**Battleship final version updated v3.0**

**#include<iostream>**

**#include<iomanip>**

**#include <fstream>**

**#include <stdlib.h>**

**#include<ctime>**

**#include<windows.h>**

**using namespace std;**

**//arr1-4 is for easy difficulty**

**//arr5-8 is for medium difficulty**

**//arr9-12 is for hard difficulty**

**char arr1[8][8]; //array for storing player 1 ships in easy difficulty**

**char arr2[8][8]; //array for storing player 2 ships in easy difficulty**

**char arr3[8][8]; //array for player 1 to guess where player 2 ship is in easy difficulty**

**char arr4[8][8]; //array for player 2 to guess where player 1 ship is in easy difficulty**

**char arr5[12][12]; //array for storing player 1 ships in medium difficulty**

**char arr6[12][12]; //array for storing player 2 ships in medium difficulty**

**char arr7[12][12]; //array for player 1 to guess where player 2 ship is in medium difficulty**

**char arr8[12][12]; //array for player 1 to guess where player 2 ship is in medium difficulty**

**char arr9[15][15]; //array for storing player 1 ships in hard difficulty**

**char arr10[15][15]; //array for storing player 2 ships in hard difficulty**

**char arr11[15][15]; //array for player 1 to guess where player 2 ship is in hard difficulty**

**char arr12[15][15]; //array for player 1 to guess where player 2 ship is in hard difficulty**

**//grid 1-4 is for easy difficulty**

**//grid 5-8 is for medium difficulty**

**//grid 9-12 is for hard difficulty**

**void grid1(char arr1[8][8]);**

**void grid2(char arr2[8][8]);**

**void grid3(char arr3[8][8]);**

**void grid4(char arr4[8][8]);**

**void grid5(char arr5[12][12]);**

**void grid6(char arr6[12][12]);**

**void grid7(char arr7[12][12]);**

**void grid8(char arr8[12][12]);**

**void grid9(char arr9[15][15]);**

**void grid10(char arr10[15][15]);**

**void grid11(char arr11[15][15]);**

**void grid12(char arr12[15][15]);**

**void mainmenu();**

**void instructions();**

**void credits();**

**void player\_vs\_player\_easy();**

**void player\_vs\_player\_medium();**

**void player\_vs\_player\_hard();**

**void player\_vs\_machine\_Easy();**

**void player\_vs\_Machine\_Medium();**

**void player\_vs\_Macine\_Hard();**

**int main()**

**{**

**char carrier[2] = { 'A','A' };**

**char battleships[3] = { 'B','B','B' };**

**char cruisers[4] = { 'C','C','C','C' };**

**char submarines[4] = { 'D','D','D','D' };**

**char destroyers[2] = { 'E','E' };**

**cout << "\t\t\t\txxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" << endl;**

**cout << "\t\t\t\txxxxxxxxxxxxxxxxxxxxx|---| /\\ --|-- --|-- | |--- --- | | --|-- |---| xxxxxxxxxxxxxxxxxxxxxx" << endl;**

**cout << "\t\t\t\txxxxxxxxxxxxxxxxxxxxx|---| /\_\_\\ | | | |--- |\_\_ |\_\_| | |\_\_\_| xxxxxxxxxxxxxxxxxxxxxx" << endl;**

**cout << "\t\t\t\txxxxxxxxxxxxxxxxxxxxx|---|/ \\ | | |\_ |\_\_\_ \_ | | | \_\_|\_\_ | xxxxxxxxxxxxxxxxxxxxxx" << endl;**

**cout << "\t\t\t\txxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" << endl;**

**cout << endl;**

**cout << endl;**

**cout << endl;**

**cout << endl;**

**cout << endl;**

**//Initializing all array with '\*'**

**for (int i = 0;i < 8;i++)**

**{**

**for (int j = 0;j < 8;j++)**

**{**

**arr1[i][j] = '\*';**

**}**

**}**

**for (int i = 0;i < 8;i++)**

**{**

**for (int j = 0;j < 8;j++)**

**{**

**arr2[i][j] = '\*';**

**}**

**}**

**for (int i = 0;i < 8;i++)**

**{**

**for (int j = 0;j < 8;j++)**

**{**

**arr3[i][j] = '\*';**

**}**

**}**

**for (int i = 0;i < 8;i++)**

**{**

**for (int j = 0;j < 8;j++)**

**{**

**arr4[i][j] = '\*';**

**}**

**}**

**for (int i = 0;i < 12;i++)**

**{**

**for (int j = 0;j < 12;j++)**

**{**

**arr5[i][j] = '\*';**

**}**

**}**

**for (int i = 0;i < 12;i++)**

**{**

**for (int j = 0;j < 12;j++)**

**{**

**arr6[i][j] = '\*';**

**}**

**}**

**for (int i = 0;i < 12;i++)**

**{**

**for (int j = 0;j < 12;j++)**

**{**

**arr7[i][j] = '\*';**

**}**

**}**

**for (int i = 0;i < 12;i++)**

**{**

**for (int j = 0;j < 12;j++)**

**{**

**arr8[i][j] = '\*';**

**}**

**}**

**for (int i = 0;i < 15;i++)**

**{**

**for (int j = 0;j < 15;j++)**

**{**

**arr9[i][j] = '\*';**

**}**

**}**

**for (int i = 0;i < 15;i++)**

**{**

**for (int j = 0;j < 15;j++)**

**{**

**arr10[i][j] = '\*';**

**}**

**}**

**for (int i = 0;i < 15;i++)**

**{**

**for (int j = 0;j < 15;j++)**

**{**

**arr11[i][j] = '\*';**

**}**

**}**

**for (int i = 0;i < 15;i++)**

**{**

**for (int j = 0;j < 15;j++)**

**{**

**arr12[i][j] = '\*';**

**}**

**}**

**mainmenu();**

**return 0;**

**}**

**void instructions()**

**{**

**cout << "Rules:\n\n";**

**cout << "Proper battle grounds will be given to each of the players" << endl;**

**cout << endl;**

**cout << "1.Firstly the control of ships will be given to each player and they will align their ships vertically or horizontally" << endl;**

**cout << "2.Aesteric represents available options on the grid." << endl;**

**cout << "3.Then there will be a toss player to win the toss will get the first turn" << endl;**

**cout << "4.Each players ships will be hidden from the other player" << endl;**

**cout << "5.If player1 guesses the position of player 2's ship 'X' will appear on player 1's input grid to let him know which places have been attacked" << endl;**

**cout << "6.If player 2 will guess the position of player 1's ship 'O' will appear on player 2's input grid." << endl;**

