(1) 1. 3DES.DEC(k,c):

1. Check that $k=(k_1,k_2,k_3)$.

2. Output DES.DEC$(k_1,$ DES.ENC$(k_2,$ DES.DEC$(k_3,c)))$

Correctness:

We assume DES is correct, i.e., $\forall k \in K, m \in M$ DES.DEC$(k,$ DES.ENC$(k,m)) = m$.

Moreover, since DES is a Feistel construction, ENC and DEC algorithms are inverses of each other, we know DES.ENC$(k,$ DES.DEC$(k,m)) = m$. Then,

3DES.DEC$(k,$ 3DES.ENC$(k,m))$

$=$ DES.DEC$(k_1,$ DES.ENC$(k_2,$ DES.DEC$(k_3,$ DES.ENC$(k_3,$ DES.DEC$(k_2,$ DES.ENC$(k_1,m))))))$

$=$ DES.DEC$(k_1,$ DES.ENC$(k_2,$ DES.DEC$(k_2,$ DES.ENC$(k_1,m))))$

$=$ DES.DEC$(k_1,$ DES.ENC$(k_1,m)) = m$

2. For every key k, you can check if $m=$ DES.DEC(k,c) or $c=$ DES.ENC(k,m).

key space for DES

(a) $|K| = 2^{56} \geq 2^{54,8}$ ns $\approx$ a year

(b) $1024 = 2^{10} \Rightarrow 2^{56}/2^{10} = 2^{46} = 2^4 \cdot 2^{42}$ ns $\approx$ 16 hours

(c) K': key space for 3DES $\Rightarrow |K'| = 2^{3\times 56} = 2^{168} \geq 2^{88,5}$ ns $\approx$ age of universe

(2) 1. Take r as a global variable.

$L^G_{\text{PRF-real}}$

$k \leftarrow K$

Lookup$(x \in \{0,1\}^{in})$:

1. $y \leftarrow G(k,x)$
2. output y

$L^G_{\text{PRF-rand}}$

$T = [\ ]$

Lookup$(x \in \{0,1\}^{in})$:

1. If $T[x]$ is undefined then $T[x] \leftarrow \{0,1\}^{out}$
2. Output $T[x]$

2. $L^{G}_{PRF\text{-}real} \equiv \bar{L}^{G}_{PRF\text{-}real} \circ L^{F}_{PRF\text{-}real}$

where $\bar{L}^{G}_{PRF\text{-}real}$:

$k \leftarrow K$

1. $y \leftarrow \text{Lookup}(x)$
2. $y' \leftarrow y \oplus r$
3. output y'

3. $L^{G}_{PRF\text{-}real} \equiv \bar{L}^{G}_{PRF\text{-}real} \circ L^{F}_{PRF\text{-}real} \equiv \bar{L}^{G}_{PRF\text{-}real} \circ L^{F}_{PRF\text{-}rand} := L$

4. $\text{Lookup}(x) \oplus r$, output of L, can be replaced with the call to the random library since XORing with r will produce something random. (Remember OTP.)

Hence, $L \equiv L^{G}_{PRF\text{-}rand}$, and consequently $L^{G}_{PRF\text{-}real} \equiv L^{G}_{PRF\text{-}rand}$.

③ 1.

```python
def Birthdayproblem(q,N):
    p = 1
    for i in range(q-1):
        p = p * (1-(i/N))
    return 1-p

def Upperbound(q,N):
    return (q*(q-1))/2*N
```

2. Assume #people $= 22 = q$ and #days $= 365 = N$, then Birthdayproblem$(22, 365) \approx \frac{1}{2}$

3. The only way to distinguish F from G is if we find two inputs to the library such that Lookup(x) = Lookup(x'). So, prob. of distinguishing F from G is upper bounded by (success of finding two people with the same birthday) $\frac{q(q-1)}{2N}$ where $N = 2^{32}$. (length of output)

Compute for $q = 2, 2^8, 2^{12}, 2^{16}$.