Problem Statement

We are developing a rewards bidding system - **BidBlitz** for Flipkart, where Flipkart plus members will have the opportunity to win a lavish item each day.

As an engineer, your task is to build a feature that allows members to place bids using their Flipkart Super Coins.

- The winner of the item will be the member who places the lowest bid. ● At the end of each day, the system should declare the winner. ● Members should be able to view the winners of past events.

This system aims to enhance member engagement and provide an exciting and rewarding experience for Flipkart plus users.

Explanations

1. What is the bid ?
    a. Pledge Super coin to buy the lavish item
2. What is the BidBlitz Event ?
    a. It is event in which members submit the bids and at the end of the event , winner is decided based on some criteria (mentioned in the requirements)

Requirements

1. System should be able to add members and each member will have super coins assigned by system
    a. Number of super coins assigned by system should be greater than zero
2. System should be able to add event where event name should be unique for each event
    a. System can only add one event in a single day
3. Members can register for the event and only registered members can participate in the event.
4. Members should be able to submit bids for a particular event as per the below conditions
    a. Member can only submit all bids at single go and at max 5 bids can be submitted
    b. Member should have atleast max of 5 bids super coins in his wallet
        i. Suppose member submit bids -> 100,500,400,800,900 ii. Then member should have at least 900 super coins available
        iii. Only the max bid would be deducted from the member wallet
            1. In above given example, only 900 super coins will

be deducted from member wallet

c. Each bid should be unique for the member for that event i. Suppose member submit 4 bids -> 100, 200, 300, 400 ii. As each bid has unique value

d. Each bid should be greater than zero

5. System admin will declare the winner.
   a. How is the winner decided?
      i. Member with lowest bid will be declared as winner
      ii. If the lowest bid is not unique then member who submitted lowest bid first will be declared as winner

Bonus Requirement

1. Members can see the winners of past events.
   a. How many past events can be made visible -> 5
   b. Order by ascending or descending
   c. Winners should be sorted by event date

Commands

1. ADD_MEMBER <id> <name> <number_of_super_coins>
   a. Example : ADD_MEMBER 1 akshay 10000
   b. Output : Akshay added successfully
   c. Example : ADD_MEMBER 2 chris 5000
   d. Output : Chris added successfully

2. ADD_EVENT <id> <event_name> <prize_name> <date>
   a. Example : ADD_EVENT 1 BBD IPHONE-14 2023-06-06
   b. Output : BBD with prize IPHONE-14 added successfully

3. REGISTER_MEMBER <member_id> <event_id>
   a. Example : REGISTER_MEMBER 1 1
   b. Output : Akshay registered to the BBD event successfully

4. SUBMIT_BID <member_id> <event_id> <bid_1> <bid_2> <bid_3> <bid_4> <bid_5> a. Example : SUBMIT_BID 1 1 100 200 400 500 600
   b. Output : BIDS submitted successfully
   c. Example SUBMIT_BID 2 1 100 200 400 500
   d. Output : BIDS submitted successfully
   e. Example : SUBMIT_BID 10 1 100 200 300 400 500
   f. Output : Member did not registered for this event

5. DECLARE_WINNER EVENT_ID
   a. Example : DECLARE_WINNER 1
   b. Output : Akshay wins the IPHONE-14 with lowest bid 100

BONUS

6. LIST_WINNERS <order_by>
   a. Example : LIST_WINNERS asc
   b. Output : [{event_id, winner_name, lowest_bid, date}]

Guidelines:

- Input can be read from a file or STDIN or coded in a driver method. [No Api and No UI]
- Output can be written to a file or STDOUT. [No Api]
- Store all interim/output data **in-memory data structures**. The usage of databases is not allowed.
- Restrict internet usage to looking up syntax.
- Language should be Java only.
- Save your code/project by your name and email it or upload on the google drive link provided. Your program will be executed on another machine. So, explicitly specify dependencies, if any, in your email.

Expectations:

- The code should be demo-able (very important). The code should be functionally correct and complete.
  - At the end of this interview round, an interviewer will provide multiple inputs to your program for which it is expected to work
- The code should handle edge cases properly and fail gracefully. Add suitable exception handling, wherever applicable.
  - An example would be to display an error message when the member trying to register for same event again or member does not have enough super coins to bid
  - The code should be readable, modular, testable, and extensible. Use intuitive names for your variables, methods, and classes.
  - It should be easy to add/remove functionality without rewriting a lot of code.
  - Do not write a monolithic code.
- Don’t use any databases.