Code B: Hackathon

Overview

In this competition a set of hedge funds will be duking it out by trading equity stocks on our simulated exchange in order to see who can end up with the highest net worth at the end of the competition.

By now you should have assembled in teams of up to 3 people and each team should have a login and password.
Your team will act as both the Traders and R&D department of your hedge fund. You will have the option of buying and selling shares of the stock in any way you see fit, whether it is creating a high frequency AI platform, or creating a user interface you can use to track prices and enter trades manually. Feel free to be creative, we will give out prizes to both the team which has the highest net worth, as well as the team that builds the most interesting, insightful and novel user interface to track how well you are doing, as well as the environment of the exchange. Get creative!

The Exchange

There will be 10 stocks with a set of parameters following parameters. All of the parameters will be updated approximately every 1 second. We will call this a period. Any dividend payouts will happen at this time as well. Even though the interval will be the same for all stocks, the stocks will not all be updated at the exact same time; the updates will be spaced out evenly within each period.

The parameters of each stock will be as follows:

1. Earnings(E) - the earnings of the company as a dollar value as a double.
2. Number of shares (N)- the total number of shares of this stock traded on the exchange as a positive integer less than 1 billion
3. Dividend Ratio (D) - percentage of earnings paid as dividends as a double between 0 and 1. One share will receive ED/N each period. This will be a dollar amount as a double.
4. Initial Price (I)- the cost to buy one share before anybody else has, a dollar amount as a double.
5. Market Value per Share (S) - this will be recalculated every time a trade is made. If the trade price is p and number of shares traded is n, it will be recalculated as (n/N)Ap + ((N-n)/N)AS + (1-A)S
6. Volatility (V) - a double between 0 and 1. Each period, a volatility v between -V and V will be chosen. The earnings will be recalculated as E(1/(1+v)). The distribution we pick v from may or may not be uniform, and it may or may not be the same for different stocks. It will however always be the same for the given stock. Figuring out the distribution will give you an edge. We will not tell you the types of distributions we pick from.
7. Ticker Symbol (K) - a text string of length 1-10 representing the stock

In order to buy and sell these stocks, you will have a cash balance in dollars. Should it ever go negative, you will only be able to sell shares of stock until you are in the green again. It is always a good idea to keep sizable cash balance.
Your total net worth is equal to the sum of all Market Value per Share as well as cash. This will be what is used at the end of the competition. You will not be able to know exact market values per

share, but it should be possible to get estimates from inferring what the stock is trading at.

Operations

The first thing you need to do in order to start trading away is open a TCP connection to machine MM on port PP. Enter your username and password separated by a space and ending in a newline and you should be able to perform the following operations continuously as needed.

1. MY_CASH
   Will output your current cash.
   Output Format: MY_CASH_OUT <cash>
2. MY_SECURITIES
   Will output all shares you own.
   Output Format: MY_SECURITIES_OUT <ticker> <shares> <dividend_ratio> <ticker> <shares> <dividend_ratio> …
3. MY_ORDERS
   Will output all of your current orders bid & ask on the exchange.
   Output Format: MY_ORDERS_OUT [<BID or ASK> <price> <shares>…]
4. SECURITIES
   Will output all securities in the exchange.
   Output Format: SECURITIES_OUT <ticker> <net_worth> <dividend_ratio> <volatility>…
5. ORDERS <ticker>
   Will list all orders on the exchange for a ticker.
   Output Format: SECURITY_ORDERS_OUT [<BID or ASK> <price> <shares>…]
6. BID <ticker> <price> <shares>
   Place a new bid.
   Output Format: BID_OUT DONE or ERROR Not Enough Cash to make bid order
7. ASK <ticker> <price> <shares>
   Place a new ask.
   Output Format: ASK_OUT DONE or ERROR Not Enough Shares Owned
8. CLEAR_BID <ticker>
   Clear your bid.
   Output Format: CLEAR_BID_OUT DONE or ERROR No Security Specified
9. CLEAR_ASK <ticker>
   Clear your ask.
   Output Format: CLEAR_ASK_OUT DONE or ERROR No Security Specified
10. SUBSCRIBE
    Subscribe to any trades **you** make as well as any bids. The market may automatically withdraw due to lack of funds.
    Output Format: BUY <ticker> <price> <shares> or SELL <ticker> <price> <shares>
    *This is the only command which will cause writes socket asynchronously.
    **Use this to help create cool UI.
11. UNSUBSRCIBE
    Unsubscribe from updates.
12. CLOSE_CONNECTION
    End the connection gracefully.

Loose Ends

We encourage your team to get creative as possible, as stated we will judge the best strategy separately from the best UI.

That being said, do not do anything against the spirit of the competition. That means don't login as another team, don't look over their shoulder to get information about who is bidding what, and don’t try to sniff packets. If you have any questions about it, ask us first.