(1) Basic requirement :

```python
def cholesky_decomposition(cov_matrix):...

def construct_cov_matrix(n, sigma_lst, rho_dict):...

'''
Rainbow Option
parameters of call_put:
call on max: max(max(S1, S2, ..., Sn) - K, 0)
call on min: max(min(S1, S2, ..., Sn) - K, 0)
put on max: max(K - max(S1, S2, ..., Sn), 0)
put on min: max(K - min(S1, S2, ..., Sn), 0)
'''
def rainbow_MC(asset_amount, S0_lst, K, T, r, q_lst, sigma_lst,
rho_dict, sims, reps, call_put):...




# main
asset_amount = 2
S0_lst = [95, 95]
K = 100
T = 0.5
r = 0.01
q_lst = [0.05, 0.05]
sigma_lst = [0.5, 0.5]
rho_dict = {'rho12': 0.3}
sims = 10000
reps = 20

rainbow_MC(asset_amount, S0_lst, K, T, r, q_lst, sigma_lst, rho_dict,
sims, reps, 'call on max')
rainbow_MC(asset_amount, S0_lst, K, T, r, q_lst, sigma_lst, rho_dict,
sims, reps, 'call on min')
rainbow_MC(asset_amount, S0_lst, K, T, r, q_lst, sigma_lst, rho_dict,
sims, reps, 'put on max')
rainbow_MC(asset_amount, S0_lst, K, T, r, q_lst, sigma_lst, rho_dict,
sims, reps, 'put on min')
```

直接調整 **#main** 下面的變數，再直接執行即可。

輸出看起來會是這樣 ：

```
==================================================
Rainbow Option : European Call On Max
==================================================
[ Asset amount = 2 ]
--------------------------------------------------
Mean : 17.691301
Standard Error : 0.269224
95% C.I. : [17.152853, 18.229748]
```

(2) Bonus1 : **AntitheticVariate+MomentMatching.py**

同上，直接調整 **#main** 下面的變數，再直接執行即可。最後，如果模擬數量 **sims** 若輸入基數，我的程式碼會對其進行例外處理（我設計的程式碼無法處理 **sims** 為基數的情況）。