

# REPORT ASSIGNMENT

Name: Nguyễn Đức Huy

Class: CC01

## 1. *Beginning*

In the beginning of the game, we will print the rule of the game for two players to read. If the players understand the rule, press any key to continue.

---

```
WELCOME TO GAME FOUR IN ROW!

Here is the rule of the game
In the first move, you must drop the piece in the center column of the board
After first move, you have 3 times to undo the move(before your opponent's turn)
You can block the next opponent's move 1 time. But you can not use it when your opponent have opportunity to win.
If you want to remove one arbitrary piece of the opponent, press R(1 time)
If you place a piece at an inappropriate column, it will be count as a violation
If you violate over 3 times, you will lose the game
If you have already understood the rule of the game, press any key to continue
```

After reading the rule, game will require two players to enter their names.

We store name of each player to \$s1 and \$s2 so we can use it later.

```
Now, let's play the game
Player1, please enter your name(not exceed 20 chars): HUY
Player2, please enter your name(not exceed 20 chars): TAI
```

After entering their name, game will assign piece randomly for each player.

We use syscall 42 to random choose between 0 and 1.

If the result is 0, we will assign the player 1 X and the player 2 O.

Else if the result is 1, we will assign the player 1 O and the player 2 X.

```
The player 1: X
The player 2: O
```

## 2. Game

In each move, the console will print the number of remaining undo, violation and the name of the player.

We use list violation and list undo to store the number of violation and undo times of each player.

For example, in the beginning:

List violation = (3, 3) (each player has 3 times remaining to violate)

We store an array with 42 '\*' to print the board.

First, we will output array[0] -> array[6]. Then create a new line.

Then continue output array[7] -> array[13]. Then create a new line.

The same until array[41].

```
The number of remaining violation: 3
The number of remaining undo: 3
```

	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	*

```
HUY
0 1 2 3 4 5 6
It's your turn, please enter your instruction:
```

In the first move, player must place the piece in the center of the column of the board. If you enter other instructions, you will be required to enter the instruction again.

We will load the char you input and compare it with '3'. If it's different, we will require you to input again.

The number of remaining violation: 3  
The number of remaining undo: 3

```
| * * * * * |  
| * * * * * |  
| * * * * * |  
| * * * * * |  
| * * * * * |  
| * * * * * |  
0 1 2 3 4 5 6
```

HUY  
It's your turn, please enter your instruction: 8 → Not middle column  
You must drop the piece in the center of column in the first move  
HUY  
It's your turn, please enter your instruction: ← Enter the instruction again

After first move, the player can enter instruction with 0, 1, 2, 3, 4, 5, 6, R.

If you enter other instructions, it will be count as violation and game will require you to enter the instruction again.

*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	○	*	*	*
*	*	*	X	*	*	*
0	1	2	3	4	5	6

It's your turn, please enter your instruction: 8

The instruction is inappropriate. Please enter the instruction again

It's your turn, please enter your instruction:

**Enter column:**

We will assign  $j = 4$ . And we will check from  $t = 35 + j$ .

If `array[t] = '*'` we will assign piece of this player to this position.

Else reduce  $t = t - 7$  until find black position or  $t < 0$ .

If  $t < 0$ , your instruction will be count as violation.

## Undo:

If you enter appropriate instruction, the console will print the board again and ask you want to undo the last instruction or not. If yes, press Y. If no, press any other keys to continue.

If you run out of undo times and you still press Y, it will be count as a violation.

Case when you press Y when you still have undo times:

It's your turn, please enter your instruction: 4

The new piece is inserted

A 6x7 grid of stars. The stars are arranged in 6 rows and 7 columns. The grid is as follows:

*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*

An arrow points to the star at row 5, column 4.

The number of remaining violation: 3

```
The number of remaining undo: 2
```

Press Y, it will undo the last instruction

A 6x6 grid of stars. The stars are arranged in 6 rows and 6 columns. The star at row 4, column 5 is highlighted with a blue arrow. The star at row 4, column 4 is replaced by an 'O', and the star at row 4, column 3 is replaced by an 'X'.

It's your turn, please enter your instruction: 3

Then you can enter again the instruction

**Block:**

The idea I used in block is to assign each blank piece('\*') with opponent's piece. After that, use function to check victory. But carefully that it must have the piece below that piece so it must be count as an opportunity to win. For example:

It's your turn, please enter your instruction: 3

Assign X into this piece and check victory

*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	X	*	*	*	*
*	O	X	O	*	*	*
*	X	X	X	O	*	*
*	O	O	O	X	O	*
0	1	2	3	4	5	6

It has a piece below

It's an opportunity to win

Do you want to block the next opponent's move (press B if you want)

You can't block the next opponent's move because the opponent have op

```
The number of remaining undo: 3
```

Assign X into this piece and check victory

The piece below is blank

It's your turn, please enter your instruction: 1

It's not an opportunity to win so you can block

If you don't undo the last instruction, the player will be asked to want block the next opponent's move or not. If yes, press B. If no, press any other keys to continue.

