] << "\n3 | " << pmfylx[3][0] << " " << pmfylx[3][1] << " " << pmfylx[3][2] << " " <<
pmfylx[3][3] << "\n4 | " << pmfylx[4][0] << " " << pmfylx[4][1] << " " << pmfylx[4][2] << " " <<
pmfxly[4][3] << " " << pmfylx[4][4] << "\n5 | " << pmfylx[5][0] << " " << pmfylx[5][1] << " " <<
pmfylx[5][2] << " " << pmfylx[5][3] << " " << pmfylx[5][4] << " " << pmfylx[5][5] << "\n6 | " <<
pmfylx[6][0] << " " << pmfylx[6][1] << " " << pmfylx[6][2] << " " << pmfylx[6][3] << " "<<
pmfylx[6][4] <<" "<< pmfylx[6][5] << " " << pmfylx[6][6] << "\n7 | " << pmfylx[7][0] << " " <<
pmfylx[7][1] << " " << pmfylx[7][2] << " " << pmfylx[7][3] << " " << pmfylx[7][4] << " " <<
pmfylx[7][5] << " " << pmfylx[7][6] << " " << pmfylx[7][7];
}
int main() {
  lab();
  return 0;
}
```