**cout << "7.If the guess is incorrect '.' will replace the aesteric on the grid." << endl;**

**cout << "8.Player to destroy all the ships of the other player will win." << endl;**

**}**

**void credits()**

**{**

**string st1 = " Made By ";//<< endl;**

**string st2 = "Muhammad Sheharyar Ajmal (F21-9304)";// << endl;**

**string st3 = "Abdul Raheem Shehzad (F21-9052) ";// << endl;**

**fstream file; // Storing in a File**

**file.open("credit.txt", ios::out);**

**file << st2 << " ";**

**file << st2 << " ";**

**file << st3 << " ";**

**file.close();**

**cout << st1 << endl;**

**cout << st2 << endl;**

**cout << st3 << endl;**

**}**

**//functions to display grids**

**void grid1(char arr1[8][8])**

**{**

**system("cls");**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7";**

**cout << endl;**

**for (int row = 0;row < 8;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 8;col++)**

**{**

**cout << setw(4) << arr1[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void grid2(char arr2[8][8])**

**{**

**system("cls");**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7";**

**cout << endl;**

**for (int row = 0;row < 8;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 8;col++)**

**{**

**cout << setw(4) << arr2[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void grid3(char arr3[8][8])**

**{**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7";**

**cout << endl;**

**for (int row = 0;row < 8;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 8;col++)**

**{**

**cout << setw(4) << arr3[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void grid4(char arr4[8][8])**

**{**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7";**

**cout << endl;**

**for (int row = 0;row < 8;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 8;col++)**

**{**

**cout << setw(4) << arr4[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void grid5(char arr5[12][12])**

**{**

**system("cls");**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7" << setw(4) << "C 8 " << setw(4) << "C 9 " << setw(4) << "C 10 " << setw(4) << "C 11 ";**

**cout << endl;**

**for (int row = 0;row < 12;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 12;col++)**

**{**

**cout << setw(4) << arr5[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void grid6(char arr6[12][12])**

**{**

**system("cls");**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7" << setw(4) << "C 8 " << setw(4) << "C 9 " << setw(4) << "C 10 " << setw(4) << "C 11 ";**

**cout << endl;**

**for (int row = 0;row < 12;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 12;col++)**

**{**

**cout << setw(4) << arr6[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void grid7(char arr7[12][12])**

**{**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7" << setw(4) << "C 8 " << setw(4) << "C 9 " << setw(4) << "C 10 " << setw(4) << "C 11 ";**

**cout << endl;**

**for (int row = 0;row < 12;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 12;col++)**

**{**

**cout << setw(4) << arr7[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void grid8(char arr8[12][12])**

**{**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7" << setw(4) << "C 8 " << setw(4) << "C 9 " << setw(4) << "C 10 " << setw(4) << "C 11 ";**

**cout << endl;**

**for (int row = 0;row < 12;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 12;col++)**

**{**

**cout << setw(4) << arr8[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void grid9(char arr9[15][15])**

**{**

**system("cls");**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7" << setw(4) << "C 8 " << setw(4) << "C 9 " << setw(4) << "C 10 " << setw(4) << "C 11 " << setw(4) << "C12 " << setw(4) << "C13 " << setw(4) << "C14 ";**

**cout << endl;**

**for (int row = 0;row < 15;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 15;col++)**

**{**

**cout << setw(4) << arr9[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void grid10(char arr10[15][15])**

**{**

**system("cls");**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7" << setw(4) << "C 8 " << setw(4) << "C 9 " << setw(4) << "C 10 " << setw(4) << "C 11 " << setw(4) << "C12 " << setw(4) << "C13 " << setw(4) << "C14 ";**

**cout << endl;**

**for (int row = 0;row < 15;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 15;col++)**

**{**

**cout << setw(4) << arr10[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void grid11(char arr11[15][15])**

**{**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7" << setw(4) << "C 8 " << setw(4) << "C 9 " << setw(4) << "C 10 " << setw(4) << "C 11 " << setw(4) << "C12 " << setw(4) << "C13 " << setw(4) << "C14 ";**

**cout << endl;**

**for (int row = 0;row < 15;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 15;col++)**

**{**

**cout << setw(4) << arr11[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void grid12(char arr12[15][15])**

**{**

**cout << setw(13) << "C 0" << setw(4) << "C 1" << setw(4) << "C 2" << setw(4) << "C 3" << setw(4) << "C 4" << setw(4) << "C 5" << setw(4) << "C 6" << setw(4) << "C 7" << setw(4) << "C 8 " << setw(4) << "C 9 " << setw(4) << "C 10 " << setw(4) << "C 11 " << setw(4) << "C 12 " << setw(4) << "C 13 " << setw(4) << "C 14 ";**

