Project 2    PULL CODE

1. Display: name, EUIDm email, department name, course number
2. Declare enum constant MenuChoices (1 = displayLeft, 2 = displayRight, 3 = Guess, 4 = Change, 5 =  Exit)
    a. MenuChoices m_choices;
3. write Function: getName- this is for getting the name
    a. cin player's name
    b. only alphabet and white space allowed
        i. if not error message and ask for the name again
        ii. keep asking until correct
    c. convert every initial to uppercase and every other to lowercase
4. Write Function: genShowMatrix- this is the visible array
    a. Set all matrix points to -1
    b. Will be called when the visible array needs to be shown
5. Write Function: genHideMatrix- this is the hidden array
    a. Set all matrix to randomly generated numbers between the bounds that are already generated (numbers may repeat)
    b. will be called when the hidden array is needed
6. Write Function: showMartix- displays corresponding 2D array
7. Write Funcion: setdisplayLeft
    a. If player chooses this display the left/smaller bound instead of -1
        i. Provide message saying: if correct guess they will only earn 1 point and if incorrect guess they will lose 10 points
        ii. Make sure the player can not display both bounds simultaneously
8. Write function: setdisplayRight
    a. if play chooses this display the right/bigger bound instead of -1

        i. Provide message saying: if correct guess they will only earn 1 point and if incorrect guess they will lose 10 points
        ii. Make sure the player can not display both bounds simultaneousl

9. Write function: Eliminate
    a. Called from guess
    b. obtain 2 integer values representing the row index and the column index from the guess function
    c. this sets all values in corresponding row and column to zero in both  visible and hidden arrays
        i. EX. received parameters: row 1 and column 3 the function will set all values in the 2nd row and 4th column equal to 0
10. Write function: allZeros of Boolean return type
    a. will check if all elements in a 2D array is zero
        i. if so return true otherwise return false
11. Write function: guess
    a. Ask user to guess the values in the hidden array
        i. bounds are hidden initially so players donesnt know what numbers are used to create the matrix unless the player chooses to reveal one of the two bounds
    b. check if guess matches any value in hidden array.
        i. if match call function eliminate and pass the corresponding row and column indices where the match occurred
            1. increment points accordingly and provide suitable message
    c. if no match decrement points and provide suitable message
    d. if player displayed a bound increment points by 1 and decrement points by 10 when appropriate

    e. if player has NOT displayed either bound increment by 5 points and decrement by 5 points when appropriate
    f. update player with the point balance after every guess
    g. this function will be called from main function
12. Write function: initialize- sets the starting parameters of the game as well as to restart game
    a. generate lower and upper bounds
        i. generate 2 rand ints one in ranger of 100-199 and the other in range of 200-299
        ii. set displayed lower and upper bounds to -1
        iii. call function genHideMatric to cgenerate hidden array (need to pass the lower and upper bounds)
        iv. call function genShowMatrix to generate displayed array

INSIDE MAIN FUNCTION

13. declare int=100 (represents the points each player starts with
14. declare 2 ints= -1 for upper and lower bounds
    a. incase the player wants to display you will replace the -1
15. call function initialize to set the starting game parameters
16. call function getName to get name of player and display a welcom message using the name

LOOP THIS (17-18)

17. based on enum constant data, generate menu choice for the player. Using an int variable, ask the player to select from the menu
    a. `cout << displayNumL << "     " << displayNumR << endl;`
    b. `cout << "1. Display left number" << endl;`
    c. `cout << "2. Display right number" << endl;`

d. cout << "3. Guess a number in between" << endl;

e. cout << "4. Generate new numbers" << endl;

f. cout << "5. Exit" << endl;

g. cout << "What would you like to do? (1-5)" << endl;

18. Design switch case block with a default case, using enum data
    a. based on players input of step 17 one of the cases will execute
        i. Use a variable of your enum const type as the switching expression
        ii. Left bound
            1. call setdisplayLeft
        iii. Right bound
            1. call setdisplayRight
        iv. Guess
            1. call guess
            2. after returning check if all values in the hidden array have been eliminated by calling allZeros
                a. if yes the user has won and ask if the user wants to play another game
                    i. if user choose to play another game call initialize
        v. Rest Game
            1. call initialize
            2. deduct 1 point and provide player with new point balance
        vi. Exit
            1. display goodbye message using the name of the player and display final point balance
        vii. Default
            1. provide error message and ask player to enter again

FUNCTIONS:

- getName -
- setdisplayLeft -
- setdisplayRight -
- guess
- eliminate
- allZeros
- showMatrix -
- genHideMatrix -
- genShowMatrix-
- initialize -
- main -

code name:  euidProject2.cpp