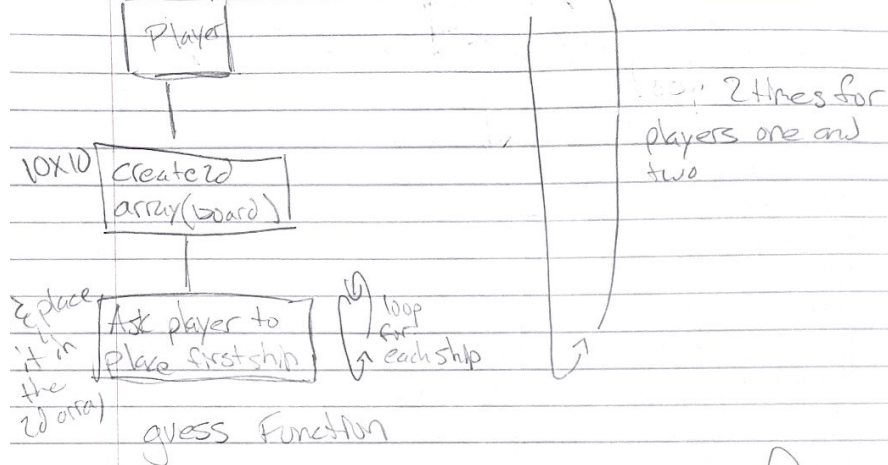1. The user inputs would be where they would want to put the shits. Another input would there turns of guessing where a ship is within the matrix.
2. The programs outputs would be whether or not the player hit or missed a ship after they guessed, the blank board and the updated board after every turn. It should also output the winner at the end of the game.
3. I'm assuming they are guessing within the limits of the 10x10 board and that they know which positions have already been guessed. I'm also assuming that the players are placing their ships within the matrix as well.
4. The main task is the alternate from board to board as the two players miss and hit the others ships. It also should update the output board after each turn. This involves many subtasks of locating the guesses within the opponent's board, determining if a ship is there, and then updating it. Another subtask would be having the players place there ships within the matrix and holding those positions throughout the entire game.

1. The game boards should be 2d dynamic arrays stored within the heap that would be deleted and nulled within the code so there isn't any memory leaks.
2. Each guess should just be a locater variable within the array. If that memory location contains a ship, the function will print a hit. It also will update the board to show that you've already guessed that location.
3. The position of the guess will be singular and shouldn't allow the user to guess more than one unit of the matrix. When the players are deciding where to put there ships, the player will input the front and back position of the boat and it will display what it will look like. Then they will be prompted if they want to keep the ship there or change and loop that until the player has placed all ships. If the positions are outside of the matrix, the user will be asked to pick a different position.
4. The ships shouldn't overlapping, so if the player inputs a ship that is already there they will be asked again.
5. My idea of this game is having two boards for each player, one that is there own board that contains there ships that they choose and a blank matrix that is updated throughout the game based on the users guesses. When it comes to guessing, the blank board would be updated depending on the comparison of the guess and the opponents personal board.
6. You would display the board with a couple for loops that print out each location of the array.
7. The hits and misses can be counted using a simple variable that is added by one every time a user hits or misses. There should be two of each of these variables for each of the players.
8. The program should be reading input from the user every turn for each players guess and when the players choose the positions of their ships at the beginning of the game.
9. The ship placement task is repeated, just with different lengths of ships for each player. The task of a players turn and updating the matrix for each board based on there guess should be repeated but alternate back and forth from each player.

Battleship

Ship placement Function

[Player]

|

10X10 → [Created 2d array(board)]

|

(place it in the 2d array) → [Ask player to place first ship] → loop for each ship

Loop 2 times for players one and two

guess Function

[Player]

|

[ask for coordinat point]

|

[search the opponents array for ship] → display hit or miss

|

[update the blank matrix]

Alternate between Players

Main Function

```
┌─────────────┐
│ Creat 2     │  * One for each player
│ blank 10x10 │
│ 2d arrays   │
└─────────────┘
        ↓
┌──────────────────┬─────────────────────┐
│ prompt and display │ * guess Functions will │
│ the matrix before  │ update these arrays.   │
│ and after each     │                        │
│ players turn       │ * loop till one battleship │
│                    │   is completly hit     │
└──────────────────┴─────────────────────┘
        ↓
┌──────────────────┬─────────────────────┐
│ break and        │                     │
│ display winner   │                     │
└──────────────────┴─────────────────────┘
```
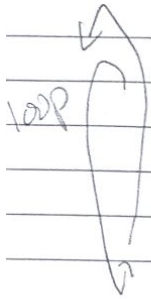
loop

Inputs:

Ship location, good- within the 10x10 matrix, bad- outside of matrix or overlapping other ships, max- 10/j, min- 0

Users guess, good- within the 10x10 matrix, bad- outside the matrix or not a valid guess, max- 10/j, min- 0