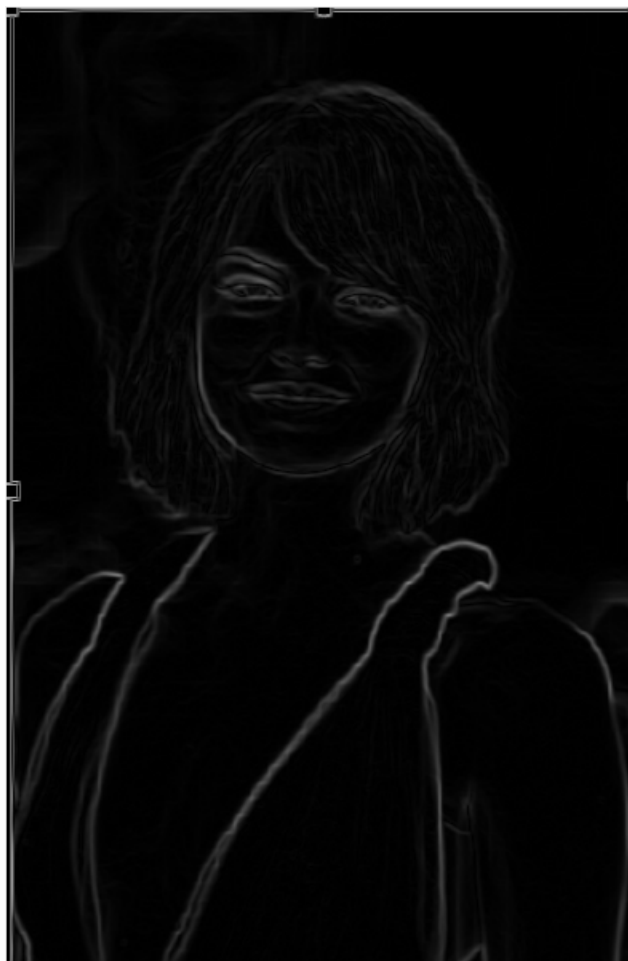


Input



Output



```

import cv2
import sys
import copy
import numpy as np
import itertools, math
import time
import matplotlib.pyplot as plt

prewitt_x = [[-1, 0, 1],
              [-1, 0, 1],
              [-1, 0, 1]]

prewitt_y = [[1, 1, 1],
              [0, 0, 0],
              [-1, -1, -1]]

def normalise(map):
    return map / map.max()

def apply_filter(filter, grayImage):
    height, width = grayImage.shape
    filtered_map = np.zeros((height, width))
    grayimage_ = np.pad(grayImage, (1, 1), 'edge')

    directions = [(-1, -1), (-1, 0), (-1, 1),
                  (0, -1), (0, 0), (0, 1),
                  (1, -1), (1, 0), (1, 1)]

    for i in range(1, height + 1):
        for j in range(1, width + 1):
            val = 0
            for k in directions:
                val += grayimage_[i + k[0], j + k[1]] * filter[1 + k[0]][1 + k[1]]
            filtered_map[i - 1, j - 1] = val

    return normalise(filtered_map)

def main(image_path):
    originalImage = cv2.imread(image_path)
    grayImage = cv2.cvtColor(originalImage, cv2.COLOR_BGR2GRAY)

```

```
map_x = apply_filter(rewitt_x, grayImage)
map_y = apply_filter(rewitt_y, grayImage)

final_edges = np.sqrt(np.multiply(map_x, map_x) + np.multiply(map_y, map_y))
final_edges = normalise(final_edges)

final_edges = np.float32(final_edges)
cv2.imshow('edges', final_edges)
cv2.waitKey()
cv2.destroyAllWindows()

if __name__ == '__main__':
    if len(sys.argv) != 2:
        print("Usage: python3 code.py <path_of_image>")
        exit(1)

    image_path = sys.argv[1]
    main(image_path)
```