Solution 1: discrete probabilistic inference

1 Q.1: sampling from a joint distribution

1.1 Compute $\mathbb{E}[(1+Y_1)^X]$

(Approach 1) Using LOTUS,

$$
\begin{aligned}
\mathbb{E}[(1+Y_i)^X] &= \sum_{y=0}^{1}\sum_{k=0}^{2}(1+y)^k \mathbb{P}(X=k, Y_i=y) && \text{(LOTUS)}\\
&= \sum_{y=0}^{1}\sum_{k=0}^{2}(1+y)^k \mathbb{P}(X=k)\cdot\mathbb{P}(Y_i=y \mid X=k)\\
&= \frac{1}{2+1}\sum_{y=0}^{1}\sum_{k=0}^{2}(1+y)^k \cdot\mathbb{P}(Y_i=y \mid X=k)\\
&= \frac{1}{2+1}\left[\sum_{k=0}^{2}(1+0)^k\mathbb{P}(Y_i=0 \mid X=k) + \sum_{k=0}^{2}(1+1)^k\mathbb{P}(Y_i=1 \mid X=k)\right]\\
&= \frac{1}{2+1}\left[\sum_{k=0}^{2}\left(1-\frac{k}{2}\right) + \sum_{k=0}^{2}2^k\frac{k}{2}\right]\\
&= \frac{1}{3}\left(1+\frac{1}{2}+2\cdot\frac{1}{2}+4\right)\\
&= \frac{13}{6}.
\end{aligned}
$$

(Approach 2) Using the law of total probability,

$$
\begin{aligned}
\mathbb{E}[(1+Y_i)^X] &= \sum_{k=0}^{2} \mathbb{E}[(1+Y_i)^k | X=k]\mathbb{P}(X=k) && \text{(total prob.)} \\
&= \frac{1}{2+1}\sum_{k=0}^{2} \mathbb{E}[(1+Y_i)^k | X=k] && (X \sim \text{Unif}\{0,1,2\}) \\
&= \frac{1}{2+1}\sum_{k=0}^{2} \left[1 \cdot \left(1-\frac{k}{2}\right) + 2^k \cdot \frac{k}{2}\right] && \text{(total prob.)} \\
&= \frac{1}{3}\left(1 + \frac{1}{2} + 2 \cdot \frac{1}{2} + 4\right) \\
&= \frac{13}{6}.
\end{aligned}
$$

1.2 Write a simulator

Define the function

```r
require(extraDistr)
```

```
Loading required package: extraDistr
```

```r
forward_sample = function(
    K, # sample X from {0...K}
    n  # number of Y values to sample
  ){
  X  = rdunif(1, min=0, max=K)
  p  = X/K
  Ys = rbern(n, p)
  list(X = X, Ys = Ys)
}
```

The outcome of the function is a named list: the first entry is the value of X, while the second entry is the vector of Y values.

Now use the function to draw a sample of (X, Y_1, Y_2, Y_3, Y_4) using a fixed seed

```r
set.seed(1224)
forward_sample(K=2, n=4)
```

```
$X
[1] 1

$Ys
[1] 0 1 1 1
```

1.3 Using the law of large numbers to check part 1

We can define a function that computes a Monte Carlo approximation of the expected value of our test function

```
mc_expectation = function(
    K, # sample X from {0...K}
    S  # number of Monte Carlo simulations
  ){
  fs = replicate(S, { # `replicate` evaluates an expression multiple times
    res = forward_sample(K=2, n=1) # draw X coin and a single Y value from it
    (1 + res$Ys[1])^(res$X)        # evaluate test function with simulated data
  })
  mean(fs) # return average of function across simulations
}
```

1.4 Compare the Monte Carlo simulation and the exact value

Run the Monte Carlo approximation using 100000 simulated (X, Y) pairs

```
mc_expectation(K=2, S=100000)
```

```
[1] 2.16396
```

Compare this to the exact value of $13/6 = 2.1\overline{6}$.

2 Q.2: computing a conditional

2.1 Mathematical expression

Using Bayes' rule, we can calculate the probability that we sampled $X = x$ given that we observed $Y = (0 \dots 0)$

$$
\begin{aligned}
\mathbb{P}(X = x | Y_{1:n} = (0, \dots, 0)) &= \frac{\mathbb{P}(X = x)\mathbb{P}(Y_{1:n} = (0, \dots, 0) | X = x)}{\sum_{k=0}^{K} \mathbb{P}(X = k)\mathbb{P}(Y_{1:n} = (0, \dots, 0) | X = k)} \\
&= \frac{\mathbb{P}(Y_{1:n} = (0, \dots, 0) | X = x)}{\sum_{k=0}^{K} \mathbb{P}(Y_{1:n} = (0, \dots, 0) | X = k)} && (X \sim \text{Unif}) \\
&= \frac{\left(1 - \frac{x}{K}\right)^n}{\sum_{k=0}^{K} \left(1 - \frac{k}{K}\right)^n}.
\end{aligned}
$$