**cout << endl;**

**for (int row = 0;row < 15;row++)**

**{**

**cout << setw(8) << "R " << row;**

**for (int col = 0;col < 15;col++)**

**{**

**cout << setw(4) << arr12[row][col];**

**}**

**cout << endl;**

**}**

**cout << endl;**

**}**

**void mainmenu()**

**{**

**HANDLE h;**

**h = GetStdHandle(STD\_OUTPUT\_HANDLE);**

**SetConsoleTextAttribute(h, 4);**

**int num;**

**cout << "1. Play Game " << endl;**

**cout << "2. Rules " << endl;**

**cout << "3. High score " << endl;**

**cout << "4. Credits " << endl;**

**cout << endl;**

**cout << "Select Option: ";**

**cin >> num;**

**cout << endl;**

**if (num == 1)**

**{**

**cout << "1. Player vs Player \n2. Player vs Machine" << endl;**

**cout << endl;**

**cout << "Select Game Mode: ";**

**cin >> num;**

**cout << endl;**

**if (num == 1)**

**{**

**cout << "1. easy" << endl;**

**cout << "2. medium" << endl;**

**cout << "3. hard" << endl;**

**cout << endl;**

**cout << "Select difficulty: ";**

**cin >> num;**

**cout << endl;**

**if (num == 1)**

**{**

**player\_vs\_player\_easy();**

**}**

**else if (num == 2)**

**{**

**player\_vs\_player\_medium();**

**}**

**else if (num == 3)**

**{**

**player\_vs\_player\_hard();**

**}**

**else**

**{**

**cout << "Invalid Input!!";**

**}**

**}**

**else if (num == 2)**

**{**

**cout << "1. easy" << endl;**

**cout << "2. medium" << endl;**

**cout << "3. hard" << endl;**

**cout << endl;**

**cout << "Select difficulty: ";**

**cin >> num;**

**cout << endl;**

**if (num == 1)**

**{**

**player\_vs\_machine\_Easy(); // Calling the Player VS Machine Function**

**}**

**else if (num == 2)**

**{**

**player\_vs\_Machine\_Medium();**

**}**

**else if (num == 3)**

**{**

**player\_vs\_Macine\_Hard();**

**}**

**else**

**{**

**cout << "Invalid Input!!";**

**}**

**}**

**else**

**{**

**cout << "Invalid Input!!";**

**}**

**}**

**else if (num == 2)**

**{**

**instructions();**

**}**

**else if (num == 3)**

**{**

**cout << "Not available yet.";**

**}**

**else if (num == 4)**

**{**

**credits();**

**}**

**else**

**{**

**cout << "Invalid Input!!";**

**}**

**}**

**//toss to decide which player will go first**

**//it doesn't output anything and decides the turn as the game starts**

**int toss(int Player)**

**{**

**bool x = true, y = false;**

**int max = 2;**

**srand(time(0));**

**max = rand() % max;**

**if (max == 0)**

**{**

**return x;**

**}**

**else**

**{**

**return y;**

**}**

**}**

**void player\_vs\_player\_easy()**

**{**

**const int a = 8, b = 5, c = 4, d = 3, e = 2;**

**char carrier[a] = { 'A','A','A','A','A','A','A','A' };**

**char battleships[b] = { 'B','B','B','B','B' };**

**char cruisers[c] = { 'C','C','C','C' };**

**char submarines[d] = { 'D','D','D' };**

**char destroyers[e] = { 'E','E' };**

**int count\_p1 = 0, count\_p2 = 0;**

**char arr2[8][8];**

**char arr1[8][8];**

**bool end = false, toss1, player1 = false, player2 = false;**

**int turn = 0, row, column;**

**char player = 'X', align;**

**cout << "\t" << "Player 1 = X Player 2 = O\n\n";**

**grid1(arr1);**

**cout << "Player 1 Turn to place ships: " << endl;**

**for (int i = 0;i < 5;i++)**

**{**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align; //align takes character input to set ships vertically or horizontally**

**cout << "Enter Row: ";**

**cin >> row; //taking x-axis co-ordinates from the player**

**cout << "Enter column: ";**

**cin >> column; //taking y-axis co-ordinates from the player**

**cout << endl;**

**switch (turn) //ships are placed on the arr 1 by player 1**

**{**

**case 0:**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr1[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr1[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid1(arr1); //grid is displayed after every ship placed to show the user which spaces are left and where the ship is placed**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr1[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr1[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid1(arr1);**

**break; //grid 1 function displays arr 1**

**case 1:**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr1[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr1[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid1(arr1);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr1[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr1[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid1(arr1);**

**break;**

**case 2:**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr1[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr1[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid1(arr1);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr1[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr1[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid1(arr1);**

**break;**

**case 3:**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr1[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr1[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid1(arr1);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr1[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr1[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid1(arr1);**

**break;**

**case 4:**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr1[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr1[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid1(arr1);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr1[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr1[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid1(arr1);**

**break;**

**}**

**}**

**grid2(arr2); //grid 2 function displays arr2 (grid number displays corresponding array number that it represents)**

**turn = 0;**

**cout << "Player 2 Turn to place ships: " << endl;**

**for (int i = 0;i < 5;i++) //player 2 places its ships on arr 2**

**{**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**switch (turn)**

**{**

**case 0:**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr2[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr2[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid2(arr2);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr2[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr2[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid2(arr2);**

**break;**

**case 1:**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr2[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr2[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid2(arr2);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr2[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr2[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid2(arr2);**

**break;**

**case 2:**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr2[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr2[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid2(arr2);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr2[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr2[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid2(arr2);**

**break;**

**case 3:**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr2[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr2[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid2(arr2);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr2[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr2[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid2(arr2);**

**break;**

**case 4:**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr2[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr2[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid2(arr2);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr2[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr2[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid2(arr2);**

**break;**

**}**

**}**

**system("cls");**

**toss1 = toss(player); //Toss decides which player goes first after both players place their ships**

**if (toss1 == true)**

**{**

**player = 'X';**

**}**

**else**

**{**

**player = 'O';**

**}**

**cout << "\t" << "Player 1 = X Player 2 = O\n\n"; //When player 1 destroys a player 2 ship 'X' is displayed on the guessing grid of player 1 and similaryly 'O' is displayed on pplayer 2's guessing grid when player 2 destroys a player 1 ship**

**if (player == 'X') //if player 1 wins toss turns take place in this order**

**{**

**do**

**{**

**cout << "Player 1 turn: " << endl;**

**grid3(arr3);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr2[row][column] == 'A' || arr2[row][column] == 'B' || arr2[row][column] == 'C' || arr2[row][column] == 'D' || arr2[row][column] == 'E')**

**{**

**arr3[row][column] = 'X';**

**count\_p1++; //count incremented as player 1 destroys a player 2 ship**

**}**

**else**

**{**

**arr3[row][column] = '.';**

**}**

**grid3(arr3);**

**if (count\_p1 == 44) //this is the winning condition. As all ships which in this case are 15 ships are destroyed count is checked and to end the loop break is used to break out of the loop**

**{**

**end = true; //loop works till end=false so it is set to true when all ships are destroyed to end the loop**

**player1 = true;**

**break;**

**}**

**cout << endl;**

**cout << "Player 2 turn: " << endl;**

**grid4(arr4);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr1[row][column] == 'A' || arr1[row][column] == 'B' || arr1[row][column] == 'C' || arr1[row][column] == 'D' || arr1[row][column] == 'E') //this if condition checks if their is an opponent ship on the place guessed by the player and accordingly inputs 'X'/'O'(according to the player) or '.' if their is no ship on guessed location**

**{**

**arr4[row][column] = 'O';**

**count\_p2++; //count incremented as player 2 destroys a player 1 ship**

**}**

**else**

**{**

**arr4[row][column] = '.';**

**}**

**grid4(arr4);**

**if (count\_p2 == 44) //This is our winning condition for player 2**

**{**

**end = true;**

**player2 = true;**

**}**

**} while (end == false);**

**if (player1 == true) //displays result according to which player gets to count 15**

**{**

**cout << "Player 1 wins!!" << endl;**

**cout << "\a" << "Player 2 looses!!" << endl;**

**}**

**else if (player2 == true)**

**{**

**cout << "Player 2 wins!!" << endl;**

**cout << "\a" << "Player 1 looses!!" << endl;**

**}**

**}**

**else if (player == 'O') //if player 2 wins toss turns take place in this order**

**{**

**do**

**{**

**cout << "Player 2 turn: " << endl;**

**grid4(arr4);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr1[row][column] == 'A' || arr1[row][column] == 'B' || arr1[row][column] == 'C' || arr1[row][column] == 'D' || arr1[row][column] == 'E')**

