For the Line\_of\_Credit question, I design a class in ruby. It has fields credit\_limit, apr, principle\_balance, remain limit and transactions. There are draw method for drawing money, pay\_back method for paying back money and calculation method for calculating the pay amount at the end of each month.

Users can read all the fields of the line\_of\_credit, but can only write to credit\_limit and apr fields. These two fields need to be initialized by the user. Calculation method will return the amount of total payment at the end of month. I take a special case into consideration, when the user possibly pay back more money than principle balance at the middle of the month, my line\_of\_credit will print promt telling that “pay\_back amount larger than principle balance in the card”. Then the principle balance will become negative. In the calculation at the end of month, if the result is negative, I know that user pay\_back more than needed to the credit card before the end of month, so I will print out this amount and tell “Money left on this card: ”. If not, I just print the total amount that user should pay.

**Test Cases: I include 4 scenarios for testing.**

First two are the scenarios from the questions, and my output results are the same.

Scenario 3 is that user wants to draw more money than the remain\_limit in the credit card. I will print necessary message.

Scenario 4 is that user pay back more than principle\_balance at day 20. My line\_of\_credit will print out some messages and show user’s remain money at the end of month.