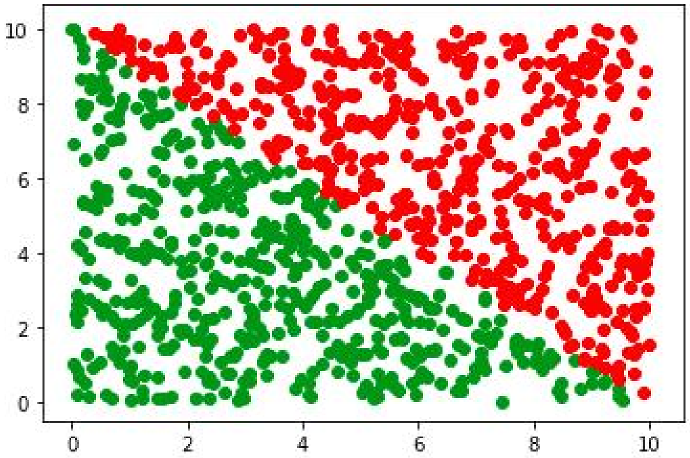
In [20]:
```python
import numpy as np
import matplotlib.pyplot as plt
from math import pi
```

In [21]:
```python
mean_1 = np.array([3, 3])
mean_2 = np.array([7, 7])
data = np.random.uniform(0, 10, (1000, 2))
```

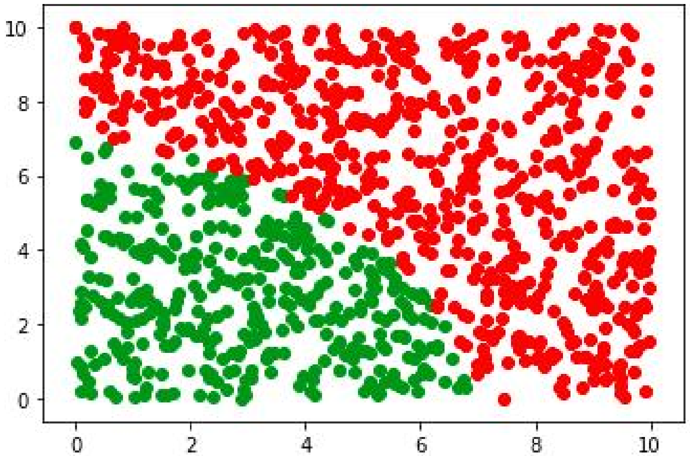
In [41]:
```python
def maximum_possibility_estimate(test, mean_1, means_2, covariance_1, covariance_2):
    covariances = [covariance_1, covariance_2]
    means = [mean_1, mean_2]
    A = []
    B = []
    inverse = []
    determinants = []
    dimensions = []

    covariance = covariances[0]
    inverse.append(covariance)
    V, D = np.linalg.eig(covariance)
    V = V.real
    determinant = 0
    dim = 0
    for v in V:
        if v > 0:
            determinant = determinant + np.log(v)
            dim = dim + 1
    dimensions.append(dim)
    determinants.append(determinant)

    covariance = covariances[1]
    inverse.append(covariance)
    V, D = np.linalg.eig(covariance)
    V = V.real
    determinant = 0
    dim = 0
    for v in V:
        if v > 0:
            determinant = determinant + np.log(v)
            dim = dim + 1
    dimensions.append(dim)
    determinants.append(determinant)

    for i in range(test.shape[0]):
        possibilities = []
        for j in range(2):
```

```
        if v > 0:
            determinant = determinant + np.log(v)
            dim = dim + 1
    dimensions.append(dim)
    determinants.append(determinant)

    covariance = covariances[1]
    inverse.append(covariance)
    V, D = np.linalg.eig(covariance)
    V = V.real
    determinant = 0
    dim = 0
    for v in V:
        if v > 0:
            determinant = determinant + np.log(v)
            dim = dim + 1
    dimensions.append(dim)
    determinants.append(determinant)

    for i in range(test.shape[0]):
        possibilities = []
        for j in range(2):
            possibility = -0.5 * ((dimensions[j] * np.log(2 * pi) + determinants[j]) + ((test[i] - means[j]).T @
            possibilities.append(possibility)
        if np.argmax(possibilities) == 0:
            A.append(test[i])
        else:
            B.append(test[i])

    A = np.array(A)
    B = np.array(B)

    plt.scatter(A[:, 0], A[:, 1], c='g')
    plt.scatter(B[:, 0], B[:, 1], c='r')
    plt.show()
```

In [50]:
```
covariance_1 = 3 * np.identity(2)
covariance_2 = covariance_1
maximum_likelihood_estimate(test, mean_1, mean_2, covariance_1, covariance_2)
```



In [51]:
```
covariance_1 = np.array([[7, 2], [1, 7]])
covariance_2 = np.array([[3, 1], [2, 3]])
maximum_likelihood_estimate(test, mean_1, mean_2, covariance_1, covariance_2)
```



Question 3

Prove that a covariance matrix is positive, semi definite ie PSD

For a sample of vectors $x_i = (x_{i1}, x_{i2}, \ldots, x_{ik})^T$, with $i = 1, \ldots, n$

the sample vector is

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

and the sample covariance matrix is.

$$Q = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T$$

For a non-zero vector, $y \in R^k$, we have

$$y^T Q y = y^T \left(\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T \right) y$$

$$= \frac{1}{n} \sum_{i=1}^{n} y^T (x_i - \bar{x})(x_i - \bar{x})^T y$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left(y (x_i - x)^T \right)^2 \geq 0 \quad \text{— (1)}$$

Hence, Q is always positive semi-definite.

