1) Facebook is designing a tool for forecasting the results of US presidential elections. This tool assumes that the vote of voter **v** is influenced by the relationships between **v** and other nodes in the network: if **v** has many friends voting for Democrats or many enemies voting for Republicans, then it is more likely that **v** votes for Democrats.

   The amount of data owned by Facebook allows for powerful sentiment analysis that provides a very precise estimate of the level of enmity $\mathbf{e_{vw} \geq 0}$ for each pair of voters **v** and **w** that are friends on this social network.

   The Facebook tool groups voters for Democrats and Republicans so that the level of enmity within each group is low, and the level of enmity among the two groups is as large as possible. Hence, the level of enmity in a set of voters is computed as the sum of enmities among each pair of these voters that are friends on the social network.

   For example, suppose that there are **n = 5** voters named **(a, b, c, d, e)** and the following pairs are friends on Facebook: **(a, b)**, **(a, c)**, **(b, d)**, **(c, d)**, and **(d, e)**. Facebook assigned to these pairs the following enmity levels: **2**, **4**, **3**, **5**, **3**.
   The Facebook tool then states that Democrats voters are **a** and **d**, whereas Republicans voters are **b**, **c**, and **e**. Indeed, in this case each group of voters has enmity level 0, and the total enmity level among the two groups is 17.

   Provide a function ***facebook_enmy(V, E)*** that takes in input:
   - a Python set **V** of voters, and
   - a Python dictionary **E** whose keys are Python tuples representing pairs of voters that have a friendship relationship on Facebook, and whose values represent the enmity level that Facebook assigned to the corresponding pair,

   and returns two Python sets, **D** and **R**, corresponding to voters for Democrats and Republicans, respectively.

   The function will be tested on an unknown instance. Among the 20 projects for which the running time of the function ***facebook_enmy*** is the smallest, the 10 projects for which the enmity level between **D** and **R** is the largest will receive a bonus point.

   Similarly, among the 20 projects for which the enmity level between **D** and **R** is the largest, the 10 projects for which the running time of the function ***facebook_enmy*** is the smallest will receive a bonus point.

2) At some times, Facebook decided to change the sentiment analysis algorithm, and to compute the level of friendship $\mathbf{f_{vw} \geq 0}$ for each pair of voters **v** and **w** that are friends on this social network. Note that levels of friendship and enmity are unrelated.

   To this reason, Facebook needs to adapt its tool for forecasting the results of US Presidential elections. The Facebook tool now needs to group voters for Democrats and Republicans so that the level of friendship within each group is large, and the level of friendship among the two groups is as low as possible. (As for enmity, the level of friendship in a set of voters is computed as the sum of friendships among each pair of these voters that are friends on the social network).

Moreover, Facebook developed a new sentiment analysis algorithm that assigns to each node **v** the likelihood $\mathbf{d}_v$ that **v** votes for Democrats and the likelihood $\mathbf{r}_v$ that **v** votes for Republicans. The improved Facebook tool hence requires also to maximize the total likelihood of returned groups, where the total likelihood is the sum over all voters **v** of the likelihood that v votes for the candidate of the group at which it is assigned.

For example, suppose that there are **n = 5** voters named **(a, b, c, d, e)** and the following pairs are friends on Facebook: **(a, b)**, **(a, c)**, **(b, d)**, **(c, d)**, and **(d, e)**. Facebook assigned to these pairs the following friendship levels: **2**, **4**, **3**, **5**, **3**. Moreover, it assigned the following likelihoods for Democrats: **1**, **3**, **1**, **2**, **2**, and the following likelihoods for Republicans: **0**, **2**, **3**, **1**, **4**.
The Facebook tool then states that all voters vote for Republicans. Indeed, in this case the total likelihood is 10 and friendship among the two groups is 0.

Provide a function ***facebook_friend(V, E)*** that takes in input:

- a Python dictionary **V** whose keys represent voters, and values are Python tuples with the first entry being the likelihood for Democrats and the second being the likelihood for Republicans;
- a Python dictionary **E** whose keys represent pairs of voters that have a friendness relationship on Facebook, and whose values represent the friendship level that Facebook assigned to the corresponding pair,

and returns two Python sets, **D** and **R**, corresponding to voters for Democrats and Republicans, respectively.

The function will be tested on an unknown instance. The 20 projects for which the running time of the function ***facebook_friend*** is the smallest will receive a bonus point.

NOTES:
The functions must be implemented in a file ***facebook.py***.
Each project may receive at most one bonus point.
Solutions must work regardless of as a voter is represented (e.g., regardless whether voters are represented as integers, strings, or some other object).