Contributors:

Name	Contribution
Muhammad Harmain	50%
Usman Ahmed	50%

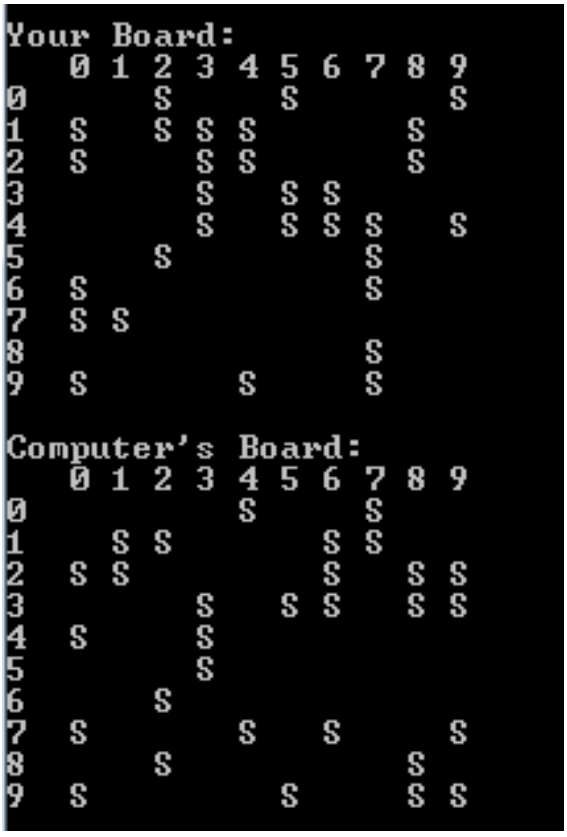
Features covered:

1: Grid-Based Game Board: The game board is displayed as a grid where players will place thier ships and track their attacks. A typical grid size is 10x10.

```c
void initializeBoards() {
    for (i = 0; i < BOARD_SIZE; i++) {
        for (j = 0; j < BOARD_SIZE; j++) {
            userBoard[i][j] = ' ';
            computerBoard[i][j] = ' ';
        }
    }
}

void printBoards() {
    printf("Your Board:\n");
    for (i = 0; i < BOARD_SIZE; i++) {
        if (i == 0) {
            printf("   ");
        } else {
            printf(" ");
        }
        printf("%d", i);
    }
    printf("\n");
    for (i = 0; i < BOARD_SIZE; i++) {
        printf("%d  ", i);
        for (j = 0; j < BOARD_SIZE; j++) {
            printf("%c ", userBoard[i][j]);
        }
        printf("\n");
    }

    printf("\nComputer's Board:\n");
    for (i = 0; i < BOARD_SIZE; i++) {
        if (i == 0) {
            printf("   ");
        } else {
            printf(" ");
        }
        printf("%d", i);
    }
    printf("\n");
    for (i = 0; i < BOARD_SIZE; i++) {
        printf("%d  ", i);
        for (j = 0; j < BOARD_SIZE; j++) {
            printf("%c ", computerBoard[i][j]);
        }
        printf("\n");
    }
}
```



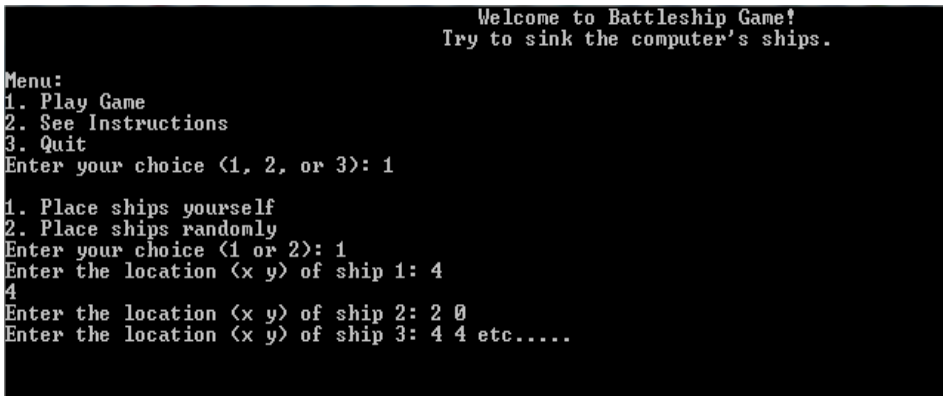
2: Ship Placement: Players can place their ships on the game board manually or randomly.

```c
void placeShips_randomly(char board[BOARD_SIZE][BOARD_SIZE]) {
    int shipSize = SHIP_SIZE;
    while (shipSize > 0) {
        int x = rand() % BOARD_SIZE;
        int y = rand() % BOARD_SIZE;
        if (board[x][y] == ' ') {
            board[x][y] = 'S';
            shipSize--;
        }
    }
}
```

```c
int isValidTarget(int x, int y) {
    return x >= 0 && x < BOARD_SIZE && y >= 0 && y < BOARD_SIZE;
}

void placeShips_manually(char board[BOARD_SIZE][BOARD_SIZE]) {
    int x, y, shipSize = SHIP_SIZE;
    for (i = 0; i < shipSize; i++) {
        printf("Enter the location (x y) of ship %d: ", i + 1);
        scanf("%d %d", &x, &y);

        if (isValidTarget(x, y) && board[x][y] == ' ') {
            board[x][y] = 'S';
        } else {
            printf("Incorrect input! Try again.\n");
            i--;
        }
    }
}
```



