Environment Definition

Environment Description:

```
quotation_matrix = readtable('Quotation Matrix.xlsx', 'ReadRowNames', true)
```

quotation_matrix = 7×7 table

	USD	EUR	JPY	GBP	CHF	CAD	AUD
1 AUD	1.2440	1.7441	0.0115	2.1829	1.0959	1.1634	NaN
2 CAD	1.0693	1.4992	0.0099	1.8763	0.9420	NaN	0.8584
3 CHF	1.1352	1.5915	0.0105	1.9920	NaN	1.0602	0.9111
4 GBP	0.5699	0.7990	0.0053	NaN	0.5013	0.5325	0.4575
5 JPY	107.8600	151.2200	NaN	188.5473	95.1715	100.9495	86.8523
6 EUR	0.7133	NaN	0.0066	1.2506	0.6275	0.6670	0.5732
7 USD	NaN	1.4001	0.0093	1.7533	0.8802	0.9338	0.8034

1.Triangular Arbitrage

Problem Description: Given a 7x7 bid quotation matrix, find all possible arbitrage opportunities and calculate their profits explicitly. Finally sum their profits with respect to currencies specified by user input.

Inputs:

Set the depth limit. Default case is 2.

```
limit_roundtrip =3
```

```
limit_roundtrip = 3
```

Set the currencies on which you want to sum the profits:

```
curr_buckets = {'USD', 'AUD', 'JPY', 'EUR'};
```

Name the xl file on which the solutions will be saved:

```
xl_filename = 'Arbitrage search.xlsx'
```

```
xl_filename =
'Arbitrage search.xlsx'
```

Set name for the transaction logging txt file:

```
log_file_roundtrip = 'log roundtrip search.txt'
```

```
log_file_roundtrip =
'log roundtrip search.txt'
```

Perform search using depth-limited-search algorithm that searchs until exhaustion.

We check the profitability of all round trip trades given a starting currency and them we repeat for the rest of the currencies.

```
currencies = quotation_matrix.Properties.VariableNames;
% Import function from utility .m file.
extract_profit_all = utils('extract_profit_all');

profits_roundtrip = {};
for i = 1: size(currencies, 2)
    % Initialize the problem
    problem_roundtrip = RoundTripTrade(currencies{1, i}, quotation_matrix);

    % Perform the search.
    solutions = depth_limited_search_all(problem_roundtrip, limit_roundtrip);

    % Covert the solution nodes to cell array and store them.
    profits_roundtrip = cat(1, profits_roundtrip, extract_profit_all(solutions, problem
end
```

Discard inefficient solutions:

```
profits_depth_filter = utils('profits_depth_filter');

profits_filtered_roundtrip = profits_depth_filter(profits_roundtrip)
```

profits_filtered_roundtrip = 19×3 cell

	1	2	3
1	1×3 cell	0.0012	1×3 cell
2	1×3 cell	2.9101e-04	1×3 cell
3	1×3 cell	0.0012	1×3 cell
4	1×3 cell	0.0023	1×3 cell
5	1×3 cell	2.1304e-04	1×3 cell
6	1×3 cell	1.2264e-04	1×3 cell
7	1×3 cell	0.0023	1×3 cell
8	1×3 cell	3.3525e-04	1×3 cell
9	1×3 cell	0.0048	1×3 cell
10	1×3 cell	0.0039	1×3 cell
11	1×3 cell	0.0048	1×3 cell
12	1×4 cell	0.0012	1×4 cell
13	1×4 cell	0.0013	1×4 cell
14	1×4 cell	0.0013	1×4 cell
15	1×4 cell	3.6525e-04	1×4 cell
16	1×4 cell	0.0012	1×4 cell

	1	2	3
17	1×4 cell	0.0023	1×4 cell
18	1×4 cell	2.2729e-04	1×4 cell
19	1×4 cell	0.0049	1×4 cell

Pretty print triangular arbitrage solutions:

