```cpp
#include <iostream>

using namespace std;

class Board
{
    int data[9];
    int magic_square[9];
    int cpu_owned[4], human_owned[4];
    int cpu_owned_index, human_owned_index;

public:
    Board()
    {
        magic_square[0] = 8;
        magic_square[1] = 3;
        magic_square[2] = 4;
        magic_square[3] = 1;
        magic_square[4] = 5;
        magic_square[5] = 9;
        magic_square[6] = 6;
        magic_square[7] = 7;
        magic_square[8] = 2;
        for(int i=0 ; i<9 ; i++)
        {
            data[i] = 2;
        }
        cpu_owned_index = 0;
        human_owned_index = 0;
    }

    void display()
    {
        int k=0;
        for(int i=0 ; i<3 ; i++)
        {
            cout << "|";
            for (int j = 0; j < 3; ++j)
            {
                if(data[k] == 2)
                    cout << " " << "|";
                else if(data[k] == 3)
                    cout << "x" << "|";
                else
                    cout << "o" << "|";
                k++;
            }
            cout << endl;
        }
    }

    int make2()
    {
        if(data[4] == 2)
            return 4;
        else
        {
            int non_corner_square[4] = {1,3,5,7};
            for (int i = 0; i < 4; ++i)
            {
                if(data[non_corner_square[i]] == 2)
                    return non_corner_square[i];
            }
        }
    }

    int posswin(char p)
    {
        int target;
```

```
        if(p == 'x')
        {
            for(int i=0 ; i<cpu_owned_index ; i++)
            {
                for(int j=i+1 ; j<cpu_owned_index ; j++)
                {
                    int diff = 15 - magic_square[cpu_owned[i]] - magic_square
[cpu_owned[j]];
                    if(!(diff < 0 || diff > 9))
                    {
                        for(int kk=0 ; kk<9 ; kk++)
                        {
                            if(magic_square[kk] == diff)
                            {
                                if(data[kk] == 2)
                                    return kk;
                            }
                        }
                    }

                }
            }
        }
        else
        {
            for(int i=0 ; i<human_owned_index ; i++)
            {
                for(int j=i+1 ; j<human_owned_index ; j++)
                {
                    int diff = 15 - magic_square[human_owned[i]] - magic_square
[human_owned[j]];
                    if(!(diff < 0 || diff > 9))
                    {
                        for(int kk=0 ; kk<9 ; kk++)
                        {
                            if(magic_square[kk] == diff)
                            {
                                if(data[kk] == 2)
                                    return kk;
                            }
                        }
                    }

                }
            }
        }
        return -1;
    }

    void go(int n, int turn)
    {
        if(turn%2 == 0)
        {
            human_owned[human_owned_index++] = n;
            data[n] = 5;
        }
        else
        {
            cpu_owned[cpu_owned_index++] = n;
            data[n] = 3;
        }
    }

    bool move(int turn)
    {
        switch(turn)
        {
            case 1:
                go(0, turn);
```

```
    break;
case 2:
    if(data[4] == 2)
        go(4, turn);
    else
        go(0, turn);
    break;
case 3:
    if(data[8] == 2)
        go(8, turn);
    else
        go(2, turn);
    break;
case 4:
    if(posswin('x') != -1){
        go(posswin('x'), turn);
        return true;
    }
    else
        go(make2(), turn);
    break;
case 5:
    if(posswin('x') != -1){
        go(posswin('x'), turn);
        return true;
    }
    else if(posswin('o') != -1)
        go(posswin('o'), turn);
    else if(data[6] == 2)
        go(6, turn);
    else
        go(2, turn);
    break;
case 6:
    if(posswin('o') != -1){
        go(posswin('o'), turn);
        return true;
    }
    else if(posswin('x') != -1)
        go(posswin('x'), turn);
    else
        go(make2(), turn);
    break;
case 7:
    if(posswin('x') != -1){
        go(posswin('x'), turn);
        return true;
    }
    else if(posswin('o') != -1)
        go(posswin('o'), turn);
    else
        for(int i=0 ; i<9 ; i++)
            if(data[i] == 2)
            {
                go(i, turn);
                break;
            }
    break;
case 8:
    if(posswin('o') != -1){
        go(posswin('o'), turn);
        return true;
    }
    else if(posswin('x') != -1)
        go(posswin('x'), turn);
    else
        for(int i=0 ; i<9 ; i++)
            if(data[i] == 2)
            {
```

```
                            go(i, turn);
                            break;
                        }
                break;
            case 9:
                if(posswin('x') != -1){
                    go(posswin('x'), turn);
                    return true;
                }
                else if(posswin('o') != -1)
                    go(posswin('o'), turn);
                else
                    for(int i=0 ; i<9 ; i++)
                        if(data[i] == 2)
                        {
                            go(i, turn);
                            break;
                        }
                break;
        }
        return false;
    }

    bool valid_move(int pos)
    {
        if(pos < 9 && data[pos] == 2)
            return true;
        else
            return false;
    }

};

int main()
{
    Board b;
    int flag = 0;
    for (int i = 1; i <= 9; ++i)
    {
        cout << "Turn = " << i << endl;
        int pos;
        if(i%2 == 0)
        {
            cout << "HUMAN ? ";
            cin >> pos;
            if(b.valid_move(pos))
                b.go(pos, i);
            else{
                cout << "INVALID MOVE\n";
                return 0;
            }
        }
        else
        {
            if(b.move(i))
            {
                flag = 1;
                break;
            }
        }
        b.display();
        cout << endl;
    }
    if(flag){
        b.display();
        cout << "CPU WINS\n";
    }
    else
        cout << "DRAW\n";
```

```
    return 0;
}

// OUTPUT

// Turn = 1
// |x| | |
// | | | |
// | | | |

// Turn = 2
// HUMAN ? 8
// |x| | |
// | | | |
// | | |o|

// Turn = 3
// |x| |x|
// | | | |
// | | |o|

// Turn = 4
// HUMAN ? 1
// |x|o|x|
// | | | |
// | | |o|

// Turn = 5
// |x|o|x|
// | | | |
// |x| |o|

// Turn = 6
// HUMAN ? 4
// |x|o|x|
// | |o| |
// |x| |o|

// Turn = 7
// |x|o|x|
// |x|o| |
// |x| |o|
// CPU WINS
```