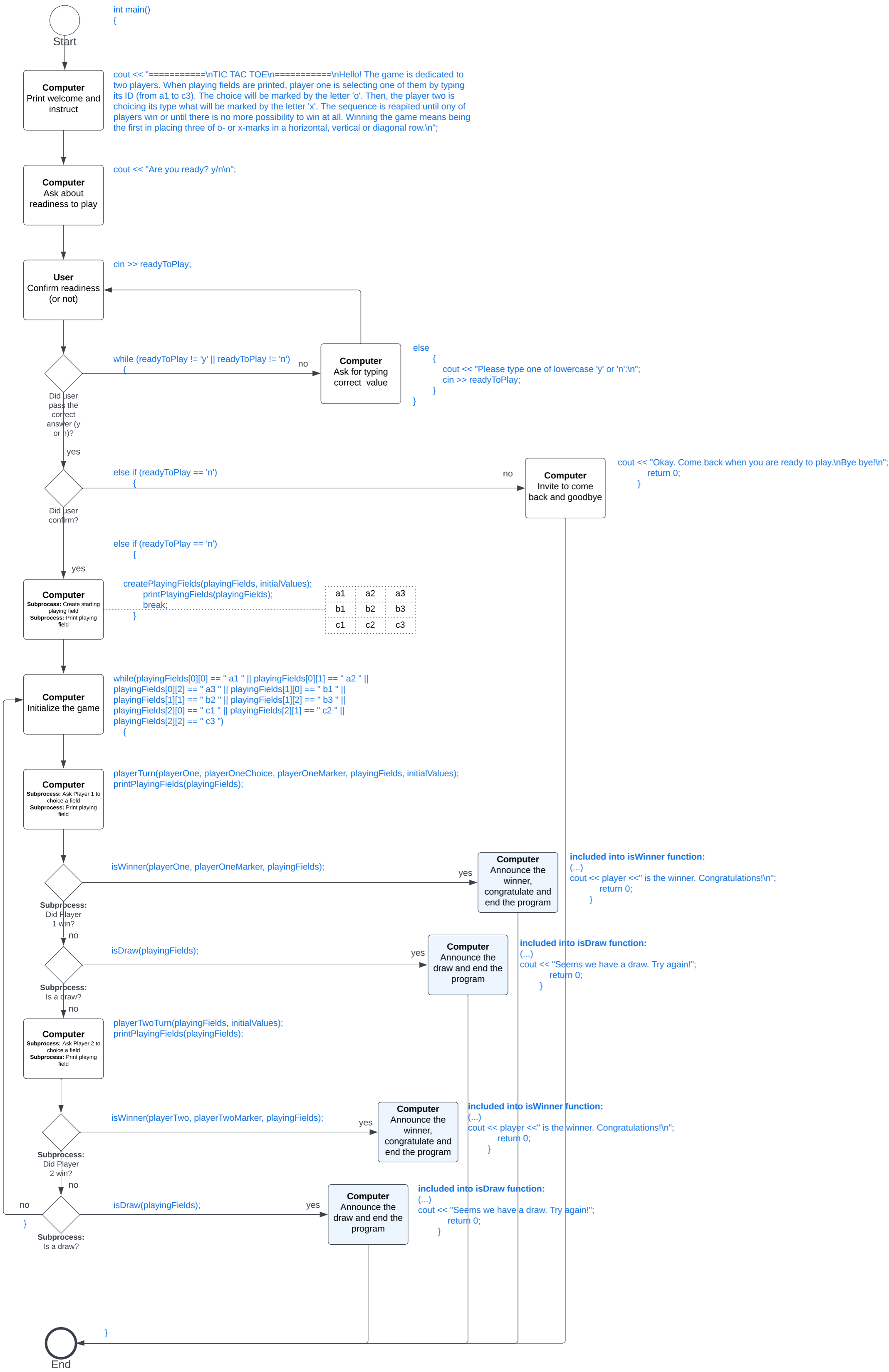


FUNCTIONS

int main()



VARIABLES

Global scope

```
string initialValues[] = {" a1 ", " a2 ", " a3 ", " b1 ", " b2 ", " b3 ", " c1 ", " c2 ", " c3 "};
string playingFields[3][3];
string playerOne = "Player 1";
string playerTwo = "Player 2";
string playerOneChoice;
string playerTwoChoice;
string playerOneMarker = " o ";
string playerTwoMarker = " x ";
char readyToPlay;
```

