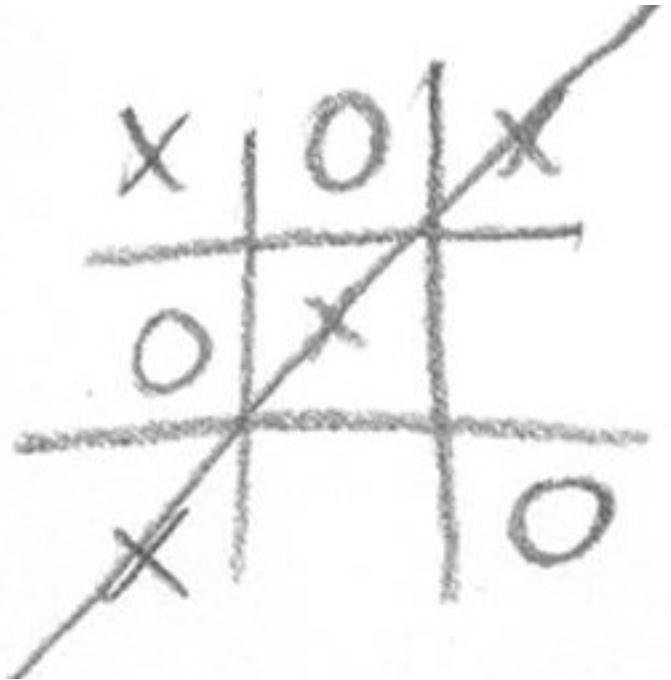


tic-tac-toe

컴프2에서 다룬 문제로 <server 숙제>로 올려보자

tic-tac-toe

- 우리가 알고 있는 오목(5목) 이전에는 3목이 있었다. 외국에서는 이 게임을 '틱택토(Tic-Tac-Toe)'라 부르는데, 쉽고 재미있어서 유아들을 위한 게임으로 알려져 있다. 틱택토는 3X3칸을 가진 게임판을 만들고, 각자 동그라미 심볼(O)와 가위표 심볼(X)을 고른 뒤 번갈아 가며 심볼을 그리는 게임이다. 가로, 세로, 대각선 상관 없이 직선으로 자기 심볼 3개를 먼저 만들면 승리한다.



Lab(tic-tac-toe 준비)

- Player X와 Player O가 차례로 3x3칸으로 된 게임판의 칸을 선택하는 것을 구현한다.
 - 실행예는 오른 편과 같다.
 - 함수화
 - 각 선택이 표시된(X와 O로) 게임판을 출력하는 부분을 함수화하라
 - void display(char b[][3])
 - 그 외 다른 부분은 함수화해도 좋다(물론 안 해도 좋다. 즉 이 경우는 main함수와 display함수로 프로그램이 구성된다)
 - 위의 선택은 판이 모두 완성될 때까지 계속된다고 하자.

```
  0 1 2
  ----
0 |
1 |
2 |
Player X<행 열>:0 0
  0 1 2
  ----
0 | X
1 |
2 |
Player O<행 열>:1 0
  0 1 2
  ----
0 | X
1 | O
2 |
Player X<행 열>:0 1
  0 1 2
  ----
0 | X X
1 | O
2 |
Player O<행 열>:1 1
  0 1 2
  ----
0 | X X
1 | O O
2 |
Player X<행 열>:0 2
  0 1 2
  ----
0 | X X X
1 | O O
2 |
Player O<행 열>:2 2
  0 1 2
  ----
0 | X X X
1 | O O
2 | O
Player X<행 열>:_
```

HW(tic-tac-toe)

■ Tic-tac-toe를 구현하자.

- tic-tac-toe문제를 어떻게 나눌 것인가?를 생각하고 각 부분을 함수화하여 단계적으로 문제를 풀어본다.
 - 예: 그림판 출력(display)/승리여부체크/...
- 충분한 test case를 실행시켜 보면서 프로그램을 테스트한다
 - 예: 가로 완성, 세로 완성, 대각선 완성...
- 그림판을 완성시켰는데 승부가 나지 않으면 Nobody wins!를 출력한다.

```

      0 1 2
      -----
0 :
1 :
2 :
Player X<행 열>:1 1
      0 1 2
      -----
0 :
1 :  X
2 :
Player O<행 열>:0 0
      0 1 2
      -----
0 : 0
1 :  X
2 :
Player X<행 열>:0 1
      0 1 2
      -----
0 : 0 X
1 :  X
2 :
Player O<행 열>:1 0
      0 1 2
      -----
0 : 0 X
1 : 0 X
2 :
Player X<행 열>:2 1
      0 1 2
      -----
0 : 0 X
1 : 0 X
2 :  X
Player X wins!
```

