

# 실전 개발형 코딩 테스트 문제 풀이 ②

## 문제 설명

문제 설명 | 문제를 바르게 이해하기

강사 나동빈

# 실전 개발형 코딩 테스트 문제 풀이 ②

문제 설명

개발형 코딩 테스트  
문제 설명

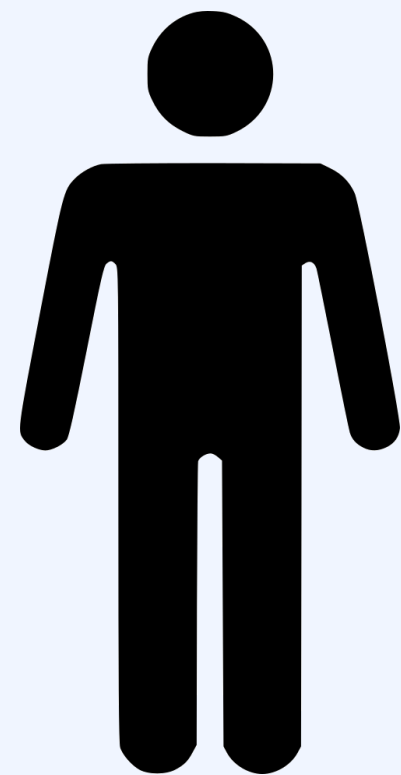
## 문제 살펴보기

개발형  
코딩 테스트  
문제 설명

문제 제목: Tic Tac Toe 게임

문제 설명

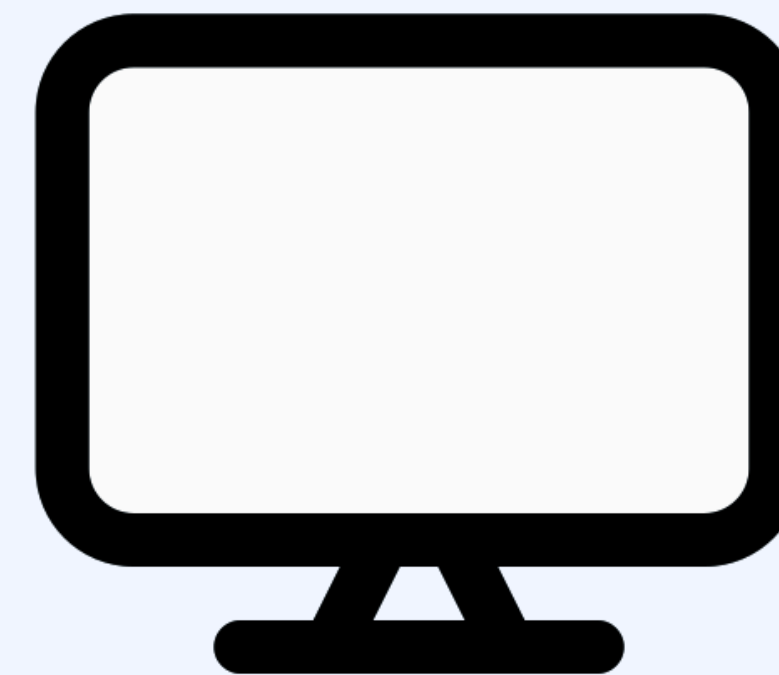
- **컴퓨터**와 **참가자**가 틱택토(Tic Tac Toe) 게임을 진행한다.
- 컴퓨터와 참가자가 번갈아 가면서 돌(O 혹은 X)을 두어 경쟁한다.



참가자

O	X	O
X	O	O
X	O	X

틱택토(Tic Tac Toe)



