Problem 3

write a program that fits a straight line to data -> f(x) = ax + b

In [2]:

```
D = {(0, 0.5),(1, 2.0),(2,1.0),(3,1.5),(4,7.5)} # init data (x,y)
D = list(D) # change set to list
for _ in range(0,5): # also change inner
    D[_] = list(D[_])
print(D)
```

```
[[1, 2.0], [4, 7.5], [0, 0.5], [3, 1.5], [2, 1.0]]
```

make a function compute_error(a, b) that computes error between the straight line f(x) = ax+b and D

In [3]:

```
def compute_error(a, b):
    e = 0
    for _ in range(5):
        e += (a * D[_][0] + b - D[_][1]) ** 2 # e = SIGMA(i=1~5) ax + b - y
    return e
```

Search for a and b such that e is minimized.

In [4]:

```
# a: 양수, b: 음수 (1, 3사분면 지남)

min = compute_error(1, -1) # error 초기 최소값
min_ab = [1, -1] # 최소값일 경우의 a, b

iter = 0.1 # 가중치 설정 (It's TRADEOFF)

for i in range(0, 1000, 1):

    if i % 100 == 0:
        print((i+100)/10,"%..")

    for j in range(-1000, 1):
        E = compute_error(i*iter, j*iter)
        if E < min:
            min = E; min_ab[0] = i*iter; min_ab[1] = j*iter

print("E min:", round(min, 3),"at [a, b]:",min_ab)
```

```
10.0 %..
20.0 %..
30.0 %..
40.0 %..
50.0 %..
60.0 %..
70.0 %..
80.0 %..
90.0 %..
100.0 %..
E min: 14.3 at [a, b]: [1.3, -0.1]
```

Plot a straight line f(x) = ax + b

In [55]:

```
import pylab as plt
import numpy as np
# a: min_ab[0], b: min_ab[1]
x = np.arange(0, 11, 1)
y = []
for _ in range(0, 11, 1):
    y.append(min_ab[0] * x[_] + min_ab[1])
plt.ylim(0, 10)
plt.xlim(0, 8)
plt.plot(x, y, 'r')
plt.title('Draw close line that fits to data')
plt.xlabel('x'); plt.ylabel('y')
plt.legend('m')

#plt.scatter - 그래프에 점찍기
for _ in range(5):
    plt.scatter(D[_][0], D[_][1])
```