24.3 -6

1)

a. main idea: This problem can be interpreted as a graph problem: Each currency is a node and each possibility of exchange between two currencies is an edge. The edges are weighted by the exchange rate. Then we can simply perform such preprocess before running algorithm:

$$x_1 x_2 x_3 \dots \dots x_k > 1$$

$$\left(\frac{1}{x_1}\right)\left(\frac{1}{x_2}\right)\left(\frac{1}{x_3}\right)\dots\dots\left(\frac{1}{x_k}\right) < 1$$

$$ln\ (\left(\frac{1}{x_1}\right)\left(\frac{1}{x_2}\right)\left(\frac{1}{x_3}\right)\dots\dots\left(\frac{1}{x_k}\right)) < \ln(1)$$

$$ln\left(\frac{1}{x_1}\right) + ln\left(\frac{1}{x_2}\right) + ln\left(\frac{1}{x_3}\right)\dots\dots + ln\left(\frac{1}{x_k}\right) < 0$$

In the graph, the weight of the edge is $ln\left(\frac{1}{x}\right)$, and we need to check If there exist a negative cycle in this graph. Clearly, we run Bellman-Ford Algorithm:

b. code:

```
Input: Graph G (V,E), P(u) = INFINITY

Output: True or False

For I in range(V):

        For each edge (u,v) in E:

                If P(u)+w(u,v)<P(v):

                        P(v) = P(u)+w(u,v)


For each edge in e:

        If P(u)+w(u,v)<P(v):

                Return True

Return False
```

C: proof: we are using DP idea here, loop over all the edges V times so that every time we consider the previous computed minimum weight so that we can get the shortest path of all the node starting from source

d. time: O(VE)

2)

The idea, proof, time is same as 1) except we memorize the shortest path here

code:

```
Input: Graph G (V,E), P(u) = INFINITY, Path(S)=0, Path(u) is NAN

Output: True or False

For I in range(V):

        For each edge (u,v) in E:

                If P(u)+w(u,v)<P(v):

                        P(v) = P(u)+w(u,v)

                        Path(v)=u

For each edge in e:

        If P(u)+w(u,v)<P(v):

                Return Path(v)
```

Time O(VE)