컴퓨터

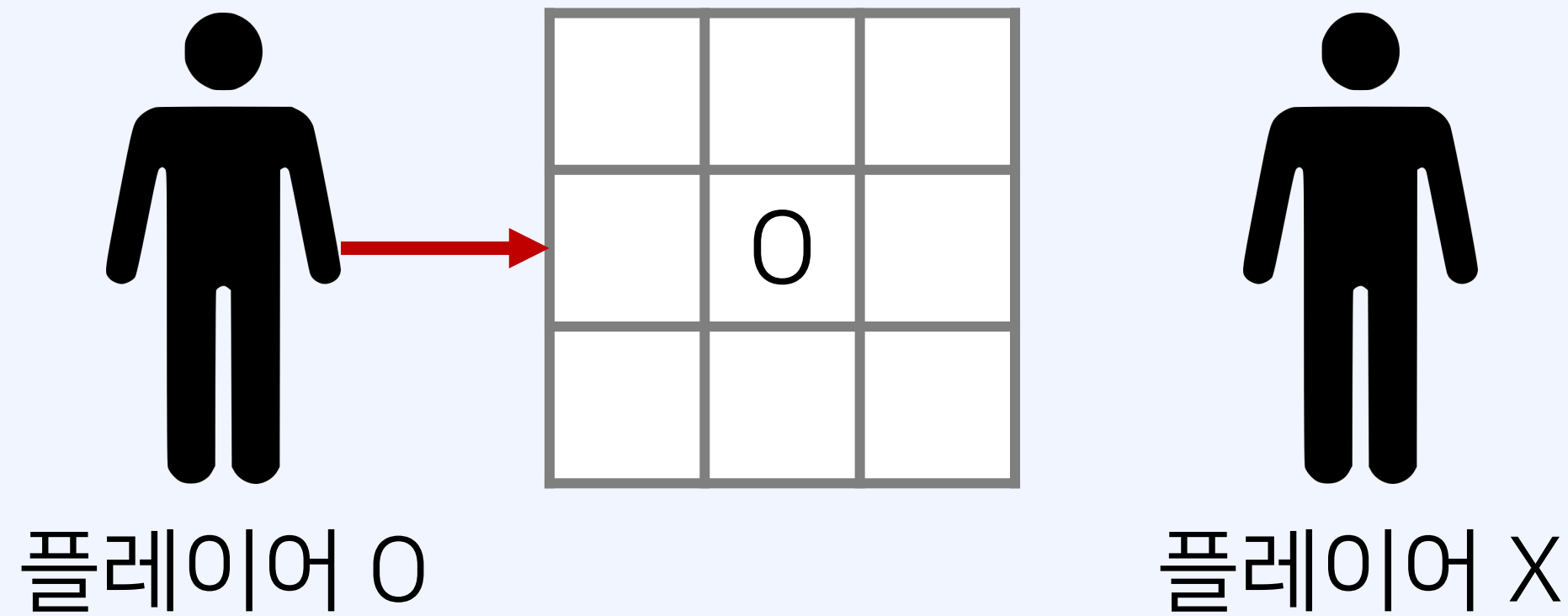
개발형 코딩 테스트  
문제 설명

문제 살펴보기

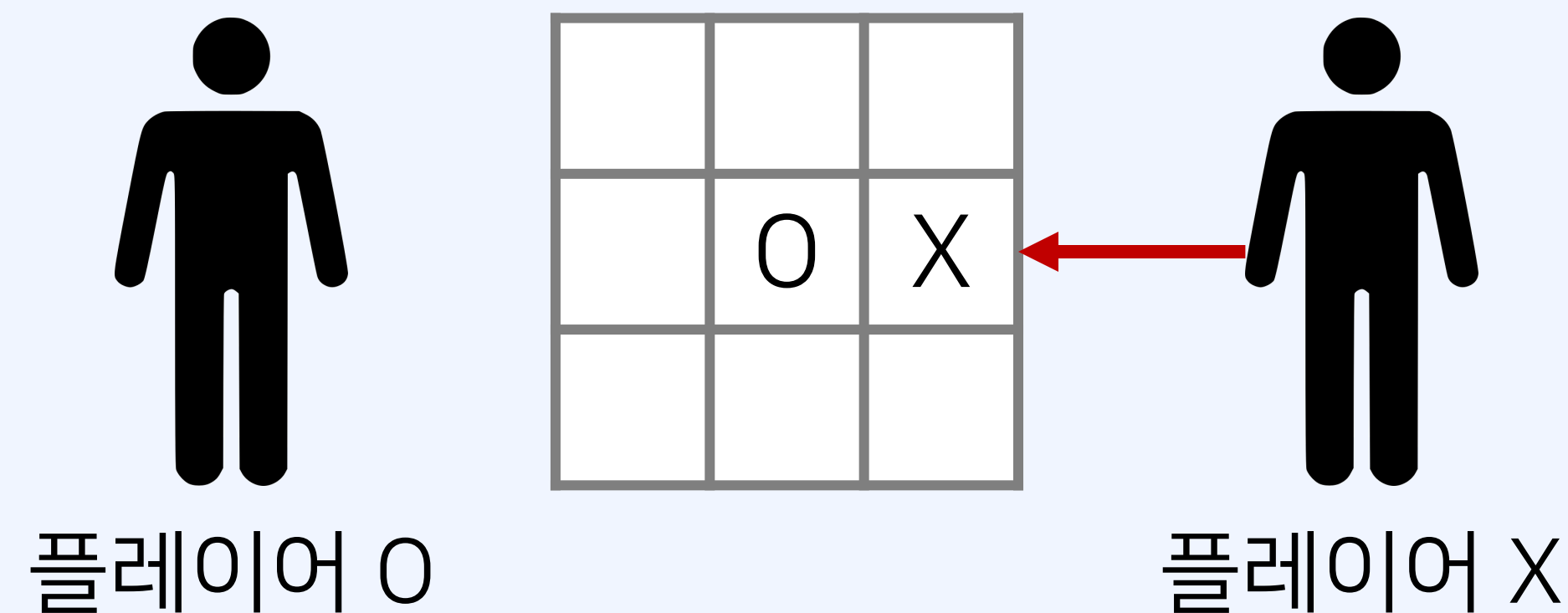
개발형  
코딩 테스트  
문제 설명

- 틱택토(Tic Tac Toe)는 먼저 3목을 만들면 이기는 게임이다.
- 플레이어 O부터 시작하여 플레이어 X와 번갈아 가며 돌을 둔다.

차례 ①



차례 ②



개발형 코딩 테스트  
문제 설명

문제 살펴보기

개발형  
코딩 테스트  
문제 설명

틱택토(Tic Tac Toe) 게임의 승리 판정

- 하나의 **행**, 하나의 **열**, 혹은 하나의 **대각선**을 차지하는 경우 즉시 승리한다.

0	X	0
X	0	0
X	0	X

무승부

0	X	X
0	0	0
X	0	X

플레이어 0 승리

0	0	X
0	X	
X	0	X

플레이어 X 승리

0	X	0
X	0	
X		0

플레이어 0 승리

0	X	0
0	X	
	X	

플레이어 X 승리

X	X	0
0	0	X
X	0	0

무승부

개발형 코딩 테스트  
문제 설명