**{**

**arr4[row][column] = 'O';**

**count\_p2++;**

**}**

**else**

**{**

**arr4[row][column] = '.';**

**}**

**grid4(arr4);**

**if (count\_p2 == 44)**

**{**

**end = true;**

**player2 = true;**

**break; //break added here to end the loop and stop rest of the program as the player wins**

**}**

**cout << endl;**

**cout << "Player 1 turn: " << endl;**

**grid3(arr3);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr2[row][column] == 'A' || arr2[row][column] == 'B' || arr2[row][column] == 'C' || arr2[row][column] == 'D' || arr2[row][column] == 'E')**

**{**

**arr3[row][column] = 'X';**

**count\_p1++;**

**}**

**else**

**{**

**arr3[row][column] = '.';**

**}**

**grid3(arr3);**

**if (count\_p1 == 44)**

**{**

**end = true;**

**player1 = true;**

**}**

**} while (end == false);**

**if (player1 == true)**

**{**

**cout << "Player 1 wins!!" << endl;**

**cout << "\a" << "Player 2 looses!!" << endl;**

**}**

**else if (player2 == true)**

**{**

**cout << "Player 2 wins!!" << endl;**

**cout << "\a" << "Player 1 looses!!" << endl;**

**}**

**}**

**}**

**void player\_vs\_player\_medium() //for medium and hard difficulty only the size of the ships, arrays and grids displayed were changed the working is the same as in easy difficulty**

**{**

**const int a = 8, b = 5, c = 4, d = 3, e = 2;**

**char carrier[a] = { 'A','A','A','A','A','A','A','A' };**

**char battleships[b] = { 'B','B','B','B','B' };**

**char cruisers[c] = { 'C','C','C','C' };**

**char submarines[d] = { 'D','D','D' };**

**char destroyers[e] = { 'E','E' };**

**int count\_p1 = 0, count\_p2 = 0;**

**char arr6[12][12];**

**char arr5[12][12];**

**bool end = false, toss1, player1 = false, player2 = false;**

**int turn = 0, row, column;**

**char player = 'X', align;**

**cout << "\t" << "Player 1 = X Player 2 = O\n\n";**

**grid5(arr5);**

**cout << "Player 1 Turn to place ships: " << endl;**

**for (int i = 0;i < 5;i++)**

**{**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**switch (turn)**

**{**

**case 0:**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid5(arr5);**

**break;**

**case 1:**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid5(arr5);**

**break;**

**case 2:**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid5(arr5);**

**break;**

**case 3:**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid5(arr5);**

**break;**

**case 4:**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid5(arr5);**

**break;**

**}**

**}**

**grid6(arr6);**

**turn = 0;**

**cout << "Player 2 Turn to place ships: " << endl;**

**for (int i = 0;i < 5;i++)**

**{**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**switch (turn)**

**{**

**case 0:**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid6(arr6);**

**break;**

**case 1:**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid6(arr6);**

**break;**

**case 2:**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid6(arr6);**

**break;**

**case 3:**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid6(arr6);**

**break;**

**case 4:**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid6(arr6);**

**break;**

**}**

**}**

**system("cls");**

**toss1 = toss(player);**

**if (toss1 == true)**

**{**

**player = 'X';**

**}**

**else**

**{**

**player = 'O';**

**}**

**cout << "\t" << "Player 1 = X Player 2 = O\n\n";**

**if (player == 'X')**

**{**

**do**

**{**

**cout << "Player 1 turn: " << endl;**

**grid7(arr7);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr6[row][column] == 'A' || arr6[row][column] == 'B' || arr6[row][column] == 'C' || arr6[row][column] == 'D' || arr6[row][column] == 'E')**

**{**

**arr7[row][column] = 'X';**

**count\_p1++;**

**}**

**else**

**{**

**arr7[row][column] = '.';**

**}**

**grid7(arr7);**

**if (count\_p1 == 66)**

**{**

**end = true;**

**player1 = true;**

**break;**

**}**

**cout << endl;**

**cout << "Player 2 turn: " << endl;**

**grid8(arr8);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr5[row][column] == 'A' || arr5[row][column] == 'B' || arr5[row][column] == 'C' || arr5[row][column] == 'D' || arr5[row][column] == 'E')**

**{**

**arr8[row][column] = 'O';**

**count\_p2++;**

**}**

**else**

**{**

**arr8[row][column] = '.';**

**}**

**grid8(arr8);**

**if (count\_p2 == 66)**

**{**

**end = true;**

**player2 = true;**

**}**

**} while (end == false);**

**if (player1 == true)**

**{**

**cout << "Player 1 wins!!" << endl;**

**}**

**else if (player2 == true)**

**{**

**cout << "Player 2 wins!!" << endl;**

**}**

**}**

**else if (player == 'O')**

**{**

**do**

**{**

**cout << "Player 2 turn: " << endl;**

**grid8(arr8);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr5[row][column] == 'A' || arr5[row][column] == 'B' || arr5[row][column] == 'C' || arr5[row][column] == 'D' || arr5[row][column] == 'E')**

**{**

**arr8[row][column] = 'O';**

**count\_p2++;**

**}**

**else**

**{**

**arr8[row][column] = '.';**

**}**

**grid8(arr8);**

**if (count\_p2 == 66)**

**{**

**end = true;**

**player2 = true;**

**break;**

**}**

**cout << endl;**

**cout << "Player 1 turn: " << endl;**

**grid7(arr7);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr6[row][column] == 'A' || arr6[row][column] == 'B' || arr6[row][column] == 'C' || arr6[row][column] == 'D' || arr6[row][column] == 'E')**