Another proof.

Defining $z_i = (x_i - \bar{x})$ for $i = 1, \ldots, n$.

For any non-zero $y \in R^k$, (1) is zero if and only if $z_i, y = 0$ for each $i = 1, \ldots, n$. Suppose the set $\{z_1, \ldots, z_n\}$ spans R^k. Then, there are real numbers $\alpha_1 \ldots \alpha_n$ such that $y = \alpha z_1 + \ldots + \alpha_n z_n$. Then we have.

$y^T y = \alpha_1 z_1^T y + \ldots + \alpha_n z_n^T y = 0$, yielding $y = 0$, a contradiction.

Hence is z_i's span is R^k, the Q is +ve definite.

The condition is equivalent to $\text{rank}[z_1 \ldots z_n] = k$.

a) $$\frac{e^{-1/2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)}{\sqrt{2\pi}\ |\Sigma|^{1/2}} = \frac{e^{-1/2}(x-\mu_2)^T \Sigma^{-1}(x-\mu_2)}{\sqrt{2\pi}\ |\Sigma|^{1/2}}$$

Taking log on both sides

$$= -\frac{1}{2}\left[\begin{bmatrix} x_1-3 \\ x_2-3 \end{bmatrix}^T \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1-3 \\ x_2-3 \end{bmatrix}\right] = -\frac{1}{2}\left[\begin{bmatrix} x_1-7 \\ x_2-7 \end{bmatrix}^T \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1-7 \\ x_2-7 \end{bmatrix}\right]$$

$$= \begin{bmatrix} 3(x_1-3) & 3(x_2-3) \end{bmatrix} \begin{bmatrix} x_1-3 \\ x_2-3 \end{bmatrix} = \begin{bmatrix} 3(x_1-7) & 3(x_1-7) \end{bmatrix} \begin{bmatrix} x_1-7 \\ x_2-7 \end{bmatrix}$$

$$3(x_1-3)^2 + 3(x_2-3)^2 = 3(x_1-7)^2 + 3(x_2-7)^2$$

$$= x_1^2 - 6x_1 + 9 + x_2^2 - 6x_2 + 9 = x_1^2 + 49 - 14x + x_2^2 + 49 - 14x_2$$

$$8(x_1 + x_2) = 80.$$

$$\boxed{x_2 = 10 - x_1}$$

$$\frac{e^{-1/2 (x_1-\mu)^T \Sigma_1^{-1} (x-\mu_1)}}{\sqrt{2\pi} \quad |\Sigma|^{1/2}} = \frac{e^{-1/2 (x-\mu_2)^T \Sigma_2^{-1} (x-\mu_2)}}{\sqrt{2\pi} \; |\Sigma|^{1/2}}$$

Taking log on both sides

$$\left(\begin{bmatrix} x_1-3 \\ x_2-3 \end{bmatrix}^T \frac{1}{7}\begin{bmatrix} 3 & -1 \\ -2 & 3 \end{bmatrix}\begin{bmatrix} x_1-3 \\ x_2-3 \end{bmatrix}\right) + \ln 7 = \left(\begin{bmatrix} x_1-7 \\ x_2-7 \end{bmatrix}^T \frac{1}{47}\begin{bmatrix} 7 & -2 \\ -1 & 7 \end{bmatrix}\begin{bmatrix} x_1-7 \\ x_2-7 \end{bmatrix}\right) + \ln 4$$

$$\frac{1}{7}\left[3(x_1-3) - 2(x_2-3) - (x_1-3) + 3(x_2-3)\right]\begin{bmatrix} x_1-3 \\ x_2-3 \end{bmatrix} + \ln 7.$$

$$= \frac{1}{47}\left[7(x_1-7) - (x_2-7) - 2(x_1-7) + 7(x_2-7)\right]\begin{bmatrix} x_1-7 \\ x_2-7 \end{bmatrix} + \ln 47.$$

$$= \frac{1}{7}\left(3(x_1-3)^2 - 2(x_2-3)(x_1-3) - (x_1-3)(x_2-3) + 3(x_2-3)^2\right] + \ln 7$$

$$= \frac{1}{47}\left[7(x_1-7)^2 + (x_2-7)^2 - (x_2-7)(x_1-7) - 2(x_1-7)(x_2-7)\right] + \ln 47.$$