```python
import os
import cv2
import numpy as np

""" CONSTS """
PATH = os.path.dirname(os.path.abspath(__file__))


""" 12 задание """
def twelve():
    """ a """
    image_with_details = cv2.imread(f'{PATH}/first.jpg')
    cv2.imshow("Default Details", image_with_details)
    print('Original Dimensions : ', image_with_details.shape)
    width = int(image_with_details.shape[1] * 0.5)
    height = int(image_with_details.shape[0] * 0.5)
    dim = (width, height)
    resized = cv2.resize(image_with_details, dsize=dim)
    width = int(resized.shape[1] * 0.5)
    height = int(resized.shape[0] * 0.5)
    dim = (width, height)
    resized = cv2.resize(resized, dsize=dim)
    width = int(resized.shape[1] * 0.5)
    height = int(resized.shape[0] * 0.5)
    dim = (width, height)
    resized = cv2.resize(resized, dsize=dim)
    cv2.namedWindow('Resized', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('Resized', 1280, 854)
    print('Resized Dimensions : ', resized.shape)
    cv2.imshow("Resized", resized)
    cv2.waitKey(0)
    """ b """
    pyr_image = cv2.pyrDown(image_with_details)
    pyr_image = cv2.pyrDown(pyr_image)
    pyr_image = cv2.pyrDown(pyr_image)
    cv2.namedWindow('Resized', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('Resized', 1280, 854)
    print('Resized Dimensions : ', resized.shape)
    cv2.imshow("Resized", resized)
    cv2.namedWindow('PyrImage', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('PyrImage', 1280, 854)
    print('PyrImage : ', pyr_image.shape)
    cv2.imshow("PyrImage", pyr_image)
    cv2.waitKey(0)
    """ c """
    """
    Изображение уменьшенное при помощи pyrDown получилось более размытым. pyrDown понижает шаг дискретизации
    пирамиды Гаусса, в то время как resize меняет размер применяя раличные методы интерполяции.
    """


""" 13 задание """
def thirteen():
    image = cv2.imread(f'{PATH}/second.jpg')
    ret, thresh = cv2.threshold(image, 128, 255, cv2.THRESH_MASK)
    cv2.namedWindow('THRESH_MASK', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('THRESH_MASK', 1280, 854)
    cv2.imshow("THRESH_MASK", thresh)
    ret, thresh = cv2.threshold(image, 128, 255, cv2.THRESH_BINARY)
    cv2.namedWindow('THRESH_BINARY', cv2.WINDOW_KEEPRATIO)
```

