Problem #1

Trail length L with stopping points x1, x2, … xn., assuming that distance from x1 to xn is L, and no detours or alternate pathways.

Only *d* miles per night.

**It does minimize the number of camps the vampires must make.**

Assume xa – x1 < d for a< n, and xa+1 – x1 > d. In other words, the vampires must make a stop at xa starting at x1 if they wanted to utilize the full distance of d, and consider the set A = {xa+1, xa+2, …, xa+ α} for α < n, α < d, and xa+ α – xa <= d, but xa + α + 1 – xa >d. In other words, A contains the set of all possible points which the vampires can reach after having rested at xa.

Let set B = {set of all stopping points | vampires can reach after having rested at xb where 1 < b < a}. In other words, B = {xb+1, xb+2, … ,xb+ β} for β < n, β < d, and xb+ β – xb <= d, and point xb is before point xa, assuming the vampires stopped early on the track without going their full potential length starting at x1. We need to show that there does not exist one point in B which cannot be reached having rested at xa.

Case 1: B = {xa} or = {xb, xb+1,…,xb+x|b+x <a}. Well, A = {xa+1, xa+2, …, xa+ α}, and there does not exist øi ∈ A and ∆j ∈ B where i < j. So the points in A are farther than B, and so traveling the farthest in the night yields farther points the next night.

Case 2: points that exist in B but is after xa, which case, b+ β > a

Assume there does exist one point in B which can be reached having rested at xb which cannot be reached having rested at xa. Therefore, xb+ β ⊄ A. **But** xb+ β  has to be in A as xa is farther than xb, and since xb+ β - xb <=d, xb+ β - xa <=d must be true since b < a. Therefore, all the points in B can be reached if the vampires went the farthest distance assuming the points in B are ahead and not behind(, for if they were behind, refer to case 1.)

Therefore, traveling to xa, the max distance yields the farthest points the night after. The way which yields the farthest travel distance the vampires could make on any given night is the way which minimizes the number of camps to travel (given that there are other algorithms which could yield the same result on multiple occasions as this algorithm, this algorithm is the fastest and minimalist on infinite accounts).

Proved!

Problem #2

Solution, sudo code:

For every vampire residence,

check if surrounded by at least one blood bank.

If not, place one blood bank four miles down that vampire residence.

Repeat.

**Works**. Every vampire residence is occupied by one blood bank, as, in the sudo code, the check which triggers if it is not occupied by at least one blood bank places a blood bank within its radius.

**Big O (n)**