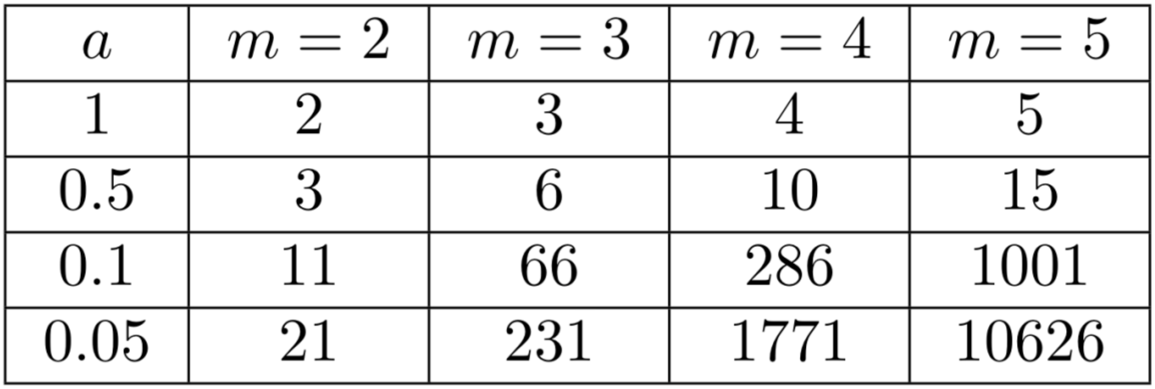
FAQ:

Q: My API returns ‘nan’ values for every query which includes weekends. What shall I do? Shall I raise an exception each time I see an ‘nan’?
A: As shown in the example in the brief, weekend ‘nan’ shouldn’t logically raise an exception. If you do that, you’ll raise 5 exceptions and fail. During tests the required time will exceed a full week. If your API isn’t smart enough to filter out weekends, do it on your own.

Q: Some of you have problems figuring out the set Ω.
A: The table below will give you a hint if you’re in the right direction. It lists the number of elements in the set for m stocks and value of a. Note that in the table, a is defined as the reciprocal of the value of the a defined in the original brief.



a	$m = 2$	$m = 3$	$m = 4$	$m = 5$
1	2	3	4	5
0.5	3	6	10	15
0.1	11	66	286	1001
0.05	21	231	1771	10626

Q: I get a deviation which grows in time from supplied tests, what can I do?
A: There are multiple possibilities for the causes of this, some of which are beyond your control. To allow minimal variation the supplied test files allow a mismatch of 0.01% in each value. During actual grading:

- this threshold will also be used for comparison of the stock prices data
- It will also be used for portfolios, for time periods shorter than 4 months
- For longer tests (up to two years) a mismatch of 0.1% in portfolios will be allowed.
- You can easily adjust the threshold in the supplied test files, for your tests.

Q: I wish to improve the accuracy of my results, what can I do?
A: There could be multiple contributing factors to low accuracy, the easiest to check are:

- What data type are you using for numpy arrays? This can be seen in the ‘dtype’ property of your numpy array (in the debugger). I use ‘float64’
- When returning values, how are you converting your numpy arrays to lists? I use the ‘to_list’ method.
- More details here: https://numpy.org/doc/stable/user/basics.types.html

Q: How will you grade my code?
A: We will run the following tests:

1. First test will have the exact same structure as the check file uploaded to moodle, in terms of:
    a. Length of time period
    b. Amount of stocks
    c. Learn rate, and quantization
2. A test over 4 stocks, over a time period of approximately 2 years. Quantization level taken as $a = 20$. Only universal portfolio computed.
3. A test over 4 stocks, over a time period of less than 3 months. Quantization level taken as $a = 20$. Both portfolios computed.
4. A test over 20 stocks, over a time period of approximately 2 years. Learn rate taken s taken as $\eta = 0.7$. Only exponential gradient considered.

Q: How long can execution take?
A: We don't grade you by execution time even though it is important in real life. For the actual tests described above my code takes about 35 seconds. You will have up to 10 minutes to complete the whole thing. After 10 minutes the grading system will kill your code. You don't want this happening.