**{**

**arr7[row][column] = 'X';**

**count\_p1++;**

**}**

**else**

**{**

**arr7[row][column] = '.';**

**}**

**grid7(arr7);**

**if (count\_p1 == 66)**

**{**

**end = true;**

**player1 = true;**

**}**

**} while (end == false);**

**if (player1 == true)**

**{**

**cout << "Player 1 wins!!" << endl;**

**cout << "\a" << "Player 1 looses!!" << endl;**

**}**

**else if (player2 == true)**

**{**

**cout << "Player 2 wins!!" << endl;**

**cout << "\a" << "Player 1 looses!!" << endl;**

**}**

**}**

**}**

**void player\_vs\_player\_hard()**

**{**

**const int a = 8, b = 5, c = 4, d = 3, e = 2;**

**char carrier[a] = { 'A','A','A','A','A','A','A','A' };**

**char battleships[b] = { 'B','B','B','B','B' };**

**char cruisers[c] = { 'C','C','C','C' };**

**char submarines[d] = { 'D','D','D' };**

**char destroyers[e] = { 'E','E' };**

**int count\_p1 = 0, count\_p2 = 0;**

**char arr10[15][15];**

**char arr9[15][15];**

**bool end = false, toss1, player1 = false, player2 = false;**

**int turn = 0, row, column;**

**char player = 'X', align;**

**cout << "\t" << "Player 1 = X Player 2 = O\n\n";**

**grid9(arr9);**

**cout << "Player 1 Turn to place ships: " << endl;**

**for (int i = 0;i < 5;i++)**

**{**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**switch (turn)**

**{**

**case 0:**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid9(arr9);**

**break;**

**case 1:**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid9(arr9);**

**break;**

**case 2:**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid9(arr9);**

**break;**

**case 3:**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid9(arr9);**

**break;**

**case 4:**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr9[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr9[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr9[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr9[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr9[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr9[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr9[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr9[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid9(arr9);**

**break;**

**}**

**}**

**grid10(arr10);**

**turn = 0;**

**cout << "Player 2 Turn to place ships: " << endl;**

**for (int i = 0;i < 5;i++)**

**{**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**switch (turn)**

**{**

**case 0:**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid10(arr10);**

**break;**

**case 1:**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr10[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr10[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr10[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr10[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr10[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr10[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr10[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr10[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid10(arr10);**

**break;**

**case 2:**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid10(arr10);**

**break;**

**case 3:**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr10[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr10[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr10[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr10[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr10[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr10[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr10[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr10[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid10(arr10);**

**break;**

**case 4:**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr10[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr10[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr10[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr10[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr10[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr10[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid10(arr10);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr10[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr10[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid10(arr10);**

**break;**

**}**

**}**

**system("cls");**

**toss1 = toss(player);**

**if (toss1 == true)**

**{**

**player = 'X';**

**}**

**else**

**{**

**player = 'O';**

**}**

**cout << "\t" << "Player 1 = X Player 2= O\n\n";**

**if (player == 'X')**

**{**

**do**

**{**

**cout << "Player 1 turn: " << endl;**

**grid11(arr11);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr10[row][column] == 'A' || arr10[row][column] == 'B' || arr10[row][column] == 'C' || arr10[row][column] == 'D' || arr10[row][column] == 'E')**

**{**

**arr11[row][column] = 'X';**

**count\_p1++;**

**}**

**else**

**{**

**arr11[row][column] = '.';**

**}**

**grid11(arr11);**

**if (count\_p1 == 88)**

**{**

**end = true;**

**player1 = true;**

**break;**

**}**

**cout << endl;**

**cout << "Player 2 turn: " << endl;**

**grid12(arr12);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr9[row][column] == 'A' || arr9[row][column] == 'B' || arr9[row][column] == 'C' || arr9[row][column] == 'D' || arr9[row][column] == 'E')**

**{**

**arr12[row][column] = 'O';**

**count\_p2++;**

**}**

**else**

**{**

**arr12[row][column] = '.';**

**}**

**grid12(arr12);**

**if (count\_p2 == 88)**

**{**

**end = true;**

**player2 = true;**

**}**

**} while (end == false);**

**if (player1 == true)**

**{**

**cout << "Player 1 wins!!" << endl;**

**}**

**else if (player2 == true)**

**{**

**cout << "Player 2 wins!!" << endl;**

**}**

**}**

**else if (player == 'O')**

**{**

**do**

**{**

**cout << "Player 2 turn: " << endl;**

**grid12(arr12);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr9[row][column] == 'A' || arr9[row][column] == 'B' || arr9[row][column] == 'C' || arr9[row][column] == 'D' || arr9[row][column] == 'E')**

**{**

**arr12[row][column] = 'O';**

**count\_p2++;**

**}**

**else**

**{**

**arr12[row][column] = '.';**

**}**

**grid12(arr12);**

**if (count\_p2 == 88)**

**{**

**end = true;**

**player2 = true;**

**break;**

**}**

**cout << endl;**

**cout << "Player 1 turn: " << endl;**

**grid11(arr11);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr10[row][column] == 'A' || arr10[row][column] == 'B' || arr10[row][column] == 'C' || arr10[row][column] == 'D' || arr10[row][column] == 'E')**

**{**

**arr11[row][column] = 'X';**

**count\_p1++;**

**}**

**else**

**{**

**arr11[row][column] = '.';**

**}**

**grid11(arr11);**

**if (count\_p1 == 88)**

**{**

**end = true;**

**player1 = true;**

**}**

**} while (end == false);**

**if (player1 == true)**

**{**

**cout << "Player 1 wins!!" << endl;**

**cout << "\a" << "Player 2 looses!!" << endl;**

**}**

**else if (player2 == true)**

**{**

**cout << "Player 2 wins!!" << endl;**

**cout << "\a" << "Player 1 looses!!" << endl;**

**}**

**}**

**}**

**//======================================== Player VS Machine =====================================================**

**// Easy\_\_\_Mode**

**void player\_vs\_machine\_Easy()**

**{**

**const int a = 8, b = 5, c = 4, d = 3, e = 2;**

**char carrier[a] = { 'A','A','A','A','A','A','A','A' };**

**char battleships[b] = { 'B','B','B','B','B' };**

**char cruisers[c] = { 'C','C','C','C' };**

**char submarines[d] = { 'D','D','D' };**

**char destroyers[e] = { 'E','E' };**

**int count\_p1 = 0, count\_p2 = 0;**

**char arr2[8][8];**

**char arr1[8][8];**

**bool end = false, toss1, player1 = false, machine = false;**

**int turn = 0, row, column;**

**char player = 'X', align;**

**cout << "\t" << "Player = X Machine = O\n\n";**

**grid1(arr1);**

**cout << "--------- Machine Turn to place ships -------- " << endl;**

**for (int i = 0;i < 5;i++)**

**{**

**cout << "Enter Alignment of ship with v or h: ";**

**int align = 0 + (rand() % 2); // For Generating Random number 0 or 1 (0 for Vertical and 1 for Horizontal)**

**cout << "Align = " << align << endl;**

**if (align == 0)**

**cout << "Machine choosed Vertical " << endl;**

**else**

**cout << "Machine choosed Horizontal " << endl;**

**int row = 0 + (rand() % 8);**

**int column = 0 + (rand() % 8);**

**cout << "Machine's Random Position for Row is: " << row << " Colums is: " << column << endl;**

**switch (turn) //ships are placed on the arr 1 by player 1**

**{**

**case 0:**

**if (align == 1)**

**{**

**for (int j = 0; j < a; j++)**

**{**

**arr1[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr1[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid1(arr1);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 8);**