## 문제 살펴보기

개발형  
코딩 테스트  
문제 설명

문제 제목: Tic Tac Toe 게임

문제 요구 사항

- 컴퓨터와 경쟁하여 Tic Tac Toe 게임에서 승리하거나 비기는 것이 목표다.
- 비기는 경우는 9개의 공간에 모두 돌이 놓여서, 더 돌을 놓을 곳이 없는 경우를 말한다.
- 중간에 한 번도 패배하지 않고, 연속으로 30번 이상 승리하거나 비기면 FLAG를 얻게 된다.
- 구체적으로 “최근에 **끝난** 30개의 게임에 대해서” 한 번도 패배하지 않으면 된다.
- 누가 먼저 돌을 놓을지는 컴퓨터(서버)가 랜덤으로 설정한다.

## 개발형 코딩 테스트 문제 설명

## Start API

## 개발형 코딩 테스트 문제 설명

- **한 번의 Tic Tac Toe 게임을 시작**하기 위한 auth key를 발급하는 API다.
- Start API를 실행하면, 새롭게 틱택토(Tic Tac Toe)게임이 시작된다.
- 실제 개발형 코딩 테스트에서는 사람마다 다른 Access-Token을 부여한다.
- 본 문제에서는 간단히 ABCDEFGH12345678를 부여 받았다고 가정하자.

