1. P(at least 2 have common birthday) = 1 − P(no common birthdays)

$$= 1 - (1)\left(\frac{364}{365}\right)\left(\frac{363}{365}\right)\cdots\left(\frac{365-(n-1)}{365}\right) = 1 - \frac{365!}{(365-n)!\,365^n}$$

$P(23) = 0.507$ (see MATLAB code)

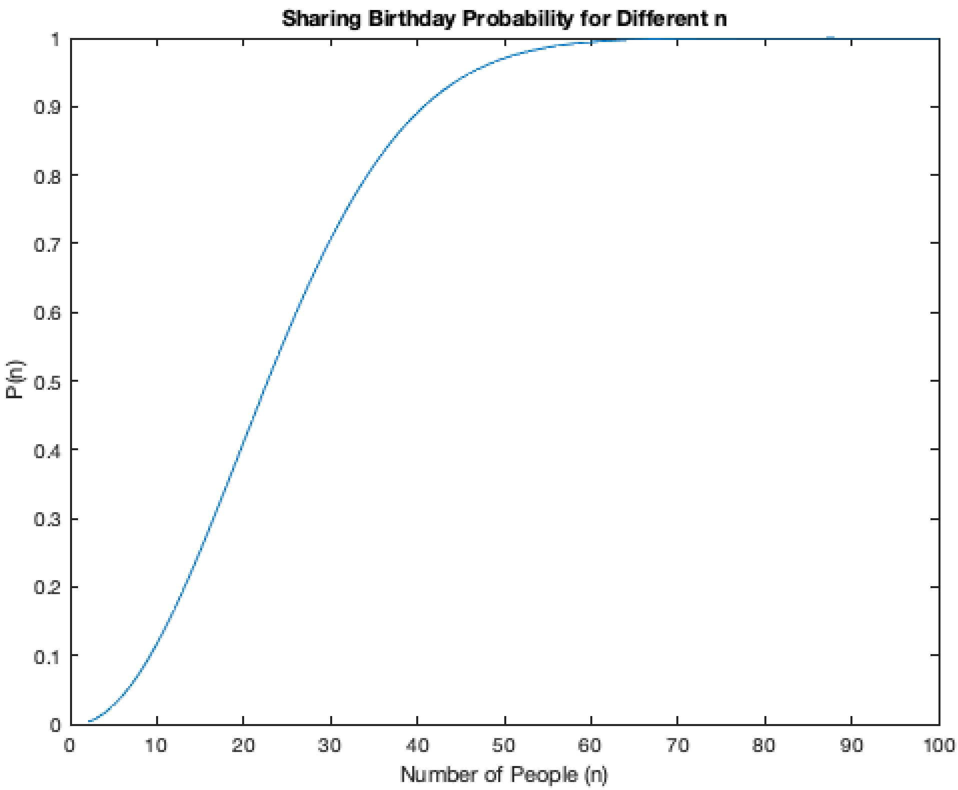
```
% populate x and y for plotting
x = 2:100;
y = [];
for i = 2:100
    y(i-1) = birthday_probability(i);
end

% plot
plot(x, y);
title("Sharing Birthday Probability for Different n");
xlabel("Number of People (n)");
ylabel("P(n)");

% print birthday probability for 23 people
fprintf("P(23): %d\n", birthday_probability(23));

% define function to return birthday probability for given n
function p = birthday_probability(n)
    product = 1.0;
    for i = 1:n
        product = product * (366 - i) / 365;
    end
    p = 1 - product;
end
```

```
P(23): 5.072972e-01
```



Published with MATLAB® R2023b

2. $$P(\text{disease} \mid \text{tested positive}) = \frac{P(\text{disease} \cap \text{tested positive})}{P(\text{tested positive})}$$

$$= \frac{1/1000 \cdot 0.99}{1/1000 \cdot 0.99 + 999/1000 \cdot 0.02} = \boxed{0.0472}$$

3. a) $E[X] = np$

b) See code.