**column = 0 + (rand() % 8);**

**if (align == 1)**

**{**

**for (int j = 0; j < a; j++)**

**{**

**arr1[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr1[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid1(arr1);**

**break;**

**case 1:**

**if (align == 1)**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr1[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr1[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid1(arr1);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 8);**

**column = 0 + (rand() % 8);**

**if (align == 1)**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr1[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr1[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid1(arr1);**

**break;**

**case 2:**

**if (align == 1)**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr1[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr1[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid1(arr1);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 8);**

**column = 0 + (rand() % 8);**

**if (align == 1)**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr1[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr1[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid1(arr1);**

**break;**

**case 3:**

**if (align == 1)**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr1[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr1[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid1(arr1);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 8);**

**column = 0 + (rand() % 8);**

**if (align == 1)**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr1[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr1[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid1(arr1);**

**break;**

**case 4:**

**if (align == 1)**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr1[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr1[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid1(arr1);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 8);**

**column = 0 + (rand() % 8);**

**if (align == 1)**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr1[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr1[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid1(arr1);**

**break;**

**}**

**}**

**grid2(arr2);**

**turn = 0;**

**cout << "Player Turn to place ships: " << endl;**

**for (int i = 0;i < 5;i++)**

**{**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**switch (turn)**

**{**

**case 0:**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr2[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr2[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid2(arr2);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr2[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr2[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid2(arr2);**

**break;**

**case 1:**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr2[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr2[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid2(arr2);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr2[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr2[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid2(arr2);**

**break;**

**case 2:**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr2[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr2[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid2(arr2);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr2[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr2[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid2(arr2);**

**break;**

**case 3:**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr2[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr2[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid2(arr2);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr2[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr2[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid2(arr2);**

**break;**

**case 4:**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr2[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr2[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid2(arr2);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr2[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr2[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid2(arr2);**

**break;**

**}**

**}**

**system("cls");**

**toss1 = toss(player);**

**if (toss1 == true)**

**{**

**player = 'X';**

**}**

**else**

**{**

**player = 'O';**

**}**

**cout << "\t" << "Player 1 = X Machine = O\n\n";**

**if (player == 'X')**

**{**

**do**

**{**

**cout << "Player 1 turn: " << endl;**

**grid3(arr3);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr2[row][column] == 'A' || arr2[row][column] == 'B' || arr2[row][column] == 'C' || arr2[row][column] == 'D' || arr2[row][column] == 'E')**

**{**

**arr3[row][column] = 'X';**

**count\_p1++;**

**}**

**else**

**{**

**arr3[row][column] = '.';**

**}**

**grid3(arr3);**

**if (count\_p1 == 44)**

**{**

**end = true;**

**player1 = true;**

**break;**

**}**

**cout << endl;**

**cout << "Machine turn: " << endl;**

**grid4(arr4);**

**cout << "Enter Row: ";**

**row= rand() % 7;**

**cout << row<<endl;**

**cout << "Enter column: ";**

**column = rand() % 7;**

**cout << column;**

**cout << endl;**

**if (arr1[row][column] == 'A' || arr1[row][column] == 'B' || arr1[row][column] == 'C' || arr1[row][column] == 'D' || arr1[row][column] == 'E')**

**{**

**arr4[row][column] = 'O';**

**count\_p2++;**

**}**

**else**

**{**

**arr4[row][column] = '.';**

**}**

**grid4(arr4);**

**if (count\_p2 == 44)**

**{**

**end = true;**

**machine = true;**

**}**

**} while (end == false);**

**if (player1 == true)**

**{**

**cout << "Player 1 wins!!" << endl;**

**cout << "\a" << "Machine looses!!" << endl;**

**}**

**else if (machine == true)**

**{**

**cout << "Computer Wins !!" << endl;**

**cout << "\a" << "Player 1 looses!!" << endl;**

**}**

**}**

**else if (player == 'O')**

**{**

**do**

**{**

**cout << "Machine turn: " << endl;**

**grid4(arr4);**

**cout << "Enter Row: ";**

**row = rand() % 7;**

**cout << row<<endl;**

**cout << "Enter column: ";**

**column = rand() % 7;**

**cout << column;**

**cout << endl;**

**if (arr1[row][column] == 'A' || arr1[row][column] == 'B' || arr1[row][column] == 'C' || arr1[row][column] == 'D' || arr1[row][column] == 'E')**

**{**

**arr4[row][column] = 'O';**

**count\_p2++;**

**}**

**else**

**{**

**arr4[row][column] = '.';**

**}**

**grid4(arr4);**

**if (count\_p2 == 44)**

**{**

**end = true;**

**machine = true;**

**break;**

**}**

**cout << endl;**

**cout << "Player 1 turn: " << endl;**

**grid3(arr3);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr2[row][column] == 'A' || arr2[row][column] == 'B' || arr2[row][column] == 'C' || arr2[row][column] == 'D' || arr2[row][column] == 'E')**

**{**

**arr3[row][column] = 'X';**

**count\_p1++;**

**}**

**else**

**{**

**arr3[row][column] = '.';**

**}**

**grid3(arr3);**

**if (count\_p1 == 44)**

**{**

**end = true;**

**player1 = true;**

**}**

**} while (end == false);**

**if (player1 == true)**

**{**

**cout << "Player 1 wins!!" << endl;**

**cout << "\a" << "Machine looses!!" << endl;**

**}**

**else if (machine == true)**

**{**

**cout << "Machine wins!!" << endl;**

**cout << "\a" << "Player 1 looses!!" << endl;**

**}**

**}**

**}**

**// --------------------------------------------------------------------------------------**

**//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Medium \_ Mode\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**// Player 1 is Player and Player 2 is Machine**

**void player\_vs\_Machine\_Medium()**

**{**

**const int a = 8, b = 5, c = 4, d = 3, e = 2;**

**char carrier[a] = { 'A','A','A','A','A','A','A','A' };**

**char battleships[b] = { 'B','B','B','B','B' };**

**char cruisers[c] = { 'C','C','C','C' };**

**char submarines[d] = { 'D','D','D' };**

**char destroyers[e] = { 'E','E' };**

**int count\_p1 = 0, count\_p2 = 0;**

**char arr6[12][12];**

**char arr5[12][12];**

**bool end = false, toss1, player1 = false, player2 = false;**

**int turn = 0, row, column;**

**char player = 'X', align;**

**cout << "\t" << "Player 1 = X Macine = O\n\n";**

**grid5(arr5);**

**cout << "Player Turn to place ships: " << endl;**

**for (int i = 0;i < 5;i++)**

**{**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**switch (turn)**

**{**

**case 0:**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr5[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid5(arr5);**

