Question 3

From the hints, we use the fish array and the binary net array as convolution vectors. The lengths will be 100n and n respectively. By multiplying them together, the resultant 101n length convolution will represent how many fish can be caught by casting the net between A[i - n : i]. This resultant vector will be reversed however. We can choose to reverse any of the three vectors (fish vector, net vector or resultant vector). Since the net is the shortest we will reverse that first before multiplying which takes o(n) time.

From the tutorial questions, convolution multiplication is calculated using Discrete Fourier Transformation which is done via computing the DFT, multiplying and then inversing the DFT. The specific algorithm we can use is Fast Fourier transform which runs in o(nlogn) time. This is done twice so the overall runtime will remain as o(nlogn).

Once we have the resulting convolution vector, we will go through it to find where result[i] is maximum. This is the point where the RIGHT of our net should be. Therefore the left of our net will be i - n (our net size is n). This takes o(n) time and again our overall runtime stays as o(nlogn).

Sample code

```
#!/usr/bin/python3
def main():
    # TESTING
    fish = [1,2,3,4,5,6]
    net = [1,1,0,1]

    assert(q1(fish, net) == 2)

    fish = [1,1,1,5,5,5,1,1,1]
    net = [1,1,1]

    assert(q1(fish, net) == 3)

def q1(fish, net):
    from scipy import signal

    # reverse net so our resultant convolution is not reversed
    net.reverse()

    # calculate convolution
    result = signal.fftconvolve(fish, net)

    # max argument is the RIGHT of the net
    # minus size off to get the LEFT of our net
    return result.argmax() - len(net) + 1

if __name__ == '__main__':
    main()
```