For $x = K/2$, this simplifies to

$$
\mathbb{P}(X = K/2 | Y_{1:n} = (0, \dots, 0)) = \frac{2^{-n}}{\sum_{k=0}^{K} \left(1 - \frac{k}{K}\right)^n}.
$$

2.2 Evaluate the expression

Define a function that computes the expression for arbitrary (K, n)

```r
conditional_prob = function(K, n){
  num = 2^(-n)
  probs = (0:K)/K
  summands = (1-probs)^n # note: this as a vectorized statement
  den = sum(summands)
  num/den
}
```

Now evaluate it for the case $(K = 2, n = 4)$

```r
conditional_prob(K=2, n=4)
```

```
[1] 0.05882353
```

3 Q.3: non uniform prior on coin types

3.1 Write the joint distribution

$$
\begin{aligned}
X &\sim \text{Categorical}(\{0,1,2\},(1/100,98/100,1/100)) \\
Y_i &\overset{\text{iid}}{\sim} \text{Bern}(X/K)
\end{aligned}
$$

3.2 Compute the conditional probability

The formula for $K = 2$ is

$$
\begin{aligned}
\mathbb{P}(X=1|Y_{1:n}=[0\dots0]) &= \frac{\mathbb{P}(X=1)\mathbb{P}(Y_{1:n}=[0\dots0]|X=x)}{\sum_{k=0}^{2}\mathbb{P}(X=k)\mathbb{P}(Y_{1:n}=[0\dots0]|X=k)} \\
&= \frac{98\cdot 2^{-n}}{\sum_{k=0}^{2}(98\cdot\mathbb{1}\{k=1\}+1\cdot\mathbb{1}\{k\neq1\})\left(1-\frac{k}{2}\right)^{n}}.
\end{aligned}
$$

Define a function that computes this probability for any $n \in \mathbb{N}$

```
conditional_prob_nonunif = function(n){
  K = 2
  num = 98 * 2^(-n)
  summands = c(1, 98, 1) * (1-(0:K)/K)^n
  den = sum(summands)
  num/den
}
```

Now evaluate it for $n = 4$

```
conditional_prob_nonunif(n=4)
```

```
[1] 0.8596491
```

The value is higher than in Q.2 because we knew that there were more fair coins in the bag.

4 Q.4: a first posterior inference algorithm

The generalized formula is

$$\mathbb{P}(X = x|Y_{1:n} = [0 \dots 0]) = \frac{\mathbb{P}(X = x)\mathbb{P}(Y_{1:n} = [0 \dots 0]|X = x)}{\sum_{k=0}^{K} \mathbb{P}(X = k)\mathbb{P}(Y_{1:n} = [0 \dots 0]|X = k)}$$
$$= \frac{\rho_x \left(1 - \frac{x}{K}\right)^n}{\sum_{k=0}^{K} \rho_k \left(1 - \frac{k}{K}\right)^n}.$$

4.1 Write a function to evaluate the formula

```
posterior_given_four_heads = function(rho){
  n = 4
  K = length(rho) - 1
  pi_unnormalized = rho * (1-(0:K)/K)^n
  normalizing_constant = sum(pi_unnormalized)
  pi_unnormalized/normalizing_constant
}
```

4.2 Test the function

When $\rho \propto (1, 98, 1)$, the value for $X = 1$ should be exactly equal to the one given in Q.3

```
posterior_given_four_heads(c(1,98,1))[2] == conditional_prob_nonunif(n=4)
```

```
[1] TRUE
```

4.3 Show output for specific rho

```
posterior_given_four_heads(1:10)
```

```
 [1] 0.2018458699 0.2520227657 0.2215966774 0.1594831564 0.0961390555
 [6] 0.0472542686 0.0174434702 0.0039378557 0.0002768805 0.0000000000
```

5 Q.5: generalizing observations

5.1 Joint distribution

$$X \sim \text{Categorical}(\{0, \dots, K\}, \rho)$$
$$n_{\text{heads}}|X, n_{\text{obs}} \sim \text{Binom}(n_{\text{obs}}, X/K)$$

Then the posterior distribution becomes

$$\mathbb{P}(X = x|n_{\text{heads}}, n_{\text{obs}}) = \frac{\rho_x \mathbb{P}(\text{Binom}(n_{\text{obs}}, x/K) = n_{\text{heads}}|X = x)}{\sum_{k=0}^{K} \rho_k \mathbb{P}(\text{Binom}(n_{\text{obs}}, k/K) = n_{\text{heads}}|X = k)}$$

5.2 Write an R function

```
posterior = function(rho, n_heads, n_observations){
  n_tails = n_observations - n_heads
  K = length(rho) - 1

  # we can leverage the fact that `dbinom` is vectorized
  pi_unnormalized = rho * dbinom(n_tails, n_observations, (0:K)/K)
  normalizing_constant = sum(pi_unnormalized)
  pi_unnormalized/normalizing_constant
}
```