**break;**

**case 1:**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr5[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid5(arr5);**

**break;**

**case 2:**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr5[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid5(arr5);**

**break;**

**case 3:**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr5[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid5(arr5);**

**break;**

**case 4:**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid5(arr5);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr5[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid5(arr5);**

**break;**

**}**

**}**

**grid6(arr6);**

**turn = 0;**

**cout << "Machine Turn to place ships: " << endl;**

**for (int i = 0;i < 5;i++)**

**{**

**cout << " Machine choosed: ";**

**int align = 0 + (rand() % 2); // For Generating Random number 0 or 1 (0 for Vertical and 1 for Horizontal)**

**cout << "Align = " << align << endl;**

**if (align == 0)**

**cout << "Machine choosed Vertical " << endl;**

**else**

**cout << "Machine choosed Horizontal " << endl;**

**int row = 0 + (rand() % 11);**

**int column = 0 + (rand() % 11);**

**switch (turn)**

**{**

**case 0:**

**if (align == 1)**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr6[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr6[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 11);**

**column = 0 + (rand() % 11);**

**if (align == 1)**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr6[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr6[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 11);**

**column = 0 + (rand() % 11);**

**if (align == 1)**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr6[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr6[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid6(arr6);**

**break;**

**case 1:**

**if (align == 0)**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 11);**

**column = 0 + (rand() % 11);**

**if (align == 0)**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 11);**

**column = 0 + (rand() % 11);**

**if (align == 0)**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr6[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid6(arr6);**

**break;**

**case 2:**

**if (align == 1)**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 11);**

**column = 0 + (rand() % 11);**

**if (align == 1)**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 11);**

**column = 0 + (rand() % 11);**

**if (align == 1)**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr6[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid6(arr6);**

**break;**

**case 3:**

**if (align == 1)**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 11);**

**column = 0 + (rand() % 11);**

**if (align == 1)**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 11);**

**column = 0 + (rand() % 11);**

**if (align == 1)**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 0)**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr6[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid6(arr6);**

**break;**

**case 4:**

**if (align == 1)**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 1)**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 11);**

**column = 0 + (rand() % 11);**

**if (align == 1)**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**column++;**

**}**

**}**

**else if (align == 1)**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**row++;**

**}**

**}**

**grid6(arr6);**

**align = 0 + (rand() % 2);**

**row = 0 + (rand() % 11);**

**column = 0 + (rand() % 11);**

**if (align == 1)**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 1)**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr6[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid6(arr6);**

**break;**

**}**

**}**

**system("cls");**

**toss1 = toss(player);**

**if (toss1 == true)**

**{**

**player = 'X';**

**}**

**else**

**{**

**player = 'O';**

**}**

**cout << "\t" << "Player 1 = X Machine = O\n\n";**

**if (player == 'X')**

**{**

**do**

**{**

**cout << "Player 1 turn: " << endl;**

**grid7(arr7);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr6[row][column] == 'A' || arr6[row][column] == 'B' || arr6[row][column] == 'C' || arr6[row][column] == 'D' || arr6[row][column] == 'E')**

**{**

**arr7[row][column] = 'X';**

**count\_p1++;**

**}**

**else**

**{**

**arr7[row][column] = '.';**

**}**

**grid7(arr7);**

**if (count\_p1 == 66)**

**{**

**end = true;**

**player1 = true;**

**break;**

**}**

**cout << endl;**

**cout << "Machine turn: " << endl;**

**grid8(arr8);**

**cout << "Enter Row: ";**

**row = rand() % 11;**

**cout << row << endl;**

**cout << "Enter column: ";**

**column = rand() % 11;**

**cout << column;**

**cout << endl;**

**if (arr5[row][column] == 'A' || arr5[row][column] == 'B' || arr5[row][column] == 'C' || arr5[row][column] == 'D' || arr5[row][column] == 'E')**

**{**

**arr8[row][column] = 'O';**

**count\_p2++;**

**}**

**else**

**{**

**arr8[row][column] = '.';**

**}**

**grid8(arr8);**

**if (count\_p2 == 66)**

**{**

**end = true;**

**player2 = true;**

**}**

**} while (end == false);**

**if (player1 == true)**

**{**

**cout << "Player wins!!" << endl;**

**}**

**else if (player2 == true)**

**{**

**cout << "Machine wins!!" << endl;**

**}**

**}**

**else if (player == 'O')**

**{**

**do**

**{**

**cout << "Machine turn: " << endl;**

**grid8(arr8);**

**cout << "Enter Row: ";**

**row = rand() % 11;**

**cout << row << endl;**

**cout << "Enter column: ";**

**column = rand() % 11;**

**cout << column;**

**cout << endl;**

**if (arr5[row][column] == 'A' || arr5[row][column] == 'B' || arr5[row][column] == 'C' || arr5[row][column] == 'D' || arr5[row][column] == 'E')**

**{**

**arr8[row][column] = 'O';**

**count\_p2++;**

**}**

**else**

**{**

**arr8[row][column] = '.';**

**}**

**grid8(arr8);**

**if (count\_p2 == 66)**

**{**

**end = true;**

**player2 = true;**

**break;**

**}**

**cout << endl;**

**cout << "Player 1 turn: " << endl;**

**grid7(arr7);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr6[row][column] == 'A' || arr6[row][column] == 'B' || arr6[row][column] == 'C' || arr6[row][column] == 'D' || arr6[row][column] == 'E')**