Local scope

```
bool inRange = false;
counters:
• int i;
• int x;
• int y;
```

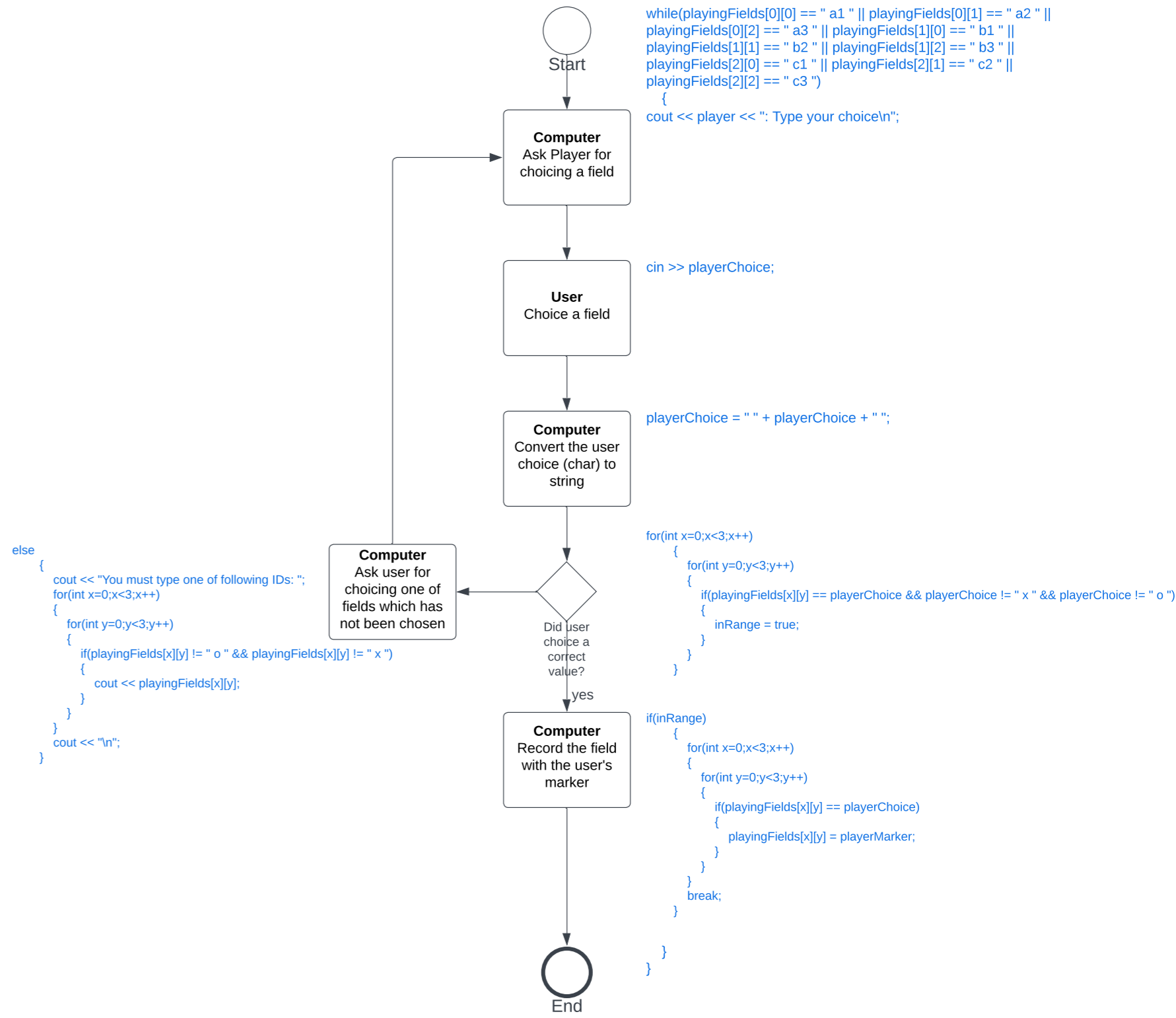
void createPlayingFields(string playingFields[3][3], string initialValues[])

```
void createPlayingFields(string playingFields[3][3], string initialValues[])
{
    int i = 0;
    for(int x=0;x<3;x++)
    {
        for(int y=0;y<3;y++)
        {
            playingFields[x][y] = initialValues[i];
            i++;
        }
    }
}
```

void printPlayingFields(string playingFields[3][3])

```
void printPlayingFields(string playingFields[3][3])
{
    for(int x=0;x<3;x++)
    {
        for(int y=0;y<3;y++)
        {
            cout<<playingFields[x][y];
        }
        cout << endl;
    }
}
```

void playerTurn(string player, string playerChoice, string playerMarker, string playingFields[3][3], string initialValues[])



void isWinner(string player, string playerMarker, string playingFields[3][3])

```
{
    if((playingFields[0][0] == playerMarker && playingFields[0][1] == playerMarker && playingFields[0][2] == playerMarker) ||
    (playingFields[1][0] == playerMarker && playingFields[1][1] == playerMarker && playingFields[1][2] == playerMarker) ||
    (playingFields[2][0] == playerMarker && playingFields[2][1] == playerMarker && playingFields[2][2] == playerMarker) ||
    (playingFields[0][0] == playerMarker && playingFields[1][1] == playerMarker && playingFields[2][2] == playerMarker) ||
    (playingFields[2][0] == playerMarker && playingFields[1][1] == playerMarker && playingFields[0][2] == playerMarker))
    {
        cout << player << " is the winner. Congratulations!\n";
        exit(0);
    }
}
```

void isDraw (string playingFields[3][3])

```
{
    if(playingFields[0][0] != " a1 " && playingFields[0][1] != " a2 " &&
    playingFields[0][2] != " a3 " && playingFields[1][0] != " b1 " &&
    playingFields[1][1] != " b2 " && playingFields[1][2] != " b3 " &&
    playingFields[2][0] != " c1 " && playingFields[2][1] != " c2 " &&
    playingFields[2][2] != " c3 ")
    {
        cout << "Seems we have a draw. Try again!";
        exit(0);
    }
}
```