

명령어를 입력하세요: +x
'x'를 큐에 추가했습니다.
현재 큐: x

명령어를 입력하세요: +y
'y'를 큐에 추가했습니다.
현재 큐: x y

명령어를 입력하세요: +z
'z'를 큐에 추가했습니다.
현재 큐: x y z

명령어를 입력하세요: +w
'w'를 큐에 추가했습니다.
현재 큐: x y z w

명령어를 입력하세요: -
'x'를 큐에서 제거했습니다.
현재 큐: y z w

명령어를 입력하세요: l
현재 큐: y z w

명령어를 입력하세요: p
큐의 맨 앞 데이터: y

명령어를 입력하세요: f
큐가 가득 찼나요? No

명령어를 입력하세요: e
큐가 비었나요? No

명령어를 입력하세요: #
큐에 있는 데이터 개수: 3

명령어를 입력하세요: h
Head 위치: 2

명령어를 입력하세요: t
Tail 위치: 4

명령어를 입력하세요: ?x
'x'가 큐에 있나요? No

명령어를 입력하세요: =>x
Tail 데이터를 'x'로 교체했습니다.
현재 큐: y z x

명령어를 입력하세요: c
큐를 비웠습니다.

명령어를 입력하세요: r
큐를 뒤집었습니다.
현재 큐: Queue is empty

명령어를 입력하세요: g 1
Error: Invalid position

명령어를 입력하세요: +x
'x'를 큐에 추가했습니다.
현재 큐: x

명령어를 입력하세요: +y
'y'를 큐에 추가했습니다.
현재 큐: x y

명령어를 입력하세요: g 1
1번 위치의 데이터: y

명령어를 입력하세요: d 1
1번 위치의 데이터를 제거했습니다.
현재 큐: x

명령어를 입력하세요: i x 1
'x'를 1번 위치에 삽입했습니다.
현재 큐: x x

명령어를 입력하세요: s
큐를 정렬했습니다.
현재 큐: x x

명령어를 입력하세요: HELP

명령어를 입력하세요: HELP

Head 위치: 1

===== 큐(Queue) 명령어 목록 =====

1. create - 큐 생성 및 초기화 (char my_queue[Max_Size])
2. +x - 데이터 'x'를 큐에 추가 (enqueue)
3. - - 큐에서 데이터 제거 (dequeue)
4. L - 큐의 모든 데이터 출력 (print)
5. P - 큐의 맨 앞 데이터 확인 (peek)
6. F - 큐가 가득 찼는지 확인 (is_full)
7. E - 큐가 비었는지 확인 (is_empty)
8. # - 큐에 있는 데이터 개수 확인 (data_count)
9. H - head 위치 반환 (head_position)
10. T - tail 위치 반환 (tail_position)
11. ?x - 'x'가 큐에 있는지 확인 (is_member)
12. =x - tail 데이터를 'x'로 교체 (replace)
13. C - 큐의 모든 데이터 삭제 (clear)

===== 추가 명령어 목록 =====

14. R - 큐의 데이터 순서 뒤집기 (reverse)
15. G n - n번 위치의 데이터 반환 (get_at)
16. D n - n번 위치의 데이터 제거 (remove_at)
17. I x n - n번 위치에 데이터 'x' 삽입 (insert_at)
18. S - 큐의 데이터 오름차순 정렬 (sort_queue)
19. HELP - 모든 명령어 목록 표시 (show_commands)

명령어를 입력하세요: q

프로그램을 종료합니다.

PS C:\Dev\C\ds9> █