Now we are gonna see how to make a move. We can declare a function for that. Where the return type will be void. And the parameter will be a character. The character is X or O which will be inputed from the user.

1st we will take a number from 1 to 9 for which cell it will be inputed.

```
int choice;
   scanf("%d",&choice);
```

Now we will input in the board.

	Col-0	Col-1	Col-2
Row-0	1	2	3
Row-1	4	5	6
Row-2	7	8	9

If we want the cell 1 2 3 then we have to select the row 0. For row we can make a formula-

Row-(choice-1) / 3;

If we input 1 2 3 we get row-0. If 4 5 6 then row-1. If 7 8 9 then row-2.

Now let's get into the column. For that we can apply the formula-

Col=(choice-1) % 3;

If we input 1 4 7 then we get col-0. If 2 5 8 then col-1. If 3 6 9 then col-2.

```
if(choice>=1 && choice<=9){
      int r=(choice-1)/3;
      int c=(choice-1)%3;



  }else{
      printf("Invalid Input.Choose from 1 to 9\n");
   }
```

Now we want to put character in the cell. We have to check if there are any character inside this. As it is a continues game meaning move after move so we have to check it if it already filled or not.

```
if(choice>=1 && choice<=9){
      int r=(choice-1)/3;
      int c=(choice-1)%3;
      if(board[r][c]!='X' && board[r][c]!='O'){
         board[r][c]=play;
      }else{
         printf("The cell is already taken. Try again!\n");
      }
```

```
    }else{
        printf("Invalid Input.Choose from 1 to 9\n");
    }
```

But see that – suppose we input in the wrong cell so it will show the error message. But there need a way to again make the move if the move is wrong. That's why we can enter the bool variable which will check if it is wrong or not. If it is right then only it will take input.

```
void make_move(char play){
    int choice; bool valid=false;
    while(!valid){
    scanf("%d",&choice);
    if(choice>=1 && choice<=9){
        int r=(choice-1)/3;
        int c=(choice-1)%3;
        if(board[r][c]!='X' && board[r][c]!='O'){
            board[r][c]=play;
            valid=true;
        }else{
            printf("The cell is already taken. Try again!\n");
        }
    }else{
        printf("Invalid Input.Choose from 1 to 9\n");
    }
    }
}
```

If we include the full program. Then-

```
char board[3][3]={'1','2','3','4','5','6','7','8','9'};
void display_board(){
    for(int i=0; i<3; i++){
        for(int j=0; j<3; j++){
            printf(" %c ", board[i][j]); // There are one space beside each side of %c
            if(j<2){ printf("|");}
        }
        printf("\n");
        if(i<2){printf("---|---|---\n"); }
    }
}
void make_move(char play){
    int choice; bool valid=false;
    while(!valid){
    scanf("%d",&choice);
    if(choice>=1 && choice<=9){
        int r=(choice-1)/3;
        int c=(choice-1)%3;
        if(board[r][c]!='X' && board[r][c]!='O'){
```

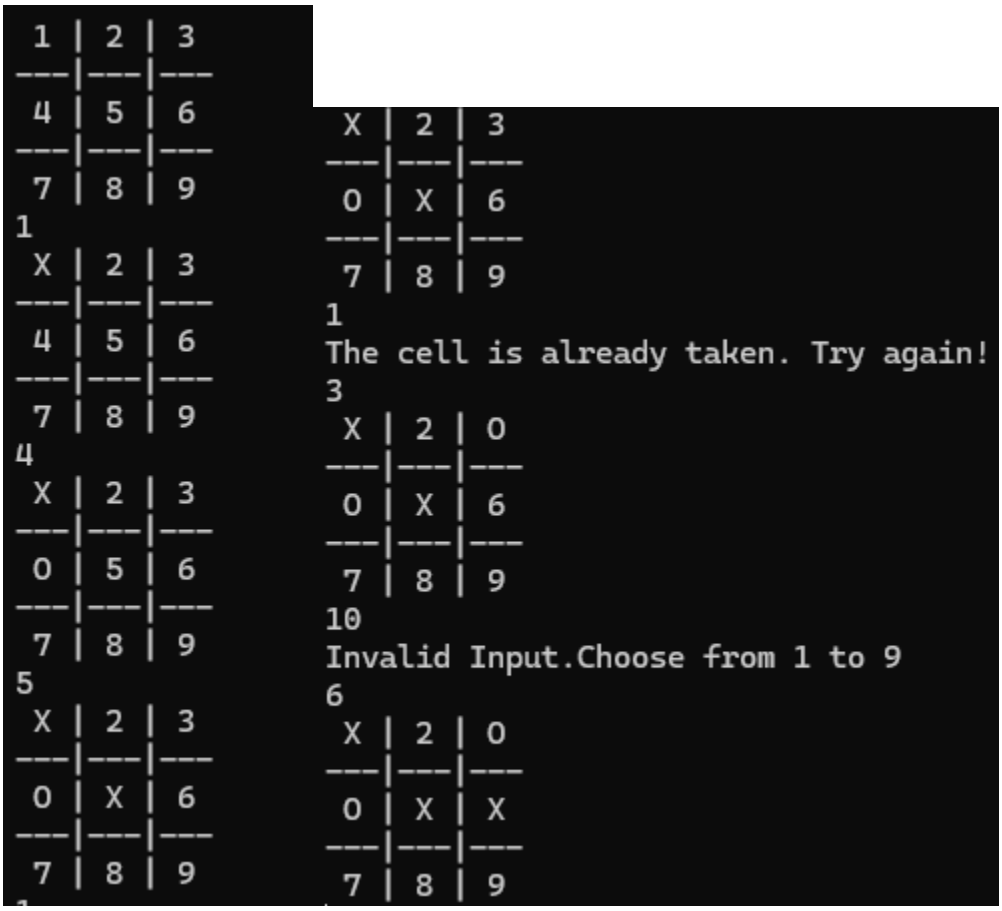
```
            board[r][c]=play;
            valid=true;
        }else{
            printf("The cell is already taken. Try again!\n");
        }
    }else{
        printf("Invalid Input.Choose from 1 to 9\n");
    }
    }
}
```

Let’s see that the program can work out fine or not.

```
#include<stdio.h>
#include<stdbool.h>
char board[3][3]={'1','2','3','4','5','6','7','8','9'};
void display_board(){
    for(int i=0; i<3; i++){
        for(int j=0; j<3; j++){
            printf(" %c ", board[i][j]); // There are one space beside each side of %c
            if(j<2){ printf("|");}
        }
        printf("\n");
        if(i<2){printf("---|---|---\n"); }
    }
}
void make_move(char play){
    int choice; bool valid=false;
    while(!valid){
    scanf("%d",&choice);
    if(choice>=1 && choice<=9){
        int r=(choice-1)/3;
        int c=(choice-1)%3;
        if(board[r][c]!='X' && board[r][c]!='O'){
            board[r][c]=play;
            valid=true;
        }else{
            printf("The cell is already taken. Try again!\n");
        }
    }else{
        printf("Invalid Input.Choose from 1 to 9\n");
    }
    }
}
int main(){
```

```
    display_board();
    int i=0;
    while(5){
        if(i%2==0){
            make_move('X');
            display_board();
        }else{
            make_move('O');
            display_board();
        }
        i++;
    }
    return 0;
}
```

Output:



We see that the program will work just fine. It is working as our wish.