### [요청(request) 양식]

GET /start

Access-Token: **[자신의 Access-Token]**

Content-Type: application/json

Access-Token	코딩 테스트에 참가자에게 부여되는 고유한 식별 토큰 값
--------------	--------------------------------

개발형 코딩 테스트  
문제 설명

Start API

개발형  
코딩 테스트  
문제 설명

- 참가자의 Access-Token이 정상적이라면, 다음과 같이 key가 발급된다.
- 컴퓨터가 먼저 두는 경우에는 돌을 둔 위치도 반환된다. 위치는 "(0,0)"부터 "(2,2)"까지다.

[응답(response) 양식]

```
{
  "status": "success",
  "auth_key": [발급된 key],
  "position": "(1,1)"
}
```

auth_key	Start API를 통해 부여받은 key로, 이후 Tic Tac Toe 게임을 진행할 때 부여받은 key를 사용
position	컴퓨터가 돌을 둔 위치로, (0,0)부터 (2,2)까지의 값 혹은 "none" 값을 가짐



개발형 코딩 테스트  
문제 설명

Start API

개발형  
코딩 테스트  
문제 설명

- 참가자의 Access-Token이 정상적이라면, 다음과 같이 key가 발급된다.
- 플레이어(참가자)가 먼저 두어야 하는 경우에는 위치 값으로 "none"이 반환된다.

[응답(response) 양식]

```
{
  "status": "success",
  "auth_key": [발급된 key],
  "position": "none"
}
```

auth_key	Start API를 통해 부여받은 key로, 이후 Tic Tac Toe 게임을 진행할 때 부여받은 key를 사용
position	컴퓨터가 돌을 둔 위치로, (0,0)부터 (2,2)까지의 값 혹은 "none" 값을 가짐

## 개발형 코딩 테스트 문제 설명

## Start API

## 개발형 코딩 테스트 문제 설명

- 참가자의 Access-Token이 정상적이지 않다면, 다음과 같이 응답한다.

[응답(response) 양식]

```
{  
  "status": "unknown candidate"  
}
```

status	사용자의 요청(request)에 대한 처리 상태
--------	----------------------------

개발형 코딩 테스트  
문제 설명

Query API

개발형  
코딩 테스트  
문제 설명

- 참가자로부터 돌을 놓을 위치(position)을 제출 받아, 돌을 놓도록 해준다.
- 위치는 "(0,0)", "(0,1)", "(0,2)", "(1,0)", "(1,1)", "(1,2)", "(2,0)", "(2,1)", "(2,2)" 중 하나여야 한다.

[요청(request) 양식]

```
POST /query
Access-Token: [자신의 Access-Token]
Content-Type: application/json
Payload: {
  "auth_key": [발급받은 key],
  "position": [참가자가 돌을 놓을 위치]
}
```

position

참가자가 두고자 하는 돌의 위치

## 개발형 코딩 테스트 문제 설명

## Query API

## 개발형 코딩 테스트 문제 설명

- 참가자 두 돌로 인해 참가자가 승리했다면 다음과 같이 응답한다.
- 또한 해당 참가자의 연속 승리 혹은 무승부 횟수를 1만큼 증가시킨다.

[응답(response) 양식]

```
{  
  "status": "success",  
  "result": "player win"  
}
```

result	현재 놓인 돌에 대한 처리 결과
--------	-------------------

## 개발형 코딩 테스트 문제 설명

## Query API

## 개발형 코딩 테스트 문제 설명

- 참가자 두 돌로 인해 게임에서 비겼다면 다음과 같이 응답한다.
- 또한 해당 참가자의 연속 승리 혹은 무승부 횟수를 1만큼 증가시킨다.

[응답(response) 양식]

```
{  
  "status": "success",  
  "result": "draw"  
}
```

result	현재 놓인 돌에 대한 처리 결과
--------	-------------------

## 개발형 코딩 테스트 문제 설명

## Query API

## 개발형 코딩 테스트 문제 설명

- 참가자 돌을 둔 뒤에, 게임이 끝나지 않았다면 컴퓨터가 돌을 둔다.
- 컴퓨터가 둔 돌로 인해서도 게임이 끝나지 않았다면, 다음과 같이 돌을 놓은 위치를 응답한다.

