

지도
방안

- 예시 프로그램을 통해, 프로그램 분석 능력을 신장시키기 위한 활동입니다.
- 우선 활동지에 있는 `travel_wall()` 함수 및 `if` 문의 역할을 적을 수 있도록 합니다.
- 프로그램 분석을 통해 로봇의 이동 경로를 그려 봅니다.
- 예시 프로그램 분석을 통해 로봇의 미로 탈출 방법이 적절하였는가를 토의하고, 토의 결과를 실제로 구현해 봅니다.

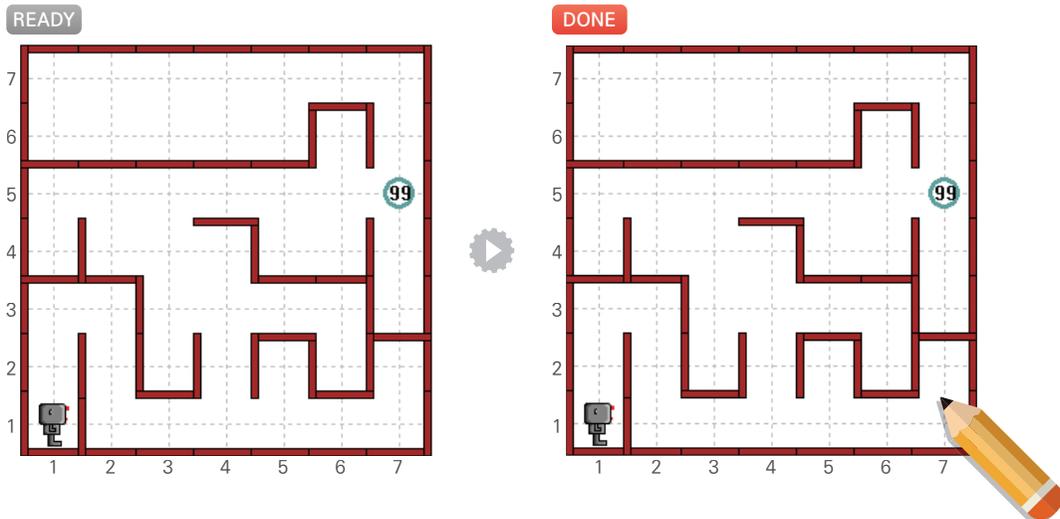
B-1 공주님을 찾아야 해요



긴급 의뢰예요! 산책하던 공주님이 미로에 갇혀버렸다고 하네요!
미로를 탐색해서 공주님께 도달하는 프로그램을 빨리 만들어 주세요!

의뢰서

- 로봇이 자동으로 미로를 탐색해 공주님(비퍼)에게 도달할 수 있도록 해 주세요.



⚙️ 아래의 예시 프로그램에 따른 로봇의 탐색 경로를 예측하여 이동경로를 그려 봅시다.

```

1 while not on_beeper():
2     if right_is_clear():
3         repeat(turn_left,3)
4         move()
5     elif front_is_clear():
6         move()
7     else :
8         repeat(turn_left,2)
9 turn_off()

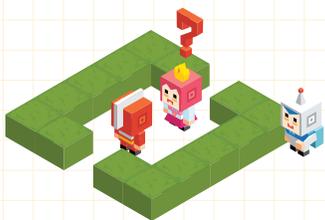
```

❁ 앞의 예시 프로그램은 우수법에 의한 미로 탐색법입니다. 좌수법을 이용하여 구현해 봅시다.

1
2
3
4
5
6
7
8
9

❁ 미로 탈출 프로그램을 직접 만들어 봅시다. (단, 월드 크기는 10×10 이상으로 합니다.)

