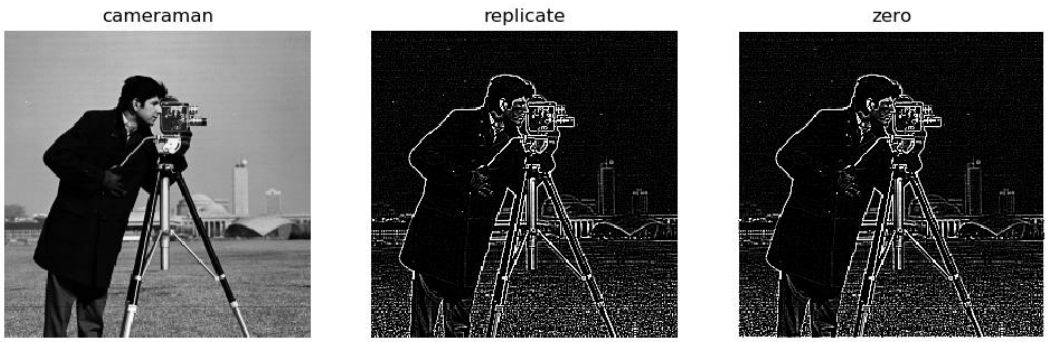
问题 1：

（程序为 q3.py）
m 为核的尺寸，边界补充个数为 a=（m-1）/2
补 0 思想：建一个比原先长宽均多 2a 的零矩阵，直接使其中间的元素等于原图像的元素。
像素复制思想：建一个比原先长宽均多 2a 的零矩阵，先判断最上面补充出来的，使其等于原图像的第零行元素，往下判断，左面补充出来的元素等于原图像第 0 列的元素，往右判断，元素等于原图像元素，右面补充出来的元素等于原图像最后一列的元素。再往下看，矩阵下面补充出来的元素等于原图像最后一行的元素。



问题 2：

（程序为 q4.py）
输入 m，可以是 no，即无输入，也可以是大于 0 的数。
no 的话代表没有给 m，所以 m = math.ceil (3*sig)*2 + 1
在写高斯核的时候，坐标是以中心为原点，由下式求得再归一化。

$$w=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

下图是输入为 no 时：

```
E:\Anaconda3\envs\pytorch\python.exe C:/Users/KangLiu/PycharmProjects/1image/q4.py
输入‘no’或者是已知的数m
no
[[1.96519161e-05 2.39409349e-04 1.07295826e-03 1.76900911e-03
  1.07295826e-03 2.39409349e-04 1.96519161e-05]
 [2.39409349e-04 2.91660295e-03 1.30713076e-02 2.15509428e-02
  1.30713076e-02 2.91660295e-03 2.39409349e-04]
 [1.07295826e-03 1.30713076e-02 5.85815363e-02 9.65846250e-02
  5.85815363e-02 1.30713076e-02 1.07295826e-03]
 [1.76900911e-03 2.15509428e-02 9.65846250e-02 1.59241126e-01
  9.65846250e-02 2.15509428e-02 1.76900911e-03]
 [1.07295826e-03 1.30713076e-02 5.85815363e-02 9.65846250e-02
  5.85815363e-02 1.30713076e-02 1.07295826e-03]
 [2.39409349e-04 2.91660295e-03 1.30713076e-02 2.15509428e-02
  1.30713076e-02 2.91660295e-03 2.39409349e-04]
 [1.96519161e-05 2.39409349e-04 1.07295826e-03 1.76900911e-03
  1.07295826e-03 2.39409349e-04 1.96519161e-05]]

Process finished with exit code 0
```

下图是输入为 3 时：

```
E:\Anaconda3\envs\pytorch\python.exe C:/Users/KangLiu/Pyc
输入‘no’或者是已知的数m
3
warring:m is too small
[[0.07511361 0.1238414  0.07511361]
 [0.1238414  0.20417996 0.1238414 ]
 [0.07511361 0.1238414  0.07511361]]

Process finished with exit code 0
```

下图是输入为 9 时：

```
E:\Anaconda3\envs\pytorch\python.exe C:/Users/KangLiu/PycharmProjects/1image/q4.py
输入‘no’或者是已知的数m
9
[[1.79106361e-08 5.93118809e-07 7.22566631e-06 3.23831897e-05
  5.33908537e-05 3.23831897e-05 7.22566631e-06 5.93118809e-07
  1.79106361e-08]
 [5.93118809e-07 1.96413974e-05 2.39281205e-04 1.07238396e-03
  1.76806225e-03 1.07238396e-03 2.39281205e-04 1.96413974e-05
  5.93118809e-07]
 [7.22566631e-06 2.39281205e-04 2.91504184e-03 1.30643112e-02
  2.15394077e-02 1.30643112e-02 2.91504184e-03 2.39281205e-04
  7.22566631e-06]
 [3.23831897e-05 1.07238396e-03 1.30643112e-02 5.85501805e-02
  9.65329280e-02 5.85501805e-02 1.30643112e-02 1.07238396e-03
  3.23831897e-05]
 [5.33908537e-05 1.76806225e-03 2.15394077e-02 9.65329280e-02
  1.59155892e-01 9.65329280e-02 2.15394077e-02 1.76806225e-03
  5.33908537e-05]
 [3.23831897e-05 1.07238396e-03 1.30643112e-02 5.85501805e-02
  9.65329280e-02 5.85501805e-02 1.30643112e-02 1.07238396e-03
```

问题 3：

（程序为 q5.py）
把问题 1、2 的两个函数整合为一个函数
twodConv(f,sig,method='zero')
程序运行后会出现四张图对应如下：

下图是一张图片在 sig=1,2,3,5 下的区别对比：

下图是分别采用 sig=1,2,3,5 的结果图：

下图分别对应 sig=1 时自己写的函数与直接调用 GaussianBlur 函数的比较以及两张图的差值图：

下图分别对应像素复制和补零差异以及他们的差值图（此时 sig=3）：

两张图进行对比，明显看得到使用 zero 方法时图片的边缘有一圈黑色。我进行的差值是 zero-replicate。可以看出两种方法效果基本一样，但 replicate 没有黑边。