Exercise 2:

a) Story points per person = 32/3 = 10.67
Story points with 4 people working 100% = 42.68
Story points with 1 person working 80% = 8.54
Total story points with new team = 42.68 + 8.54 = 51.22 story points

b) I would estimate focus factor for a brand new team by using story points as a metric. I would assume that the amount of story points that they can complete is directly related to their experience. If I had a beginner developer on the team, I would estimate that he could complete less story points than the average developer. Maybe around 60% of the man days that he worked would result in story points. If I had a senior developer, I would assume that he could probably complete more story points per man day worked. Maybe around 80% of the man days that he worked would result in story points. I would then place everyone somewhere in that spectrum. I would then take into account how much time everyone was spending on this team. If someone was spending less than 100% of their time on this project, I would discount their story points further.

   If at all possible, I would try to get information about the employee both from himself and others he has worked with in the past, in order to assess his focus factor on other teams.

   After the first sprint, I would take a look at the metrics and see how my estimates actually lined up with reality. I would then adjust the estimated story points for next sprint based off of the ones completed in this sprint.

   After a while, I would let the law of averages take over, and begin estimating story points for a given week based off of the average of previous weeks.

c) Another way to estimate story points for a project would be to have the most senior developer on the team take a look at all of the different tasks that must be completed during the sprint, and assign a certain number of story points to each one. This would be a worse way of assigning story points because even though this is the most senior developer, he may still miss things that others may be able to see. In addition, if his estimate is wrong, everyone on the team can blame him, instead of being forced to take responsibility. We are also in a sense punishing the senior developer for being senior by adding more work to his load.