**{**

**arr7[row][column] = 'X';**

**count\_p1++;**

**}**

**else**

**{**

**arr7[row][column] = '.';**

**}**

**grid7(arr7);**

**if (count\_p1 == 66)**

**{**

**end = true;**

**player1 = true;**

**}**

**} while (end == false);**

**if (player1 == true)**

**{**

**cout << "Player 1 wins!!" << endl;**

**cout << "\a" << "Machine looses!!" << endl;**

**}**

**else if (player2 == true)**

**{**

**cout << "Machine wins!!" << endl;**

**cout << "\a" << "Player 1 looses!!" << endl;**

**}**

**}**

**}**

**// ---------------------------------------------------------------------------------------------------------**

**// Hard\_\_Mode (Player VS Machine)**

**// Player 1 is Player and Player 2 is Machine**

**void player\_vs\_Macine\_Hard()**

**{**

**const int a = 8, b = 5, c = 4, d = 3, e = 2;**

**char carrier[a] = { 'A','A','A','A','A','A','A','A' };**

**char battleships[b] = { 'B','B','B','B','B' };**

**char cruisers[c] = { 'C','C','C','C' };**

**char submarines[d] = { 'D','D','D' };**

**char destroyers[e] = { 'E','E' };**

**int count\_p1 = 0, count\_p2 = 0;**

**char arr10[15][15];**

**char arr9[15][15];**

**bool end = false, toss1, player1 = false, player2 = false;**

**int turn = 0, row, column;**

**char player = 'X', align;**

**cout << "\t" << "Player 1 = X Machine= O\n\n";**

**grid9(arr9);**

**cout << "Player Turn to place ships: " << endl;**

**for (int i = 0;i < 5;i++)**

**{**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**switch (turn)**

**{**

**case 0:**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr9[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid9(arr9);**

**break;**

**case 1:**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr9[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid9(arr9);**

**break;**

**case 2:**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr9[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid9(arr9);**

**break;**

**case 3:**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**column++;**

**}**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**row++;**

**}**

**}**

**grid9(arr9);**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (align == 'h')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < d;j++)**

**{**

**arr9[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid9(arr9);**

**break;**

**case 4:**

**if (align == 'h')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr9[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < e;j++)**

**{**

**arr9[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid9(arr9);**

**break;**

**}**

**}**

**grid10(arr10);**

**turn = 0;**

**cout << "Player 2 Turn to place ships: " << endl;**

**for (int i = 0;i < 5;i++)**

**{**

**cout << "Enter Alignment of ship with v or h: ";**

**cin >> align;**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**switch (turn)**

**{**

**case 0:**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = carrier[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = carrier[j];**

**row++;**

**}**

**turn++;**

**}**

**grid10(arr10);**

**break;**

**case 1:**

**if (align == 'h')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr10[row][column] = battleships[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < b;j++)**

**{**

**arr10[row][column] = battleships[j];**

**row++;**

**}**

**turn++;**

**}**

**grid10(arr10);**

**break;**

**case 2:**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = cruisers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = cruisers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid10(arr10);**

**break;**

**case 3:**

**if (align == 'h')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = submarines[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < c;j++)**

**{**

**arr10[row][column] = submarines[j];**

**row++;**

**}**

**turn++;**

**}**

**grid10(arr10);**

**break;**

**case 4:**

**if (align == 'h')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = destroyers[j];**

**column++;**

**}**

**turn++;**

**}**

**else if (align == 'v')**

**{**

**for (int j = 0;j < a;j++)**

**{**

**arr10[row][column] = destroyers[j];**

**row++;**

**}**

**turn++;**

**}**

**grid10(arr10);**

**break;**

**}**

**}**

**system("cls");**

**toss1 = toss(player);**

**if (toss1 == true)**

**{**

**player = 'X';**

**}**

**else**

**{**

**player = 'O';**

**}**

**cout << "\t" << "Player 1 = X Machine = O\n\n";**

**if (player == 'X')**

**{**

**do**

**{**

**cout << "Player turn: " << endl;**

**grid11(arr11);**

**int row = 0 + (rand() % 14);**

**int column = 0 + (rand() % 14);**

**cout << "Machine's Random Position for Row is: " << row << " Colums is: " << column << endl;**

**if (arr10[row][column] == 'A' || arr10[row][column] == 'B' || arr10[row][column] == 'C' || arr10[row][column] == 'D' || arr10[row][column] == 'E')**

**{**

**arr11[row][column] = 'X';**

**count\_p1++;**

**}**

**else**

**{**

**arr11[row][column] = '.';**

**}**

**grid11(arr11);**

**if (count\_p1 == 45)**

**{**

**end = true;**

**player1 = true;**

**break;**

**}**

**cout << endl;**

**cout << "Machine turn: " << endl;**

**grid12(arr12);**

**//**

**// int row = 0 + (rand() % 8 );**

**// int column = 0 + (rand() % 8);**

**// cout << "Machine's Random Position for Row is: " << row << " Colums is: " << column << endl;**

**if (arr9[row][column] == 'A' || arr9[row][column] == 'B' || arr9[row][column] == 'C' || arr9[row][column] == 'D' || arr9[row][column] == 'E')**

**{**

**arr12[row][column] = 'O';**

**count\_p2++;**

**}**

**else**

**{**

**arr12[row][column] = '.';**

**}**

**grid12(arr12);**

**if (count\_p2 == 45)**

**{**

**end = true;**

**player2 = true;**

**}**

**} while (end == false);**

**if (player1 == true)**

**{**

**cout << "Player 1 wins!!" << endl;**

**}**

**else if (player2 == true)**

**{**

**cout << "Machine wins!!" << endl;**

**}**

**}**

**else if (player == 'O')**

**{**

**do**

**{**

**cout << "Machine turn: " << endl;**

**grid12(arr12);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr9[row][column] == 'A' || arr9[row][column] == 'B' || arr9[row][column] == 'C' || arr9[row][column] == 'D' || arr9[row][column] == 'E')**

**{**

**arr12[row][column] = 'O';**

**count\_p2++;**

**}**

**else**

**{**

**arr12[row][column] = '.';**

**}**

**grid12(arr12);**

**if (count\_p2 == 45)**

**{**

**end = true;**

**player2 = true;**

**break;**

**}**

**cout << endl;**

**cout << "Player 1 turn: " << endl;**

**grid11(arr11);**

**cout << "Enter Row: ";**

**cin >> row;**

**cout << "Enter column: ";**

**cin >> column;**

**cout << endl;**

**if (arr10[row][column] == 'A' || arr10[row][column] == 'B' || arr10[row][column] == 'C' || arr10[row][column] == 'D' || arr10[row][column] == 'E')**

**{**

**arr11[row][column] = 'X';**

**count\_p1++;**

**}**

**else**

**{**

**arr11[row][column] = '.';**

**}**

**grid11(arr11);**

**if (count\_p1 == 45)**

**{**

**end = true;**

**player1 = true;**

**}**

**} while (end == false);**

**if (player1 == true)**

**{**

**cout << "Player 1 wins!!" << endl;**

**cout << "\a" << "Machine looses!!" << endl;**

**}**

**else if (player2 == true)**

**{**

**cout << "Machine wins!!" << endl;**

**cout << "\a" << "Player 1 looses!!" << endl;**

**}**

**}**

**}**