3: User Input: Implement a user-friendly interface that allows players to input their attack coordinates using a simple format (e.g., “0 1”, “2 4”).

```
138     while (userShips > 0 && computerShips > 0) {
139         int userX, userY, computerX, computerY;
140
141         printf("\n");
142         printBoards();
143
144         printf("\nYour turn. Enter target coordinates (x y): ");
145         scanf("%d %d", &userX, &userY);
146
147         if (isValidTarget(userX, userY) && computerBoard[userX][userY] == 'S') {
148             hit_user++;
149             printf("\nYou hit a ship!\n");
150             computerBoard[userX][userY] = 'X';
151             computerShips--;
152         }
153         else if (isValidTarget(userX, userY) && computerBoard[userX][userY] != 'X') {
154             miss_user++;
155             printf("\nYou missed.\n");
156             computerBoard[userX][userY] = 'O';
157         }
158         else if (isValidTarget(userX, userY) && computerBoard[userX][userY] == 'X') {
159             miss_user++;
160             printf("\nYou missed.\n");
161         }
162         else if (userX>10||userX<0||userY>10||userY<0) {
163             miss_user++;
164             printf("\nYou missed.\n");
165         }
```

```
Your Board:
   0 1 2 3 4 5 6 7 8 9
0                S
1                    S
2  S S S    S S
3    S S    S S
4      S              S
5  S S S        S     S
6  S      S     S     S
7                S           S
8    S     S     S     S     S
9                S

Computer's Board:
   0 1 2 3 4 5 6 7 8 9
0  S         S     S S
1  S     S       S S
2        S                S
3  S     S S     S
4    S             S
5  S S
6                S
7          S       S S     S
8    S S     S     S S
9                S     S

Your turn. Enter target coordinates (x y): 0 0

You hit a ship!
The computer hit your ship at (3, 2)!
```

4: Hit and Miss Feedback: After each attack, provide clear visual feedback to indicate whether it was a hit or a miss.

note: A part of loop

```
183                 printf("TOTAL USER HITS: %d\n", hit_user);
184                 printf("TOTAL USER MISS: %d\n", miss_user);
185                 printf("TOTAL COMPUTER HITS: %d\n", hit_computer);
186                 printf("TOTAL COMPUTER MISS: %d\n", miss_computer);
187             }
```

```
TOTAL USER HITS: 1
TOTAL USER MISS: 1
TOTAL COMPUTER HITS: 1
TOTAL COMPUTER MISS: 1

Your Board:
   0 1 2 3 4 5 6 7 8 9
0        S   S
1          S         S
2  S S             S S S
3                    S
4              S   S   S
5      S       S       X
6  S S     S
7  S       S S   S
8  S         S     S O S
9           S S S

Computer's Board:
   0 1 2 3 4 5 6 7 8 9
0  O   S S   S S
1  X
2
3    S S S S         S
4    S   S       S S
5            S
6  S       S         S
7           S       S S S
8  S   S         S
9  S S S     S S
```

```
Your turn. Enter target coordinates (x y): 3 1

You hit a ship!
The computer missed at (9, 8)!

TOTAL USER HITS: 2
TOTAL USER MISS: 1
TOTAL COMPUTER HITS: 1
TOTAL COMPUTER MISS: 2

Your Board:
   0 1 2 3 4 5 6 7 8 9
0        S   S
1          S         S
2  S S             S S S
3                    S
4              S   S   S
5      S       S       X
6  S S     S
7  S       S S   S
8  S         S     S O S
9           S S S   O

Computer's Board:
   0 1 2 3 4 5 6 7 8 9
0  O   S S   S S
1  X
2
3    X S S S         S
4    S   S       S S
5            S
6  S       S         S
7           S       S S S
8  S   S         S
9  S S S     S S
```