The quotes in the printed (and logged) message refer to the the bid quotes.

```
pretty_print_grouped_profits = utils('pretty_print_grouped_profits');

output_roundtrip = evalc('pretty_print_grouped_profits(profits_filtered_roundtrip, quot
```

```
output_roundtrip =
    '1. Triangle: USD-->AUD-->JPY-->USD
         1.1 profit=1USD*1.244AUD/USD*86.8523JPY/AUD*0.0092666USD/JPY=0.0012077USD
         1.2 profit=1JPY*0.0092666USD/JPY*1.244AUD/USD*86.8523JPY/AUD=0.0012077JPY
         1.3 profit=1AUD*86.8523JPY/AUD*0.0092666USD/JPY*1.244AUD/USD=0.0012077AUD
     2. Triangle: USD-->CAD-->JPY-->USD
         2.1 profit=1USD*1.0693CAD/USD*100.9495JPY/CAD*0.0092666USD/JPY=0.00029101USD
         2.2 profit=1JPY*0.0092666USD/JPY*1.0693CAD/USD*100.9495JPY/CAD=0.00029101JPY
         2.3 profit=1CAD*100.9495JPY/CAD*0.0092666USD/JPY*1.0693CAD/USD=0.00029101CAD
     3. Triangle: USD-->CHF-->JPY-->USD
         3.1 profit=1USD*1.1352CHF/USD*95.1715JPY/CHF*0.0092666USD/JPY=0.0011558USD
         3.2 profit=1JPY*0.0092666USD/JPY*1.1352CHF/USD*95.1715JPY/CHF=0.0011558JPY
         3.3 profit=1CHF*95.1715JPY/CHF*0.0092666USD/JPY*1.1352CHF/USD=0.0011558CHF
     4. Triangle: USD-->JPY-->GBP-->USD
         4.1 profit=1USD*107.86JPY/USD*0.0053GBP/JPY*1.7533USD/GBP=0.0022829USD
         4.2 profit=1JPY*0.0053GBP/JPY*1.7533USD/GBP*107.86JPY/USD=0.0022829JPY
         4.3 profit=1GBP*1.7533USD/GBP*107.86JPY/USD*0.0053GBP/JPY=0.0022829GBP
     5. Triangle: EUR-->AUD-->JPY-->EUR
         5.1 profit=1EUR*1.7441AUD/EUR*86.8523JPY/AUD*0.006603EUR/JPY=0.00021304EUR
         5.2 profit=1JPY*0.006603EUR/JPY*1.7441AUD/EUR*86.8523JPY/AUD=0.00021304JPY
         5.3 profit=1AUD*86.8523JPY/AUD*0.006603EUR/JPY*1.7441AUD/EUR=0.00021304AUD
     6. Triangle: EUR-->CHF-->JPY-->EUR
         6.1 profit=1EUR*1.5915CHF/EUR*95.1715JPY/CHF*0.006603EUR/JPY=0.00012264EUR
         6.2 profit=1JPY*0.006603EUR/JPY*1.5915CHF/EUR*95.1715JPY/CHF=0.00012264JPY
         6.3 profit=1CHF*95.1715JPY/CHF*0.006603EUR/JPY*1.5915CHF/EUR=0.00012264CHF
     7. Triangle: EUR-->JPY-->GBP-->EUR
         7.1 profit=1EUR*151.22JPY/EUR*0.0053GBP/JPY*1.2506EUR/GBP=0.0022845EUR
         7.2 profit=1JPY*0.0053GBP/JPY*1.2506EUR/GBP*151.22JPY/EUR=0.0022845JPY
         7.3 profit=1GBP*1.2506EUR/GBP*151.22JPY/EUR*0.0053GBP/JPY=0.0022845GBP