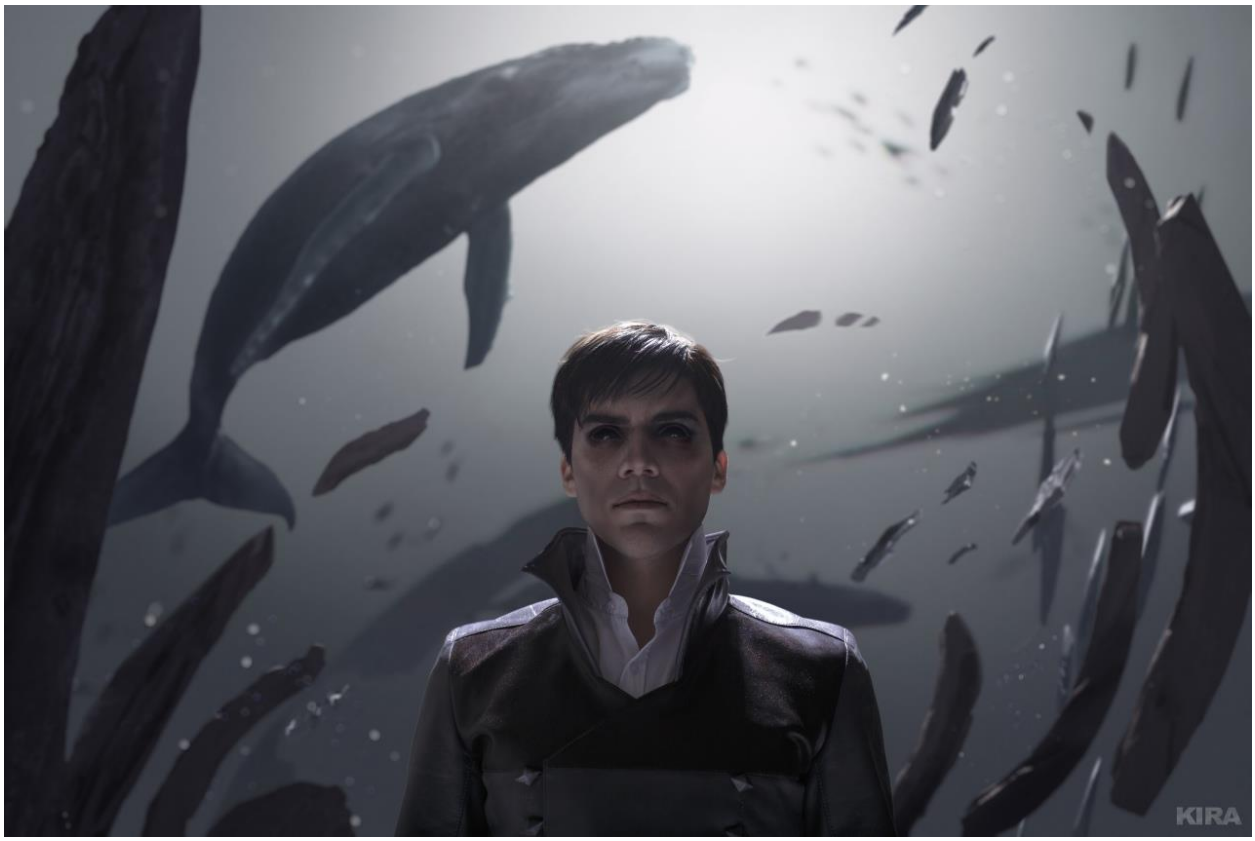
```
    cv2.resizeWindow('THRESH_BINARY', 1280, 854)
    cv2.imshow("THRESH_BINARY", thresh)
    ret, thresh = cv2.threshold(image, 128, 255, cv2.THRESH_BINARY_INV)
    cv2.namedWindow('THRESH_BINARY_INV', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('THRESH_BINARY_INV', 1280, 854)
    cv2.imshow("THRESH_BINARY_INV", thresh)
    ret, thresh = cv2.threshold(image, 128, 255, cv2.THRESH_TRUNC)
    cv2.namedWindow('THRESH_TRUNC', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('THRESH_TRUNC', 1280, 854)
    cv2.imshow("THRESH_TRUNC", thresh)
    ret, thresh = cv2.threshold(image, 128, 255, cv2.THRESH_TOZERO)
    cv2.namedWindow('THRESH_TOZERO', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('THRESH_TOZERO', 1280, 854)
    cv2.imshow("THRESH_TOZERO", thresh)
    ret, thresh = cv2.threshold(image, 128, 255, cv2.THRESH_TOZERO_INV)
    cv2.namedWindow('THRESH_TOZERO_INV', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('THRESH_TOZERO_INV', 1280, 854)
    cv2.imshow("THRESH_TOZERO_INV", thresh)
    cv2.waitKey(0)
    """ a """
    image = cv2.imread(f'{PATH}/second.jpg', cv2.CV_8UC1)
    thresh = cv2.adaptiveThreshold(image, 128,
adaptiveMethod=cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                   thresholdType=cv2.THRESH_BINARY, blockSize=5, C=5)
    cv2.namedWindow('Adaptive TRESH_BINARY C=5', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('Adaptive TRESH_BINARY C=5', 1280, 854)
    cv2.imshow("Adaptive TRESH_BINARY C=5", thresh)
    thresh = cv2.adaptiveThreshold(image, 128,
adaptiveMethod=cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                   thresholdType=cv2.THRESH_BINARY_INV, blockSize=5,
C=5)
    cv2.namedWindow('Adaptive TRESH_BINARY_INV C=5', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('Adaptive TRESH_BINARY_INV C=5', 1280, 854)
    cv2.imshow("Adaptive TRESH_BINARY_INV C=5", thresh)
    cv2.waitKey(0)
    """ b """
    thresh = cv2.adaptiveThreshold(image, 128,
adaptiveMethod=cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                   thresholdType=cv2.THRESH_BINARY, blockSize=5, C=0)
    cv2.namedWindow('Adaptive TRESH_BINARY C=0', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('Adaptive TRESH_BINARY C=0', 1280, 854)
    cv2.imshow("Adaptive TRESH_BINARY C=0", thresh)
    thresh = cv2.adaptiveThreshold(image, 128,
adaptiveMethod=cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                   thresholdType=cv2.THRESH_BINARY_INV, blockSize=5,
C=0)
    cv2.namedWindow('Adaptive TRESH_BINARY_INV C=0', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('Adaptive TRESH_BINARY_INV C=0', 1280, 854)
    cv2.imshow("Adaptive TRESH_BINARY_INV C=0", thresh)
    cv2.waitKey(0)
    thresh = cv2.adaptiveThreshold(image, 128,
adaptiveMethod=cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                   thresholdType=cv2.THRESH_BINARY, blockSize=5, C=-
5)
    cv2.namedWindow('Adaptive TRESH_BINARY C=-5', cv2.WINDOW_KEEPRATIO)
    cv2.resizeWindow('Adaptive TRESH_BINARY C=-5', 1280, 854)
    cv2.imshow("Adaptive TRESH_BINARY C=-5", thresh)
    thresh = cv2.adaptiveThreshold(image, 128,
adaptiveMethod=cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                   thresholdType=cv2.THRESH_BINARY_INV, blockSize=5,
C=-5)
    cv2.namedWindow('Adaptive TRESH_BINARY_INV C=-5', cv2.WINDOW_KEEPRATIO)
```

```
    cv2.resizeWindow('Adaptive TRESH_BINARY_INV C=-5', 1280, 854)
