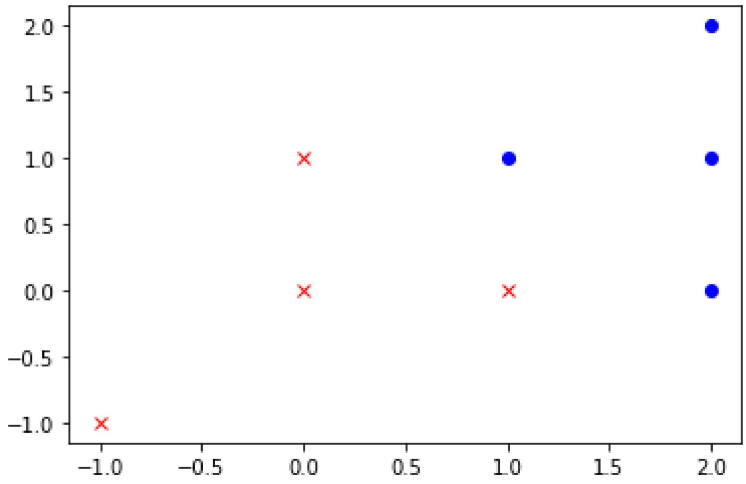
In [1]:
```
import numpy as np
import time
import matplotlib.pyplot as plt

from load_data import load_data

X,y = load_data('data.txt')
plt.plot(X[np.where(y==1)[0], 0], X[np.where(y==1)[0], 1], 'rx')
plt.plot(X[np.where(y==-1)[0], 0], X[np.where(y==-1)[0], 1], 'bo')
plt.show()
```



In [2]:
```
from smo import SMO
t = time.time()
model = SMO(max_iter=10000, kernel_type='linear', C=1.0, tol=0.001)
support_vectors, count, alpha = model.fit(X, y)
print('\n')
elapsed = time.time() - t
print('time taken to find solution', elapsed)
```

```
x shape (8, 2)
y shape (8, 1)

For iteration 1 alpha is:
 [[0. ]
 [0. ]
 [0. ]
 [0. ]
 [0.4]
 [0. ]
 [0. ]
 [0. ]]

For iteration 2 alpha is:
 [[0. ]
 [0. ]
 [0. ]
 [0. ]
 [0.9]
 [0. ]
 [0. ]
 [0. ]]

For iteration 3 alpha is:
```

```
 [[0.1 ]
 [0.34]
 [0.  ]
 [0.  ]
 [1.  ]
 [1.  ]
 [0.  ]
 [0.  ]]

For iteration 4 alpha is:
 [[0.38]
 [0.  ]
 [0.  ]
 [0.  ]
 [1.  ]
 [1.  ]
 [0.  ]
 [0.  ]]

For iteration 5 alpha is:
 [[0.69      ]
 [0.        ]
 [0.        ]
 [0.11923077]
 [1.        ]
 [1.        ]
 [0.26676923]
 [0.        ]]

For iteration 6 alpha is:
 [[0.69      ]
 [0.        ]
 [0.        ]
 [0.655     ]
 [0.26676923]
 [0.96088462]
 [0.26676923]
 [0.        ]]

For iteration 7 alpha is:
 [[0.69      ]
 [0.        ]
 [0.        ]
 [0.42370769]
 [0.26676923]
 [1.        ]
 [0.42153077]
 [0.        ]]

For iteration 8 alpha is:
 [[1.        ]
 [0.        ]
 [0.07629231]
 [0.42370769]
 [0.26676923]
 [1.        ]
 [0.6883    ]
 [0.        ]]

For iteration 9 alpha is:
 [[1.        ]
 [0.        ]
 [0.07629231]
 [0.42370769]
 [0.26676923]
```

← Optimum α

```
[1.        ]
[0.6883    ]
[0.        ]]
```

```
time taken to find solution 0.019887447357177734
```

In [3]:

```
w = model.calc_w(alpha, y, X)
bias = model.calc_b(X, y, w)
print('Final hyper plane parameters - ')
print('W: ', w.T[0], 'and b:', bias)
print('Number of support vectors: ', count)
```

```
Final hyper plane parameters -
W:  [-1.         -0.73540769] and b: 1.2427038461538458
Number of support vectors:  6
```

In [4]:

```
X,y = load_data('data.txt')
plt.plot(X[np.where(y==1)[0], 0], X[np.where(y==1)[0], 1], 'rx')
plt.plot(X[np.where(y==-1)[0], 0], X[np.where(y==-1)[0], 1], 'bo')
x1, x2 = np.min(X[:,1]), np.max(X[:,1])
y1 = -w[0]/w[1] * x1 - bias/w[1]
y2 = -w[0]/w[1] * x2 - bias/w[1]
plt.plot([x1, x2], [y1,y2], color='green')
plt.show()
```