문제해결노트



```

31
32 repeat(move,3)
33 turn_left()
34 compute()
35 repeat(turn_left,3)
36 sum += carry
37 compute()
38 move()
39 turn_off()
    
```

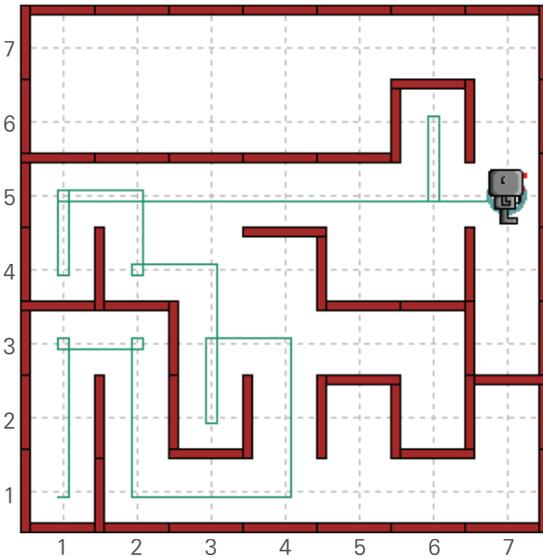
34 compute()
pick() 함수를 2번 수행하고 위치를 이동해 result_put() 함수를 실행하는 함수입니다.

36-37 일의 자리의 carry값을 sum에 합산한 후, 십의 자리 계산을 수행하는 부분입니다.

8-1 공주님을 찾아야 해요

- ① 로봇이 왼쪽에 벽을 두고 벽을 따라 미로를 탐색하도록 합니다.
- ② 더 이상 탐색할 경로가 없으면 뒤로 돌아 이동합니다.
- ③ ①, ② 동작은 while문을 활용하여 비퍼를 만날 때까지 반복합니다.

🌸 로봇의 이동경로



```

1 while not on_beeper():
2     if right_is_clear():
3         repeat(turn_left,3)
4         move()
5     elif front_is_clear():
6         move()
7     else :
8         repeat(turn_left,2)
9 turn_off()
    
```

1 비퍼가 없을 동안 계속 진행

2-5 로봇의 오른쪽에 벽이 없으면 우회전 한 후 한칸 이동합니다.

6-7 오른쪽이 막혀있고 앞쪽이 비어있으면 앞으로 한칸 이동합니다.

8 오른쪽과 앞쪽이 막혀있으면 뒤로 돌아섭니다.

❗ 좌수법에 의한 미로 탐색법

```

1 while not on_beeper():
2     if left_is_clear():
3         turn_left()
4         move()
5     elif front_is_clear():
6         move()
7     else :
8         repeat(turn_left,2)
9 turn_off()

```

9-1 공주님의 정원사가 되었어요

- ❶ 죽은 꽃의 수를 세는 dead변수는 pick()함수에도 사용되므로 전역(global)변수로 선언합니다.
- ❷ while문을 활용하여 바닥에 놓인 비퍼의 수를 파악하고, if문과 elif문을 활용해 파악한 비퍼의 수에 따른 명령을 구현합니다.

<pre> 1 dead = 0 2 3 def pick(): 4 repeat(move,2) 5 a = 0 6 global dead 7 while on_beeper(): 8 pick_beeper() 9 a += 1 10 if a==0: 11 repeat(put_beeper,4) 12 elif a==1: 13 dead += 1 14 repeat(put_beeper,5) 15 elif a==2 or a==3: 16 repeat(put_beeper,3) 17 18 turn_left() 19 repeat(move,2) 20 repeat(turn_left,3) 21 22 repeat(pick,5) 23 move() 24 turn_left() 25 repeat(move,4) 26 turn_left() 27 move() 28 repeat(put_beeper,dead) 29 repeat(turn_left,2) 30 move() 31 32 turn_off() </pre>	<p>1 죽은 꽃의 수를 저장하는 전역변수입니다.</p> <p>3-17 pick() 앞으로 두 칸 이동한 다음 비퍼를 주우며 갯수를 파악하여 0이면 비퍼를 4개 놓고, 1이면 dead값 1 증가 후 비퍼를 5개 놓고, 2 또는 3이면 비퍼를 3개 놓는 함수입니다.</p> <p>18-20 탐색을 시작할 위치로 이동합니다.</p> <p>22 pick() 함수를 5번 실행합니다.</p> <p>23-28 (11, 7) 자리에 죽은 꽃의 수(dead 변수의 값)만큼 비퍼를 내려놓습니다.</p> <p>29-30 되돌아 나오는 부분입니다.</p>
---	---