     8. Triangle: JPY-->CAD-->AUD-->JPY
         8.1 profit=1JPY*0.0099CAD/JPY*1.1634AUD/CAD*86.8523JPY/AUD=0.00033525JPY
         8.2 profit=1CAD*1.1634AUD/CAD*86.8523JPY/AUD*0.0099CAD/JPY=0.00033525CAD
         8.3 profit=1AUD*86.8523JPY/AUD*0.0099CAD/JPY*1.1634AUD/CAD=0.00033525AUD
     9. Triangle: JPY-->GBP-->AUD-->JPY
         9.1 profit=1JPY*0.0053GBP/JPY*2.1829AUD/GBP*86.8523JPY/AUD=0.0048264JPY
         9.2 profit=1GBP*2.1829AUD/GBP*86.8523JPY/AUD*0.0053GBP/JPY=0.0048264GBP
         9.3 profit=1AUD*86.8523JPY/AUD*0.0053GBP/JPY*2.1829AUD/GBP=0.0048264AUD
    10. Triangle: JPY-->GBP-->CAD-->JPY
         10.1 profit=1JPY*0.0053GBP/JPY*1.8763CAD/GBP*100.9495JPY/CAD=0.0038815JPY
         10.2 profit=1GBP*1.8763CAD/GBP*100.9495JPY/CAD*0.0053GBP/JPY=0.0038815GBP
         10.3 profit=1CAD*100.9495JPY/CAD*0.0053GBP/JPY*1.8763CAD/GBP=0.0038815CAD
    11. Triangle: JPY-->GBP-->CHF-->JPY
         11.1 profit=1JPY*0.0053GBP/JPY*1.992CHF/GBP*95.1715JPY/CHF=0.0047824JPY
         11.2 profit=1GBP*1.992CHF/GBP*95.1715JPY/CHF*0.0053GBP/JPY=0.0047824GBP
         11.3 profit=1CHF*95.1715JPY/CHF*0.0053GBP/JPY*1.992CHF/GBP=0.0047824CHF
    12. Triangle: USD-->CAD-->AUD-->JPY-->USD
         12.1 profit=1USD*1.0693CAD/USD*1.1634AUD/CAD*86.8523JPY/AUD*0.0092666USD/JPY=0.0012267USD
```

```
    12.2 profit=1JPY*0.0092666USD/JPY*1.0693CAD/USD*1.1634AUD/CAD*86.8523JPY/AUD=0.0012267JPY
    12.3 profit=1CAD*1.1634AUD/CAD*86.8523JPY/AUD*0.0092666USD/JPY*1.0693CAD/USD=0.0012267CAD
    12.4 profit=1AUD*86.8523JPY/AUD*0.0092666USD/JPY*1.0693CAD/USD*1.1634AUD/CAD=0.0012267AUD
13. Triangle: USD-->CHF-->AUD-->JPY-->USD
    13.1 profit=1USD*1.1352CHF/USD*1.0959AUD/CHF*86.8523JPY/AUD*0.0092666USD/JPY=0.0012606USD
    13.2 profit=1JPY*0.0092666USD/JPY*1.1352CHF/USD*1.0959AUD/CHF*86.8523JPY/AUD=0.0012606JPY
    13.3 profit=1CHF*1.0959AUD/CHF*86.8523JPY/AUD*0.0092666USD/JPY*1.1352CHF/USD=0.0012606CHF
    13.4 profit=1AUD*86.8523JPY/AUD*0.0092666USD/JPY*1.1352CHF/USD*1.0959AUD/CHF=0.0012606AUD
14. Triangle: USD-->EUR-->AUD-->JPY-->USD
    14.1 profit=1USD*0.7133EUR/USD*1.7441AUD/EUR*86.8523JPY/AUD*0.0092666USD/JPY=0.0012613USD
    14.2 profit=1EUR*1.7441AUD/EUR*86.8523JPY/AUD*0.0092666USD/JPY*0.7133EUR/USD=0.0012613EUR
    14.3 profit=1JPY*0.0092666USD/JPY*0.7133EUR/USD*1.7441AUD/EUR*86.8523JPY/AUD=0.0012613JPY
