1. We list all possible outcomes:

\[ S = \{ HH, THH, HTTH, TTHH, TTTTH, THHTT, HTTTH, HTTHH, TTTHH, THTHH, HTTHT, TTHTH, HTHTH, THTTH, HTTTH, TTTTH \} \]

The probabilities are: \( P(HH) = \frac{1}{4}, P(THH) = \frac{1}{8}, P(HTTHH) = P(TTHH) = \frac{1}{16} \), and each of the 16 outcomes with five tosses has probability \( \frac{1}{32} \). (Note that \( P(S) = 1 \), as it must.)

2. (a) \( F \cap E^c \cap G^c \), (b) \( E \cap F \cap G^c \), (c) \( E \cup F \cup G \), (d) \( (E \cap F) \cup (E \cap G) \cup (F \cap G) \), (e) \( E \cap F \cap G \), (f) \( E^c \cap F^c \cap G^c \), (g) \( (E \cap F)^c \cap (E \cap G)^c \cap (F \cap G)^c \), (h) \( (E \cap F \cap G)^c \). [4]

3. (a) There are \( \binom{12}{40} \) ways to choose 12 marked beetles from 40, \( \binom{n-40}{60} \) ways to choose 48 unmarked beetles from \( n-40 \), and \( \binom{n}{60} \) ways to choose 60 beetles from \( n \), so

\[ L(n) = \frac{\binom{12}{40} \binom{n-40}{60}}{\binom{n}{60}}. \]

(b) \[ \frac{L(n)}{L(n-1)} = \frac{\binom{n-40}{60} \binom{n-1}{48}}{\binom{n}{60} \binom{n-41}{48}} = \frac{(n-40)(n-60)}{(n-88)n} = \frac{n^2 - 100n + 2400}{n^2 - 88n}. \]

Thus \( L(n)/L(n-1) \leq 1 \) is equivalent to \( n^2 - 100n + 2400 \leq n^2 - 88n \), or \( n \geq 200 \). Therefore \( L(n) \) is increasing when \( n \leq 200 \) and decreasing for \( n \geq 200 \), and \( L(n) \) is maximal when \( n = 200 \).

Note: This is consistent with the guess that the proportion of marked beetles in the sample should equal the proportion in the pond, i.e., \( \frac{12}{60} = \frac{40}{n} \), which again gives \( n = 200 \).

4. (a) i. \[ \frac{1}{\binom{5}{2}} \binom{13}{1} \binom{12}{3} \binom{4}{2} \binom{4}{1}^3 = 0.4226, \]

where the factors in the numerator are respectively the number of ways to choose: value of \( a \), values \{b, c, d\}, suits for the pair, suits for \( b, c, d \). [2]

ii. \[ \frac{1}{\binom{5}{2}} \binom{13}{2} \binom{11}{1} \binom{4}{2} \binom{4}{1}^2 = 0.0475, \]

where the factors in the numerator are respectively the number of ways to choose: values \{a, b\}, value of \( c \), suits for \( a, b \), suit for \( c \). [2]

(b) Think of the dice as having different colours, so the sample space has size \( 6^3 \).

i. \[ \frac{1}{6^3} \binom{6}{1} \binom{5}{3} \binom{5}{2} \binom{3}{1} \binom{2}{1} \binom{1}{1} = 0.4630, \]

where the binomial coefficients respectively count the number of ways to choose: \( a, \{b, c, d\} \), 2 dice for value \( a \), 1 die for value \( b \), 1 die for value \( c \), 1 die for value \( d \). [2]

ii. \[ \frac{1}{6^5} \binom{6}{2} \binom{4}{1} \binom{5}{2} \binom{3}{2} \binom{1}{1} = 0.2315, \]

where the binomial coefficients respectively count the number of ways to choose: \{a, b\}, c, 2 dice for value \( a \), 2 dice for value \( b \), 1 die for value \( c \). [2]

5. (a) A configuration is equivalent to a list of \( n+(m-1) \) objects with \( m-1 \) of them specified as barriers, so the number of configurations is \( \binom{n+m-1}{m-1} \). [2]

(b) A configuration is equivalent to specifying which of the \( m \) urns contain one of the \( n \) balls, so there are \( \binom{m}{n} \) configurations. [2]
6. (a)

```python
import numpy as np

def Birthday(n):
    # Generate the random birthdays
    a = np.random.randint(1,365,size=n)
    y = 0

    # Check if there are duplicate entries:
    # Create indices k, l to run over pairs of elements of the vector a
    k = 1
    while ((y == 0) & (k < n)):
        l = k + 1
        while ((y == 0) & (l <= n)):
            if a[1,k] == a[1,l]:
                y = 1
            else:
                l = l+1
        k = k + 1
    return(y)
```

(b)

```python
def X(n):
    i = 1
    s = 0
    while i < 1000:
        i += 1
        s = s + Birthday(n)
    return(s / 1000)

def Y(n):
    Y = 1 - (np.math.factorial(365) / ((365**n) * np.math.factorial(365-n)))
    return(Y)
```

```python
def Iterator(func):
    L = []
    for i in range(2,61):
        L.append(func(i))
    return(L)
```

(c)

```python
import matplotlib.pyplot as plt

# Commands for plotting:
plt.title('Birthday Problem')
plt.plot(n,Iterator(Y),label = "Y(n)")
plt.plot(n,Iterator(X),label = "X(n)")
xlabel('Number of people')
ylabel('Probability of a match')
plt.show()
```