[응답(response) 양식]

```
{  
  "status": "success",  
  "position": "(1,1)"  
}
```

position	컴퓨터가 돌을 둔 위치로, (0,0)부터 (2,2)까지의 값을 가짐
----------	---------------------------------------

## 개발형 코딩 테스트 문제 설명

## Query API

## 개발형 코딩 테스트 문제 설명

- 참가자 돌을 둔 뒤에, 게임이 끝나지 않았다면 컴퓨터가 돌을 둔다.
- 컴퓨터가 둔 돌로 인해 컴퓨터가 승리했다면 다음과 같이 응답한다.
- 또한 해당 참가자의 연속 승리 혹은 무승부 횟수를 0으로 초기화한다.

### [응답(response) 양식]

```
{  
  "status": "success",  
  "result": "computer win",  
  "position": "(1,1)"  
}
```

result	현재 놓인 돌에 대한 처리 결과
--------	-------------------

개발형 코딩 테스트  
문제 설명

Query API

개발형  
코딩 테스트  
문제 설명

- 참가자 돌을 둔 뒤에, 게임이 끝나지 않았다면 컴퓨터가 돌을 둔다.
- 컴퓨터가 둔 돌로 인해 게임에서 비겼다면 다음과 같이 응답한다.
- 또한 해당 참가자의 연속 승리 혹은 무승부 횟수를 1만큼 증가시킨다.

[응답(response) 양식]

```
{
  "status": "success",
  "result": "draw",
  "position": "(1,1)"
}
```

result

현재 놓인 돌에 대한 처리 결과



개발형 코딩 테스트  
문제 설명

Query API

개발형  
코딩 테스트  
문제 설명

- 참가자가 연속으로 30번 이상 승리하거나 비기면 FLAG를 얻게 된다.
- 마지막에 컴퓨터가 두어서 끝나는 경우에는 "position" 속성이 포함된다.

[응답(response) 양식]

```
{
  "status": "success",
  "result": [문제의 정답 FLAG],
  "position": "(1,1)"
}
```

result	현재 놓인 돌에 대한 처리 결과
position	(옵션) 컴퓨터가 돌을 놓은 위치

## 개발형 코딩 테스트 문제 설명

## Query API

## 개발형 코딩 테스트 문제 설명

- 참가자의 Access-Token이 정상적이지 않다면, 다음과 같이 응답한다.

[응답(response) 양식]

```
{  
  "status": "unknown candidate"  
}
```

status	사용자의 요청(request)에 대한 처리 상태
--------	----------------------------

## 개발형 코딩 테스트 문제 설명

## Query API

## 개발형 코딩 테스트 문제 설명

- 참가자의 key가 정상적이지 않다면, 다음과 같이 응답한다.

[응답(response) 양식]

```
{  
  "status": "unknown authentication key"  
}
```

status	사용자의 요청(request)에 대한 처리 상태
--------	----------------------------

개발형 코딩 테스트  
문제 설명

Query API

개발형  
코딩 테스트  
문제 설명

- 참가자가 돌의 위치(position) 값을 보내지 않았다면, 다음과 같이 응답한다.

[응답(response) 양식]

```
{
  "status": "position is needed"
}
```

status	사용자의 요청(request)에 대한 처리 상태
--------	----------------------------

## 개발형 코딩 테스트 문제 설명

## Query API

## 개발형 코딩 테스트 문제 설명

- 참가자가 보낸 돌의 위치(position)에 이미 돌이 있다면, 다음과 같이 응답한다.

[응답(response) 양식]

```
{  
  "status": "already taken position"  
}
```

status	사용자의 요청(request)에 대한 처리 상태
--------	----------------------------

## 개발형 코딩 테스트 문제 설명

## Query API

## 개발형 코딩 테스트 문제 설명

- 참가자가 보낸 돌의 위치 값이 정상적이지 않은 경우, 다음과 같이 응답한다.
- 바둑판의 범위를 벗어났거나, 형식이 올바르지 않은 경우 등에 해당한다.

[응답(response) 양식]

```
{  
  "status": "invalid position"  
}
```

status	사용자의 요청(request)에 대한 처리 상태
--------	----------------------------