    14.4 profit=1AUD*86.8523JPY/AUD*0.0092666USD/JPY*0.7133EUR/USD*1.7441AUD/EUR=0.0012613AUD
15. Triangle: USD-->EUR-->CAD-->JPY-->USD
    15.1 profit=1USD*0.7133EUR/USD*1.4992CAD/EUR*100.9495JPY/CAD*0.0092666USD/JPY=0.00036525USD
    15.2 profit=1EUR*1.4992CAD/EUR*100.9495JPY/CAD*0.0092666USD/JPY*0.7133EUR/USD=0.00036525EUR
    15.3 profit=1JPY*0.0092666USD/JPY*0.7133EUR/USD*1.4992CAD/EUR*100.9495JPY/CAD=0.00036525JPY
    15.4 profit=1CAD*100.9495JPY/CAD*0.0092666USD/JPY*0.7133EUR/USD*1.4992CAD/EUR=0.00036525CAD
16. Triangle: USD-->EUR-->CHF-->JPY-->USD
    16.1 profit=1USD*0.7133EUR/USD*1.5915CHF/EUR*95.1715JPY/CHF*0.0092666USD/JPY=0.0011708USD
    16.2 profit=1EUR*1.5915CHF/EUR*95.1715JPY/CHF*0.0092666USD/JPY*0.7133EUR/USD=0.0011708EUR
    16.3 profit=1JPY*0.0092666USD/JPY*0.7133EUR/USD*1.5915CHF/EUR*95.1715JPY/CHF=0.0011708JPY
    16.4 profit=1CHF*95.1715JPY/CHF*0.0092666USD/JPY*0.7133EUR/USD*1.5915CHF/EUR=0.0011708CHF
17. Triangle: USD-->EUR-->JPY-->GBP-->USD
    17.1 profit=1USD*0.7133EUR/USD*151.22JPY/EUR*0.0053GBP/JPY*1.7533USD/GBP=0.0023315USD
    17.2 profit=1EUR*151.22JPY/EUR*0.0053GBP/JPY*1.7533USD/GBP*0.7133EUR/USD=0.0023315EUR
    17.3 profit=1JPY*0.0053GBP/JPY*1.7533USD/GBP*0.7133EUR/USD*151.22JPY/EUR=0.0023315JPY
    17.4 profit=1GBP*1.7533USD/GBP*0.7133EUR/USD*151.22JPY/EUR*0.0053GBP/JPY=0.0023315GBP
18. Triangle: EUR-->CHF-->AUD-->JPY-->EUR
    18.1 profit=1EUR*1.5915CHF/EUR*1.0959AUD/CHF*86.8523JPY/AUD*0.006603EUR/JPY=0.00022729EUR
    18.2 profit=1JPY*0.006603EUR/JPY*1.5915CHF/EUR*1.0959AUD/CHF*86.8523JPY/AUD=0.00022729JPY
    18.3 profit=1CHF*1.0959AUD/CHF*86.8523JPY/AUD*0.006603EUR/JPY*1.5915CHF/EUR=0.00022729CHF
    18.4 profit=1AUD*86.8523JPY/AUD*0.006603EUR/JPY*1.5915CHF/EUR*1.0959AUD/CHF=0.00022729AUD
19. Triangle: JPY-->GBP-->CHF-->AUD-->JPY
    19.1 profit=1JPY*0.0053GBP/JPY*1.992CHF/GBP*1.0959AUD/CHF*86.8523JPY/AUD=0.0048875JPY
    19.2 profit=1GBP*1.992CHF/GBP*1.0959AUD/CHF*86.8523JPY/AUD*0.0053GBP/JPY=0.0048875GBP
    19.3 profit=1CHF*1.0959AUD/CHF*86.8523JPY/AUD*0.0053GBP/JPY*1.992CHF/GBP=0.0048875CHF
    19.4 profit=1AUD*86.8523JPY/AUD*0.0053GBP/JPY*1.992CHF/GBP*1.0959AUD/CHF=0.0048875AUD
'
```

```
% Write output
log_file = fopen(log_file_roundtrip, 'w+');
fwrite(log_file, output_roundtrip);
fclose(log_file);
```