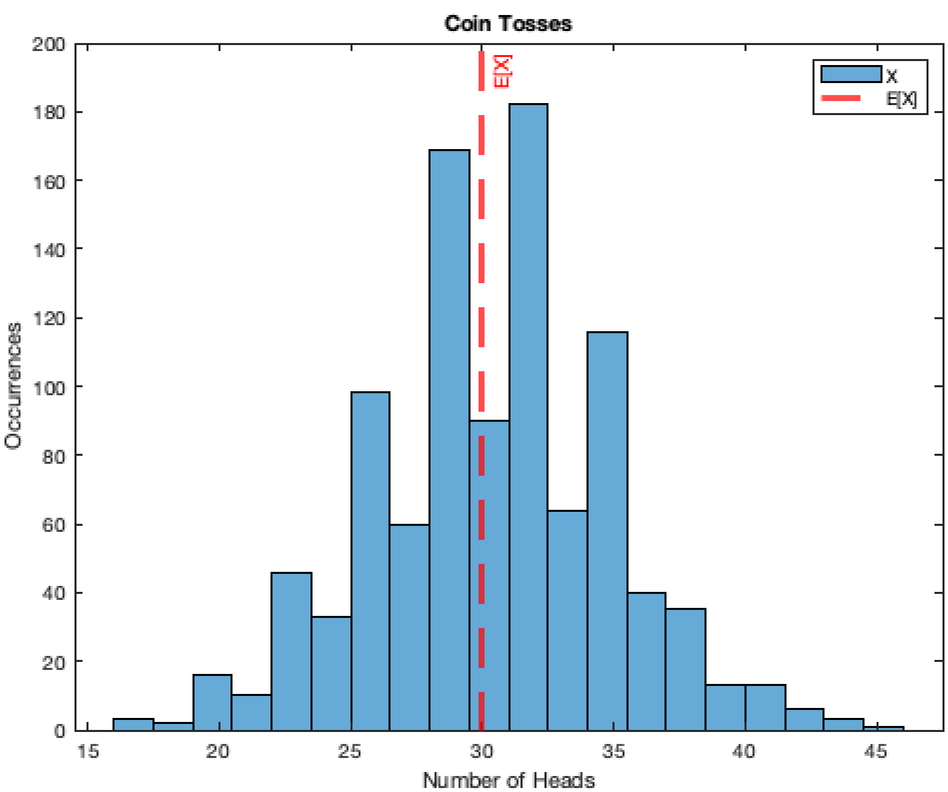
```
p = 0.3;
n = 100;
r = 10^3;

X = binornd(n, p, r, 1);

histogram(X, 20);
xline(30, '--r', {'E[X]'}, LineWidth=3);
title("Coin Tosses");
xlabel("Number of Heads");
ylabel("Occurrences");
legend("X", "E[X]")

% compare average X to E[X]:
fprintf("Average number of heads: %.2f\n", mean(X));
fprintf("Expected number of heads: %d\n", n * p);
```

```
Average number of heads: 30.13
Expected number of heads: 30
```



Published with MATLAB® R2023b

4. See code.

```
% define A, B, and AB
A = [2 4 6];
B = [1 2 3 4];
AB = [2 4];

% count occurrences
As = 0.0;
Bs = 0.0;
ABs = 0.0;

% run simulations
simulations = 10^4;
for i = 1:simulations
    die_roll = unidrnd(6);
    if (ismember(die_roll, A))
        As = As + 1;
    end
    if (ismember(die_roll, B))
        Bs = Bs + 1;
    end
    if (ismember(die_roll, AB))
        ABs = ABs + 1;
    end
end
As = As / simulations;
Bs = Bs / simulations;
ABs = ABs / simulations;

fprintf("P(A): %.3f\n", As);
fprintf("P(B): %.3f\n", Bs);
fprintf("P(AB): %.3f\n", ABs);
fprintf("P(A)P(B): %.3f\n", As*Bs);
```

```
P(A): 0.502
P(B): 0.663
P(AB): 0.337
P(A)P(B): 0.333
```

Published with MATLAB® R2023b

5. a) $Z \sim U[0,1]$, $X = F^{-1}(Z)$

The CDF of X is

$$P(X \le x) = P(F^{-1}(Z) \le x)$$
$$= P(Z \le F(x))$$
$$= F(x)$$

since Z is a standard uniform distribution, $P(Z \le z) = z$ for any $z \in [0,1]$.

Hence, $X \sim F$.

