```
import numpy as np
import cvxpy as cp

# Exchange rate data.
tickers = ["USD", "EUR", "GBP", "CAD", "JPY", "CNY", "RUB", "MXN", "INR", "BRL"]
n = len(tickers)
F = np.zeros((n, n))
# USD
data = ([1.0, 0.87, 0.76, 1.31, 108.90, 6.72, 65.45, 19.11, 71.13, 3.69],
# EUR
[1.0, 0.88, 1.51, 125.15, 7.72, 75.23, 21.96, 81.85, 4.24],
# GBP
[1.0, 1.72, 142.94, 8.82, 85.90, 25.08, 93.50, 4.84],
# CAD
[1.0, 82.93, 5.11, 49.82, 14.54, 54.23, 2.81],
# JPY
[1.0, 0.062, 0.60, 0.18, 0.65, 0.034],
# CNY
[1.0, 9.74, 2.85, 10.61, 0.55],
# RUB
[1.0, 0.29, 1.09, 0.056],
# MXN
[1.0, 3.73, 0.19],
# INR
[1.0, 0.052],
# BRL
[1.0])
for i in range(n):
    F[i,i:] = data[i]
for j in range(n):
    for i in range(j+1,n):
        F[i,j] = 1.035/F[j,i]

# Initial and final portfolios.
c_req = np.arange(1,n+1)
c_req = 1e4*c_req/c_req.sum()
c_init = c_req[::-1]

# Defining our optimization variable
x = cp.Variable((n,n))

#Define a matrix of all ones
one_m = np.ones(n)

#Defining the Post Exchange holding of currency
c_final = c_init - x.T@one_m + (x/F)@one_m

#Defining the cost function that needs to minimized as per the problem
cost = (c_init-c_final)@(np.sqrt(F[:,0]/F[0,:]))

#Defining the Constraints in the problem
Constraints = [x >= 0, cp.diag(x) == 0, c_final>=c_req, x.T@one_m <= c_init]

#Defining the Objective function to minimze the cost
```

```
#Defining the objective function to minimize the cost
Objective = cp.Minimize(cost)

prob = cp.Problem(Objective, Constraints)

final_cost = prob.solve()

print("The optimal cost of Exchange is",final_cost,"USD")
```

```
The optimal cost of Exchange is 7.720059340057959 USD
```