5.3 Test your code

Check that we recover the value from Q.4

```
all.equal(posterior(1:10,4,4), posterior_given_four_heads(1:10))
```

```
[1] TRUE
```

5.4 Show output for specific values

```
posterior(1:10,2,10)
```

```
 [1] 0.000000e+00 2.623172e-07 7.712124e-05 1.936196e-03 1.678830e-02
 [6] 7.685073e-02 2.168539e-01 3.780516e-01 3.094419e-01 0.000000e+00
```

6 Appendix: Alternative solutions

6.1 Q1

6.1.1 Alternative forward simulator

```
forward_sample = function() {
  x <- rdunif(1, min=0, max=2)
  y1 <- rbern(1, x/2)
  y2 <- rbern(1, x/2)
  y3 <- rbern(1, x/2)
  y4 <- rbern(1, x/2)
  c(x, y1, y2, y3, y4)
}
```

6.1.2 Alternative Monte Carlo estimation code

```
sum <- 0.0
n_iterations <- 10000
for (iteration in 1:n_iterations) {
  sample <- forward_sample()
  sum <- sum + (1+sample[2])^sample[1]
}
print(sum/n_iterations)
```

```
[1] 2.1841
```

6.1.3 General formula for $\mathbb{E}[(1+Y_1)^X]$

Using the law of total probability,

$$
\begin{aligned}
\mathbb{E}[(1+Y_i)^X] &= \sum_{k=0}^{K} \mathbb{E}[(1+Y_i)^k | X=k]\mathbb{P}(X=k) && \text{(total prob.)} \\
&= \frac{1}{K+1} \sum_{k=0}^{K} \mathbb{E}[(1+Y_i)^k | X=k] && (X \sim \text{Unif}\{0,1,\dots,K\}) \\
&= \frac{1}{K+1} \sum_{k=0}^{K} \left[1 \cdot \left(1-\frac{k}{K}\right) + 2^k \cdot \frac{k}{K}\right] && \text{(total prob.).}
\end{aligned}
$$

Let us separate the sum into two parts. For the first term,

$$\sum_{k=0}^{K}\left(1-\frac{k}{K}\right)=K+1-\frac{1}{K}\sum_{k=0}^{K}k=K+1-\frac{1}{K}\frac{K(K+1)}{2}=\frac{K+1}{2}.$$

For the second term, we use the fact that differentiation is a linear operator

$$\frac{1}{K}\sum_{k=0}^{K}k2^{k}=\frac{2}{K}\sum_{k=1}^{K}k2^{k-1}=\frac{2}{K}\sum_{k=1}^{K}\left[\frac{\mathrm{d}}{\mathrm{d}x}x^{k}\Big|_{x=2}\right]=\frac{2}{K}\frac{\mathrm{d}}{\mathrm{d}x}\left[\sum_{k=1}^{K}x^{k}\right]\Bigg|_{x=2}. \qquad (\star)$$

Using the formula for the geometric sum,

$$\sum_{k=1}^{K}x^{k}=x\sum_{k=1}^{K}x^{k-1}=\frac{x^{K+1}-x}{x-1}.$$

Differentiating the above,

$$\frac{\mathrm{d}}{\mathrm{d}x}\sum_{k=1}^{K}x^{k}=\frac{\mathrm{d}}{\mathrm{d}x}\left[\frac{x^{K+1}-x}{x-1}\right]=\frac{(K+1)x^{K}-1}{x-1}-\frac{x^{K+1}-x}{(x-1)^{2}}.$$

Substituting $x=2$

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[\sum_{k=1}^{K}x^{k}\right]\Bigg|_{x=2}=(K+1)2^{K}-1-2^{K+1}+2=1+2^{K}(K-1)$$

Putting this back into $(\star)$ gives

$$\frac{1}{K}\sum_{k=0}^{K}k2^{k}=\frac{2(1+2^{K}(K-1))}{K}.$$

Now we replace the value of the two sums in the initial expression to obtain

$$\mathbb{E}[(1+Y_i)^X]=\frac{1}{K+1}\sum_{k=0}^{K}\left[1\cdot\left(1-\frac{k}{K}\right)+2^{k}\cdot\frac{k}{K}\right]=\frac{1}{2}+\frac{2(1+2^{K}(K-1))}{K(K+1)}$$

6.2 Q4: Alternative posterior function

```r
posterior_given_four_heads <- function(rho) {
  K <- length(rho) - 1
  gamma <- rep(0, K)
  for (k in 0:K) {
    gamma[k+1] <- rho[k+1] * (1-k/K)^4
  }
  gamma / sum(gamma)
}
```

6.3 Q5: Alternative posterior function

```
posterior = function(rho, n_heads, n_observations) {
  n_tails = n_observations - n_heads
  K <- length(rho) - 1
  gamma <- rep(0, K)
  for (k in 0:K) {
    gamma[k+1] <- rho[k+1] * dbinom(n_tails, n_observations, k/K)
  }
  gamma / sum(gamma)
}
```