b) $X \sim Exp(\lambda)$

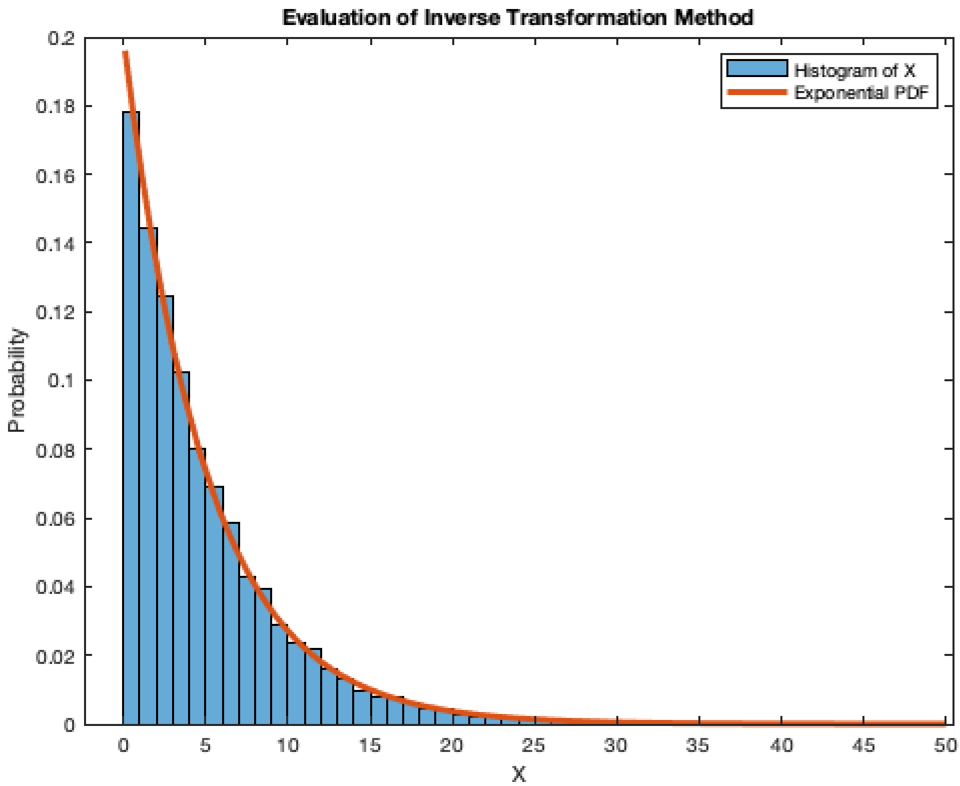
$X = F^{-1}(Z) \Rightarrow F(X) = Z$

$$F(x) = 1 - e^{-x/\lambda}$$
$$Z = 1 - e^{-x/\lambda}$$
$$e^{-x/\lambda} = 1 - Z$$
$$-\frac{x}{\lambda} = \ln(1-Z)$$
$$x = -\lambda \ln(1-Z)$$

```
N = 10^4;
lambda = 5;

Z = rand(N, 1);
X = -lambda * log(1 - Z);

histogram(X, 'Normalization', 'pdf');
hold on
fplot(@(x) exppdf(x, lambda), [0.1, 50], LineWidth=3);
hold off
xlabel("X");
ylabel("Probability");
title("Evaluation of Inverse Transformation Method");
legend("Histogram of X", "Exponential PDF");
```



Published with MATLAB® R2023b

6. Let L be the length of piece 1.

$X = \max(L, 1-L)$

$$X = \begin{cases} L & \text{if } L \geq \frac{1}{2} \\ 1-L & \text{if } L < \frac{1}{2} \end{cases}$$

$$E[X] = \int_0^{1/2} (1-\ell)\, d\ell + \int_{1/2}^1 \ell\, d\ell = 0.75$$

7. Algorithm I:

$$P(X=0) = P(Y_2=0 \mid Y_1 \neq Y_2) = \frac{P(Y_2=0 \cap Y_1 \neq Y_2)}{P(Y_1 \neq Y_2)}$$

$$= \frac{P(Y_2=0 \cap Y_1=1)}{P(Y_1 \neq Y_2)} = \frac{P(Y_2=0 \cap Y_1=1)}{P(Y_2=0 \cap Y_1=1) + P(Y_2=1 \cap Y_1=0)}$$

$$= \frac{p(1-p)}{p(1-p) + (1-p)p} = \frac{1}{2}$$

Since $P(X=0) = \frac{1}{2}$ and $P(X=1) = 1 - P(X=0) = \frac{1}{2}$, Algorithm I does simulate a fair coin.

Algorithm II:

Let the outcome of the first toss be denoted Y_1.

If $Y_1 = 1$, then the coin will continue to be tossed until it lands on 0, and X becomes 0. Therefore, the tosses follow the form 11…10, where there are k 1's and $k \geq 1$. Mathematically,

$$P(X=0) = \sum_{k=1}^{\infty} (1-p)^k p$$

If $Y_1 = 0$, then the coin will continue to be tossed until it lands on 1, and X becomes 1. Therefore, the tosses follow the form 00…01, where there are k 0's and $k \geq 1$. Mathematically,

$$P(X=1) = \sum_{k=1}^{\infty} p^k (1-p)$$

Since $p \neq 1-p$, $\sum_{k=1}^{\infty} (1-p)^k p \neq \sum_{k=1}^{\infty} p^k (1-p)$, so $P(X=0) \neq P(X=1)$

Therefore, Algorithm II is not a simulation of an unbiased coin.

8. Let X = number of pokemon balls opened to have n different types
Let X_i = number of pokemon balls opened to get the ith unique type
s.t. $X = X_1 + X_2 + X_3 + \dots + X_n$

$P(\text{finding 1st unique pokemon at a given trial}) = \frac{n}{n}$
$P(\text{finding 2nd unique pokemon at a given trial}) = \frac{n-1}{n}$
$\vdots$
$P(\text{finding } i^{th} \text{ unique pokemon at a given trial}) = \frac{n+1-i}{n}$

$X_i \sim Geo(P(\text{finding ith unique pokemon at a given trial})) = Geo\left(\frac{n+1-i}{n}\right)$
$E[X_i] = \frac{n}{n+1-i}$

$$E[X] = E[X_1 + X_2 + \dots + X_n] = E[X_1] + E[X_2] + \dots + E[X_n]$$
$$= \sum_{i=1}^{n} \frac{n}{n+1-i}$$
$$= \sum_{i=0}^{n-1} \frac{n}{n-i}$$