**A Cat, a Parrot, and a Bag of Seed:**

The problem here is the delivery each individual parcel without damaging the other two parcels. The man is the protector of all three parcels. Here is my first flow chart of facts:

* Cat can eat the parrot (cat is at the top of the food chain).
* Parrot can eat the seed (parrot is in the middle of the food chain).
* The seed is at the bottom of the food chain.
* Only the cat and seed can be safely paired together.

The main goal here is to safely transport and deliver individually three parcels (cat, parrot and a bag of seed) across a river. At two given times two parcels are on one side of the river leaving one parcel on the other side of the river. The protector (the man) can only be in one place at a given time. Now the sub goals are the delivery of each parcel individually while leaving two parcels unprotected at two given times. So how do we do this? If you transport the seed this would leave the cat and parrot alone for destruction. If you transport the cat this leaves the seed in jeopardy with the parrot. When you transport the parrot the seed and cat are safe. But when return to transport and deliver the either the seed or cat we have problems on the other side of the river as they are left alone without the protector. I am assuming that the man needs to be presented on this boat at all times. Otherwise who would guide this boat from point A to point B?

There is no solution of complete safety of the three parcels. There is always a risk factor of 33 percent. There are three individual trips needed with one event that presents danger. I believe the only possible solution is leaving the cat and parrot alone during these trips. The seed has no defense against the parrot. While the cat and parrot can fight it out while left alone. If they are left alone for say 30 minutes there is a chance both parties would survive the fight or maybe complete peace among these two would prevail.

**Socks in the dark**

The problem here is the random selection of a pair of same color socks without visible light. This situation is similar to winning a lottery, random selection. Our goal is to determine what is the smallest number of socks needed in a predetermined pool to:

* Randomly select one pair of same color socks.
* Selecting three pairs of same color socks.

This selection comes from a pool of the following sock inventory:

* 5 pairs of black socks (10 individual black socks-50% of the inventory)
* 3 pairs of brown socks (6 individual brown socks-30% of the inventory)
* 2 pairs of white socks (4 individual white socks-20% of the inventory)

This gives us a lottery count of 20 individual socks. To guarantee one pair of same color socks you only need 2 socks of the same color placed inside this drawer. When you place two white socks in the drawer you would acquire 100% selection rate from your selections.

Now for the other problem of selecting one matching pair of each color (total of three colors) lowers our lottery rate significantly. I would recommend 3 pairs of socks (one pair of each color) leaving us with a random selection of 6 individual socks. I believe that there would be a 16% chance of success of randomly selecting 3 like pairs of matching socks.

**Predicting Fingers**

The problem here is developing a system of predicting an end count based on sequential system of the following:

* For the first five numbers, she always starts the numbering system with the thumb (number 1).
* Followed by the first finger (number2), middle finger (number 3), ring finger (number 4)ending with the little finger (number 5).

Then the next round of 5 numbers goes like this:

* Ringer finger (number 6), middle finger (number 7), first finger (number 8), the thumb (number 9) and the first finger (number10).

We have three main goals here: based on this sequence of instructions which finger will be identified with the ending count of 10, 100 and 1000.

I believe we need to build a sequential code of the five-finger counts. Then we would implement a conditional code to change up the sequence with a set of instructions to reverse the order of count. Then back to the sequential code for a count of five. This program will last until the first total equals 10. Then we change the next program to hit a total 100. While the third program would hit a total of 1000.