Solution 3

Group A

1) Let I_i be the indicator if i-th person and the next (clockwise) person forms a boy-girl neighbor. Then
$$E\sum_{i=1}^{16} I_i = \sum_{i=1}^{16} EI_i = 16 * \frac{9*7*2}{16*15} = \frac{42}{5}.$$
2) The expectation of the last toss is 3.5, that means we should continue after second toss only if second toss is less than 3.5. That means if the second toss is $1, 2, 3$ we should continue, otherwise stop. The maximum expectation for last two tosses is 4.25, thus if the first toss is 5 or 6 we should stop, otherwise we continue. The maximum expectation for this problem is 4.67.
3) To generate the random numbers, we first need to generate uniform distribution. Once we have uniform distribution, we can transform it to any distribution. There are two main methods to generate uniform distribution: physical method and computation method.

   Computer can sensor some random phenomena, e.g., thermal noise, radio noise., audio noise, a computer can transform these random information to random number. For example, to mimic a fair coin, we can use last digit of the current time to module 2. However, physical method has asymmetries and systematic biases that make their outcomes not uniformly random. Also, the rate to generate is restricted due to the sampling limit of sensor.

   The second method actually generate pseudo-random number. Such generator creates long runs of numbers with good random properties but eventually the sequence repeats. One of the most common pseudo-random number is the linear congruential generator, which uses the recurrence
$$X_{n+1} = (aX_n + b) \mod m.$$
   a, b and m are large numbers which are prespecified.

   To test the goodness, we can use goodness-of-fit tests, which include χ^2 test for discrete distribution and also KS test for continuous distribution. Also the generator should generate i.i.d sequence, so a good generator should have low autocorrelation.
4) First let's see how to mimic one dice. Let the result of a coin flip be 0 or 1, 3 tosses can be encoded as 3 bits where the i-th bit is the i-th toss. The encoded number ranges from 0 to 7, if we get 0 and 1 we restart immediately, otherwise map the result as a dice uniquely. The expectation of tosses needed is $\frac{11}{3}$.

   Notice that if we drop 0 and 7, the expectation of tosses is $4 > 11/3$. The reason is that the first two digits for 0 and 1 are the same, so if we restart we do it right after 2 tosses.

   Follow the same logic, if we mimic m dices, we can use n tosses of a coin where n is the smallest number s.t. $6^n < 2^n$. The outcomes can be encoded to 0 to $2^n - 1$. We drop 0 to $2^n - 6^m - 1$ and map the rest uniquely with the result of m dices. In this way, the expectation of tosses is minimized.