- 즉, 결과는 아래의 세 가지 중의 하나이다(서버 테스트의 경우 정확한 문구가 출력되어야 결과가 맞는 것으로 간주됨에 유의할 것)
 - Player X wins!
 - Player O wins!
 - Nobody wins!
- 이미 고른 칸을 선택하면 다시 입력 프롬프트를 보여준다(실행 예 참조)

```
C:\windows\system32\cmd.exe

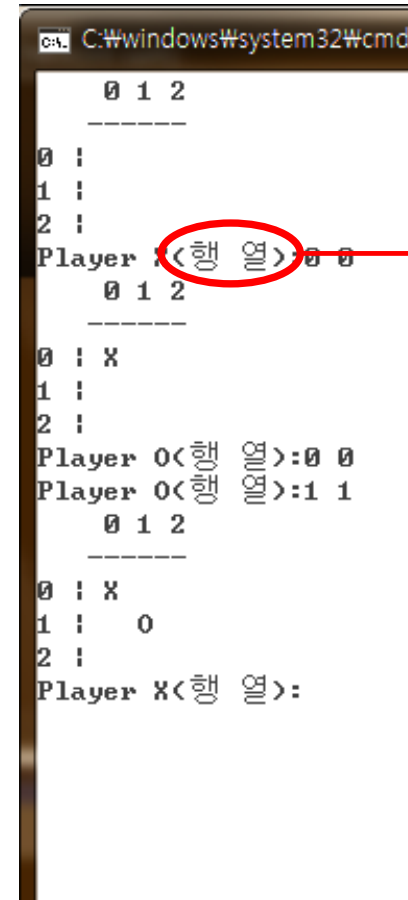
  0 1 2
  ---
0 |
1 |
2 |
Player X<행 열>:0 0
    0 1 2
    ---
0 | X
1 |
2 |
Player O<행 열>:0 0
Player O<행 열>:1 1
    0 1 2
    ---
0 | X
1 | 0
2 |
Player X<행 열>:
```

CCD 테스트 예정 (Code-Clone Detector)

- 다른 학생의 코드를 절대 복사하지 않는다.
 - 분반이 다르다고 안심말자.
 - CCD 테스트는 전체 분반을 대상으로 한다는 것을 잊지 말자.
- 작년과 마찬가지로 테스트 예정임.

CPD 테스트 예정 (Copy-and-Paste Detector)

- 올해 추가된 새로운 테스트 (자기 코드 내의 Copy & Paste)
- Code의 중복이 있다면 loop 혹은 함수를 사용하여 가능한 줄이도록 한다.
- 이를 테스트할 예정이니 코드 제출 시에 다음에 유의한다.
 - Code에 한글이 없어야 한다. (주석 포함)



```
C:\Windows\system32\cmd.exe
  0 1 2
  -----
0 |
1 |
2 |
Player X <행 열>: 0 0
  0 1 2
  -----
0 | X
1 |
2 |
Player O <행 열>: 0 0
Player O <행 열>: 1 1
  0 1 2
  -----
0 | X
1 | 0
2 |
Player X <행 열>:
```

row column 으로 변경

Example of CP (From One of Current Submissions)

```
int _main(void)
{
    char board[3][3];
    char win;
    int i;

    init_board(board);
    print_board(board);
```

```
do
{
```

```
    player_move(board, 'X');
win = get_winner(board);
if(win == 'O' || win == 'X')
    break;
if(check_draw(board) == 1)
    break;
```

```
    player_move(board, 'O');
win = get_winner(board);
if(win == 'O' || win == 'X')
    break;
if(check_draw(board) == 1)
    break;
} while(win == ' ');
```

```
if(win == 'O')
    printf("Player O wins!\n");
else if(win == 'X')
    printf("Player X wins!\n");
else
    printf("Nobody wins!\n");
```

```
return 0;
```

```
}
```

Copy & Paste