If you run out of block times and you still press B, it will be count as a violation.

Case when you press B when you still have block times:

The opponent will be blocked and you will be continued to play.

It's still your turn

```

HUY
It's your turn, please enter your instruction: 3

      | * * * * * |
      | * * * * * |
      | * * * * * |
      | * * X * * |
      | * * O * * |
      | * * X * * |
      | 0 1 2 3 4 5 6 |

Do you want to undo the last instruction (press Y if you want): o
Do you want to block the next opponent's move (press B if you want): B
The number of remaining violation: 3
The number of remaining undo: 2

      | * * * * * |
      | * * * * * |
      | * * * * * |
      | * * X * * |
      | * * O * * |
      | * * X * * |
      | 0 1 2 3 4 5 6 |

HUY
It's your turn, please enter your instruction: R

```

Press B to block next opponent's move

### Enter remove:

When you want to remove, we will ask you enter the row and the column of piece you want to remove. Let's assign  $i$  is the row of the piece you want to remove and  $j$  is the column of the piece you want to remove.

I can get the position of the piece in the array with formula  $7*i + j$ .

We will continue to check if this position is opponent's piece or not.

If it is, you can remove the piece.

Else, it will be count as a violation and you need to enter the row and the column of the piece you want to remove again.

About drop down the pieces above, with  $t = 7*i + j$ .

Assign  $\text{array}[t] = \text{array}[t-7]$  then  $t = t - 7$ .

Continue like above until  $t < 0$ .

If you enter R instruction, you will be asked about the row and column of the piece you want to remove.

If you enter the appropriate piece, the piece will be removed and any pieces above this piece will fall down.

The number of remaining violation: 3  
The number of remaining undo: 2

*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	X	*	*	*
*	*	*	O	*	*	*
*	*	*	X	*	*	*
0	1	2	3	4	5	6

HUY

It's your turn, please enter your instruction: R

The row of piece you want to remove(0->5) is: 4

The column of piece you want to remove(0->6) is: 3

The number of remaining violation: 3

The number of remaining undo: 3

The piece you want to remove is placed at (4,3)

After removing, the pieces above will drop down

*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	X	*	*	*
*	*	*	X	*	*	*
0	1	2	3	4	5	6

TAI

It's your turn, please enter your instruction:

If you enter inappropriate piece( '\*' or your piece), it will be count as a violation. You will be required to enter again.

The number of remaining violation: 3  
The number of remaining undo: 3

*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	X	*	*	*
*	*	*	X	*	*	*
0	1	2	3	4	5	6

Inappropriate piece

TAI

It's your turn, please enter your instruction: R

The row of piece you want to remove(0->5) is: 4

The column of piece you want to remove(0->6) is: 5

The piece you want to remove is not the piece of your opponent.

Please enter the instruction again

The row of piece you want to remove(0->5) is:

The piece you want to remove is placed at (4,5)

Enter again

### 3. End

When you win, the console will print your name, number of pieces you used and the piece you used.

```

| * * * * * * |
| * * * * * * |
| * * * * * * |
| * * * * * * |
| * * * * * * |
| * * * * * * |
| X X X X O O * |
| 0 1 2 3 4 5 6 |

```

```

Congratulation! Winner HUY
Number of piece: 4
Piece char X

```

When we check the win with row.

Firstly, I divide checking with each row.

With row 0, variable count to count the number of consecutive similar pieces.

If piece is different from black piece, count = 1.

Else if piece is blank, count = 0.

Else if piece is different from previous piece and not black piece, count = 1.

Else if it's the same as previous piece, count++.

When count == 4, jump out and print win game.

Continue with other rows like row 0.

The idea was the same for column. Just different from the way to access the piece with the same column, use +7

The idea was also the same for diagonal. But to create the begin and the end of the diagonal to check.

```

InDiagonal1: .word 14, 7, 0, 1, 2, 3
OutDiagonal1: .word 38, 39, 40, 41, 34, 27
InDiagonal2: .word 3, 4, 5, 6, 13, 20
OutDiagonal2: .word 21, 28, 35, 36, 37, 38

```

With diagonal from left to right, use +8 to access the next piece on the same diagonal.

With diagonal from right to left, use +6 to access the next piece on the same the diagonal.

When tie game, the console only print the board.

```
| x x o o x o o |  
| o o x x o x x |  
| x x o o x o o |  
| o o x x o x x |  
| x x o o x o o |  
| o o x x o x x |  
0 1 2 3 4 5 6
```

```
-- program is finished running (dropped off bottom) --
```

To check tie, we easily check the number of row 0 ( from array[0] -> array[6]).

If the number of pieces is 7, it's tie game.