CS101 Homework #1

1. 'turn_right', 'look_back' 함수 정의

```
def turn_right_h():
 for i in range(3):
  hubo.turn_left()

def look_back_h():
 for i in range(2):
  hubo.turn_left()

def turn_right_a():
 for i in range(3):
  ami.turn_left()

def look_back_a():
 for i in range(2):
  ami.turn_left()
```

오른쪽으로 돌거나 뒤로 도는 함수를 정의해 주었다. 오른쪽으로 도는 함수 'turn_right'는 왼쪽으로 세 번 돌게 하였고, 뒤로 도는 함수 'look_back'은 왼쪽으로 두 번 돌게 하였다. 이때 hubo에 대해서는 '_h'를 붙여 주었고, ami에 대해서는 '_a'를 붙여주었다.

2. 'stepback_and_restore' 함수 정의

```
def stepback_and_restore():
 if hubo.on_beeper():
  look_back_h()
  hubo.move()
  while hubo.on_beeper():
   hubo.turn_left()
   hubo.pick_beeper()
```

숫자에 따라 왼쪽, 오른쪽, 혹은 앞으로 한 칸 움직였을 때 비퍼가 있다면 다시 돌아오고, 없다면 그 자리에 그대로 있는 'stepback_and_restore' 함수를 정의해 주었다. 비퍼가 있다면 뒤돌아 와서 while문을 통해 드랍했던 비퍼를 주우면서 처음 방향으로 돌아서는 과정이다.

3. 'move_one_step_or_stay' 함수 정의

```
def move_one_step_or_stay():
 a=randint(1,3)
 for i in range(a):
  hubo.drop_beeper()
 for i in range(6-a):
  hubo.turn_left()
 if hubo.front_is_clear():
  hubo.move()
  stepback_and_restore()
 else :
  for i in range(a):
   hubo.pick_beeper()
  for i in range(2+a):
   hubo.turn_left()
```

1, 2, 3 중에 랜덤으로 골라 a라고 하고, a만큼 비퍼를 드랍한 다음 6-a만큼 왼쪽 방향으로 돌아서면 a 개수만큼 비퍼를 드랍한 상태로 1일 때는 왼쪽으로, 2는 원래 방향으로, 3은 오른쪽 방향으로 돌아서 있게 된다. 그러고 나서 앞에 벽이 없다면 앞으로 한 칸 움직이고, 'stepback_and_restore' 함수를 실행시키고, 만약 벽이 있다면 앞으로 이동하지 않고, 그 자리의 비퍼를 모두 줍고, 2+a만큼 왼쪽

으로 돌아 원래 방향으로 돌아서게 해 주었다.

4. 'can_move' 함수 정의

```
def can_move():
 if hubo.front_is_clear():
  hubo.move()
  if not hubo.on_beeper():
   look_back_h()
   hubo.move()
   look_back_h()
   return True
  else:
   look_back_h()
   hubo.move()
   look_back_h()
 if hubo.right_is_clear():
  turn_right_h()
  hubo.move()
  if not hubo.on_beeper():
   look_back_h()
   hubo.move()
   look_back_h()
   hubo.turn_left()
   return True
  else:
   look_back_h()
   hubo.move()
   look_back_h()
   hubo.turn_left()
 if hubo.left_is_clear():
  hubo.turn_left()
  hubo.move()
  if not hubo.on_beeper():
   look_back_h()
   hubo.move()
   look_back_h()
   turn_right_h()
   return True
  else:
   look_back_h()
   hubo.move()
   look_back_h()
   turn_right_h()
 else:
  return False
```

나는 True인 경우를 따져주고 나머지 경우는 모두 False라고 하여 'can_move'를 정의해 주었다. 휴보 앞에 벽이 없다면 휴보가 앞으로 한칸 이동하여 비퍼가 있는지 체크해 준다. 이동했는데 비퍼가 없다면 제자리로 돌아오면서 True를 return해주고, 비퍼가 있다면 그냥 되돌아 온다. 왼쪽, 오른쪽에 대해서도 똑같이 해준다. 이 세가지 경우에만 True이고, 나머지 경우는 모두 False라서 'else : / return False' 를 해주었다.

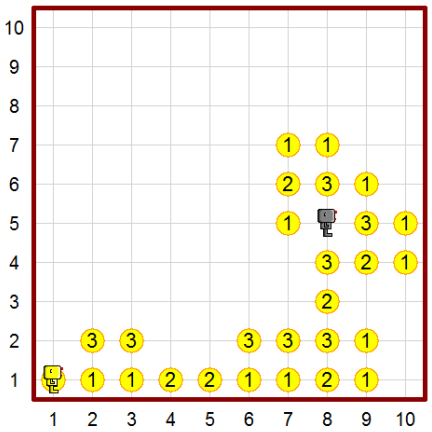
5. 'follow_trace()' 함수 정의

```
def follow_trace():
 while ami.on_beeper():
  look_back_a()
  while ami.on_beeper():
   turn_right_a()
   ami.pick_beeper()
  ami.move()
```

Ami를 뒤돌게 한 다음에 비퍼를 줍는 수만큼 오른쪽으로 돌게 하면 1일때는 왼쪽으로, 2일때는 앞으로, 3일때는 오른쪽으로 ami가 돌아서게 된다. 동시에 비퍼를 줍고 앞으로 이동시키면 비퍼를 다 줍고 이동할 수 있게 된다. 'while ami.on_beeper'를 해주었기 때문에 hubo가 있는 곳까지 이동할 수 있게 된다.

6. 결과

1) hubo가 비퍼를 다 깔았을 때



2) ami의 자취