    cv2.imshow("Adaptive TRESH_BINARY_INV C=-5", thresh)
    cv2.waitKey(0)


""" 14 задание """
def fourteen():
    image = cv2.imread(f'{PATH}/first.jpg')
    borders = cv2.morphologyEx(image, cv2.MORPH_GRADIENT, np.ones((3, 3)))
    cv2.imshow("Borders", borders)
    segment = cv2.pyrMeanShiftFiltering(image, 30, 25)
    cv2.imshow("Default", image)
    cv2.imshow("pyrMeanShiftFiltering", segment)
    dst = cv2.addWeighted(borders, 0.5, segment, 0, 0)
    cv2.imshow("Single Image with smoothing and with edges", dst)
    cv2.waitKey(0)


""" 15 задание """
def fifteen():
    image = cv2.imread(f'{PATH}/first.jpg')
    kernel = np.array([[0,0,0,0,0,0,1,0,0,],
                       [0,0,0,0,0,0,0,0,0,],
                       [0,0,0,0,0,1,0,0,0,],
                       [0,0,0,0,0,0,0,0,0,],
                       [0,0,0,0,1,0,0,0,0,],
                       [0,0,0,0,0,0,0,0,0,],
                       [0,0,0,1,0,0,0,0,0,],
                       [0,0,0,0,0,0,0,0,0,],
                       [0,0,1,0,0,0,0,0,0]])

    dst = cv2.filter2D(image, cv2.CV_8U, kernel)

    cv2.imshow("Deleting 60 degrees lines", dst)
    cv2.waitKey(0)


""" 16 задание """
def sixteen():
    image = cv2.imread(f'{PATH}/first.jpg')
    cv2.imshow("Default", image)
    gaus_kernel = np.array([[1, 2, 1], [2, 4, 6], [1, 2, 1]]) / 16
    """ a """
    gaus_image = cv2.filter2D(image, cv2.CV_8U, gaus_kernel)
    cv2.imshow("Gaus", gaus_image)
    """ b """
    kernel_line = np.array([1, 2, 1]) / 4
    kernel_column = np.array([[1], [2], [1]]) / 4
    line_image = cv2.filter2D(image, cv2.CV_8U, kernel_line)
    column_line_image = cv2.filter2D(line_image, cv2.CV_8U, kernel_column)
    cv2.imshow("Column-Line", column_line_image)
    cv2.waitKey(0)
    """ c """
    """ Вместо 9 операций в a мы делаем 6 операций в b."""


""" 18 задания """
def eighteen():
    mishen = cv2.imread(f'{PATH}/mishen.jpg')
    mishen = cv2.resize(mishen, None, fx=0.8, fy=0.8)
    """ a """
    cv2.imshow("Mishen", mishen)
```

```
    cv2.waitKey(0)
    """ b """
    d1_3x3 = cv2.Sobel(mishen, -1, 1, 0, ksize=3)
    d2_3x3 = cv2.Sobel(mishen, -1, 2, 0, ksize=3)
    d1_5x5 = cv2.Sobel(mishen, -1, 1, 0, ksize=5)
    d2_5x5 = cv2.Sobel(mishen, -1, 2, 0, ksize=5)
    d1_9x9 = cv2.Sobel(mishen, -1, 1, 0, ksize=9)
    d2_9x9 = cv2.Sobel(mishen, -1, 2, 0, ksize=9)
    d1_13x13 = cv2.Sobel(mishen, -1, 1, 0, ksize=13)
    d2_13x13 = cv2.Sobel(mishen, -1, 2, 0, ksize=13)
    cv2.imshow("Default", mishen)
    cv2.imshow('dx1 3x3 and dx2 3x3', np.hstack((d1_3x3, d2_3x3)))
    cv2.imshow('dx1 5x5 and dx2 5x5', np.hstack((d1_5x5, d2_5x5)))
    cv2.imshow('dx1 9x9 and dx2 9x9', np.hstack((d1_9x9, d2_9x9)))
    cv2.imshow('dx1 13x13 and dx2 13x13', np.hstack((d1_13x13, d2_13x13)))
    cv2.waitKey(0)
    """
    Линия по оси x стирается, если дифференцировать по x, и по оси y если
дифференцировать по y.
    Но окружности в мешени стираются по оси y, если дифференцировать по x и по y,
если дифференцировать по x.
    """


if __name__ == '__main__':
    while True:
        print("Введите номер упражнения.")
        flag = input('>>')
        if flag == '12':
            twelve()
        if flag == '13':
            thirteen()
        if flag == '14':
            fourteen()
        if flag == '15':
            fifteen()
        if flag == '16':
            sixteen()
        if flag == '18':
            eighteen()
        if flag == 'exit':
            break
```

Images :

first.jpg



second.jpg

mishen.jpg