5: Simple AI Opponent: If you decide to include an AI opponent for single-player mode, keep its behavior straightforward. It should make random.

```
167         do {
168             computerX = rand() % BOARD_SIZE;
169             computerY = rand() % BOARD_SIZE;
170         } while (userBoard[computerX][computerY] == 'X' || userBoard[computerX][computerY] == 'O');
171
172         if (userBoard[computerX][computerY] == 'S') {
173             hit_computer++;
174             printf("The computer hit your ship at (%d, %d)!\n\n", computerX, computerY);
175             userBoard[computerX][computerY] = 'X';
176             userShips--;
177         } else {
178             miss_computer++;
179             printf("The computer missed at (%d, %d)!\n\n", computerX, computerY);
180             userBoard[computerX][computerY] = 'O';
181         }
```

```
Your turn. Enter target coordinates (x y): 3 1

You hit a ship!
The computer missed at (9, 8)!

TOTAL USER HITS: 2
TOTAL USER MISS: 1
TOTAL COMPUTER HITS: 1
TOTAL COMPUTER MISS: 2

Your Board:
   0 1 2 3 4 5 6 7 8 9
0        S   S
1          S         S
2  S S             S S S
3                    S
4              S     S   S
5      S       S         X
6  S S       S
7  S       S S     S
8  S         S       S O S
9           S S S     O

Computer's Board:
   0 1 2 3 4 5 6 7 8 9
0  O   S S   S S
1  X
2
3    X S S S           S
4    S   S         S S
5              S
6  S       S           S
7          S         S S S
8  S   S           S
9  S S S       S S
```

6: Game Ending Screen: When the game ends, display a game-over screen that announces the winner and allows players to start a new game or return to the main menu.

```
189             printf("\nTHE GAME IS OVER\n");
190
191             if (computerShips == 0) {
192                 printf("CONGRATULATIONS! YOU SUNK ALL OF THE COMPUTER'S SHIPS.\nYOU WIN\n");
193             } else if (userShips == 0) {
194               printf("OOPS! THE COMPUTER SUNK ALL OF YOUR SHIPS.\nYOU LOSE\n");
195             }
196
197             printf("1. Play again\n");
198             printf("2. Exit\n");
199             printf("Enter your choice (1 or 2): ");
200             scanf(" %c", &ch_4);
201
202             switch (ch_4) {
203                 case '1':
204                     initializeBoards();
205                     placeShips_randomly(computerBoard);
206                     userShips = SHIP_SIZE;
207                     computerShips = SHIP_SIZE;
208                     break;
209
210                 default:
211                    printf("The game is exiting. Thank you for playing!\n");
212                     return 0;
213             }
214
215             break;
```

```
THE GAME IS OVER
CONGRATULATIONS! YOU SUNK ALL OF THE COMPUTER'S SHIPS.
YOU WIN
1. Play again
2. Exit
Enter your choice (1 or 2): 1
Menu:
1. Play Game
2. See Instructions
3. Quit
Enter your choice (1, 2, or 3): 1

1. Place ships yourself
2. Place ships randomly
Enter your choice (1 or 2):
```

7. Main menu: Displays choices to play the game, to view the game instructions option or to exit the game.

```c
    do {
        int hit_user = 0, hit_computer = 0, miss_user = 0, miss_computer = 0;
        printf("Menu:\n");
        printf("1. Play Game\n");
        printf("2. See Instructions\n");
        printf("3. Quit\n");
        printf("Enter your choice (1, 2, or 3): ");
        scanf(" %c", &ch_1);

        switch (ch_1) {
            case '1':
                back:
                printf("\n1. Place ships yourself\n");
                printf("2. Place ships randomly\n");
                printf("Enter your choice (1 or 2): ");
                scanf(" %c", &ch_2);

                switch (ch_2) {
                    case '1':
                        placeShips_manually(userBoard);
                        break;

                    case '2':
                        placeShips_randomly(userBoard);
                        break;

                    default:
                        printf("Invalid input. Please enter 1 or 2.\n");
                        goto back;
                }
```

```c
            case '2':
                printf("\nINSTRUCTIONS:\n");
                printf("Step 1: Place yor ships on a 10x10 map.\n");
                printf("Step 2: Start entering the row and colomn coordinates for hitting a specific position on computers map\n");
                printf("Note: The player who has most hits under his belt is the winner\n\n");
                printf("1. Play (Press 1)\n");
                printf("2. Quit (Press any key)\n");
                printf("\nEnter your choice: ");
                scanf(" %c", &ch_3);

                switch (ch_3) {
                    case '1':
                        goto back;
                        break;

                    default:
                        printf("The game is exiting. Thank you for playing!\n");
                        return 0;
                }

                break;

            case '3':
                printf("\nThe game is exiting. Thank you for playing!\n");
                return 0;

            default:
                printf("\nInvalid input. Please enter 1, 2, or 3.\n");
        }

    } while (1);
```

```
                                    Try to sink the computer's ships.

Menu:
1. Play Game
2. See Instructions
3. Quit
Enter your choice (1, 2, or 3): 1

1. Place ships yourself
2. Place ships randomly
Enter your choice (1 or 2): 2
```