Sum the generated profits to currency "buckets":

```
profits_grouped2curr_buckets = utils('profits_grouped2curr_buckets');

buckets = profits_grouped2curr_buckets(profits_filtered_roundtrip, curr_buckets)
```

```
buckets = struct with fields:
    USD: 0.0126
    AUD: 0.0105
    JPY: 0.0111
```

Save the outputs to excel file.

First we create a sheet on which we save the environment data. That is the quotation matrix.

Then we save the arbitrage opportunities with their respective profits.

Finally we save the cumulative profits of the currencies that the user chose.

```
writetable(quotation_matrix, xl_filename, 'Sheet', 'Input Quotes', "WriteRowNames", tru
write_profits(profits_filtered_roundtrip, xl_filename, 'Triangular arbitrage');
writetable(struct2table(buckets), xl_filename, 'Sheet', 'Triangular Cum Profits');
```

2. Oneway Arbitrage

Problem Description: Given a 7x7 bid quotation matrix, a starting and an ending currency, find all possible one-way arbitrade opportunitites and record their profits explicitly and cumilatively.

Inputs:

Set the depth limit. Default case is 2.

```
limit_oneway =2
```

```
limit_oneway = 2
```

Set starting and ending currencies:

```
starting_curr = 'EUR'
```

```
starting_curr =
'EUR'
```

```
ending_curr = 'JPY'
```

```
ending_curr =
'JPY'
```

Set name for the transaction logging txt file:

```
log_file_oneway = 'log oneway search.txt'
```

```
log_file_oneway =
'log oneway search.txt'
```

Perform search using depth-limited-search algorithm that searchs until exhaustion.

```
% Initialize oneway arbitrage problem instance.
problem_oneway = OneWayTripTrade(starting_curr, ending_curr, quotation_matrix);

% Perform the search.
solutions = depth_limited_search_all(problem_oneway, limit_oneway);

profits_oneway = extract_profit_all(solutions, problem_oneway);
```

Discard inefficient solutions:

```
filter_oneway_profits = utils('filter_oneway_profits');
```

```
profits_filtered_oneway = filter_oneway_profits(profits_oneway)
```

profits_filtered_oneway = 10×2 cell

	1	2
1	1×2 cell	0.2591
2	1×2 cell	0.1235
3	1×2 cell	0.2454
4	1×3 cell	0.2651
5	1×3 cell	0.2613
6	1×3 cell	0.2623
7	1×3 cell	0.1199
8	1×3 cell	0.2557
9	1×3 cell	0.0543
10	1×3 cell	0.0465

Pretty print oneway arbitrage opportunities

The quotes in the printed (and logged) message refer to the the bid quotes.

```
pretty_print_profits_oneway = utils('pretty_print_profits_oneway');
output_oneway = evalc('pretty_print_profits_oneway(profits_filtered_oneway, ending_curr
```

```
output_oneway =
    '1. Triangle: EUR-->AUD-->JPY
         profit=1EUR*1.7441AUD/EUR*86.8523JPY/AUD-151.22JPY/EUR=0.25909JPY
     2. Triangle: EUR-->CAD-->JPY
         profit=1EUR*1.4992CAD/EUR*100.9495JPY/CAD-151.22JPY/EUR=0.12354JPY
     3. Triangle: EUR-->CHF-->JPY
         profit=1EUR*1.5915CHF/EUR*95.1715JPY/CHF-151.22JPY/EUR=0.2454JPY
     4. Triangle: EUR-->CAD-->AUD-->JPY
         profit=1EUR*1.4992CAD/EUR*1.1634AUD/CAD*86.8523JPY/AUD-151.22JPY/EUR=0.26511JPY
     5. Triangle: EUR-->CHF-->AUD-->JPY
         profit=1EUR*1.5915CHF/EUR*1.0959AUD/CHF*86.8523JPY/AUD-151.22JPY/EUR=0.26125JPY
     6. Triangle: EUR-->GBP-->AUD-->JPY
         profit=1EUR*0.799GBP/EUR*2.1829AUD/GBP*86.8523JPY/AUD-151.22JPY/EUR=0.26232JPY
     7. Triangle: EUR-->GBP-->CAD-->JPY
         profit=1EUR*0.799GBP/EUR*1.8763CAD/GBP*100.9495JPY/CAD-151.22JPY/EUR=0.11987JPY
     8. Triangle: EUR-->GBP-->CHF-->JPY
         profit=1EUR*0.799GBP/EUR*1.992CHF/GBP*95.1715JPY/CHF-151.22JPY/EUR=0.25568JPY
     9. Triangle: EUR-->USD-->AUD-->JPY
         profit=1EUR*1.4001USD/EUR*1.244AUD/USD*86.8523JPY/AUD-151.22JPY/EUR=0.054337JPY
     10. Triangle: EUR-->USD-->CHF-->JPY
         profit=1EUR*1.4001USD/EUR*1.1352CHF/USD*95.1715JPY/CHF-151.22JPY/EUR=0.046495JPY
    '
```

```
log_file = fopen(log_file_oneway, 'w+');
fwrite(log_file, output_oneway);
fclose(log_file);
```

Sum the generated profits:

```
cum_profit.(ending_curr) = sum([profits_filtered_oneway{:, 2}])
```

```
cum_profit = struct with fields:
    JPY: 1.8931
```

Save the outputs to excel file.

First we create a sheet on which we save the environment data. That is the quotation matrix.

Then we save the arbitrage opportunities with their respective profits and the summation to the specified currencies.

```
write_profits(profits_filtered_oneway, xl_filename, 'OneWay arbitrage');
writetable(struct2table(cum_profit), xl_filename, 'Sheet', 'OneWay Cum Profits');
```