이번 주차에는 큐에 관한 예제와 연습문제를 풀어보고 잘 되지 않아 개인적으로 답지를 참고하여 분석해보았습니다.

예제 코드작성

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #define MAX\_QUEUE\_SIZE 5  typedef int element;  typedef struct {  int front;  int rear;  element data[MAX\_QUEUE\_SIZE];  }QueueType;  void error(char\* message)  {  fprintf(stderr, "%s\n", message);  exit(1);  }  void init\_queue(QueueType\* q)  {  q->rear = -1;  q->front = -1;  }  void queue\_print(QueueType\* q)  {  for (int i = 0; i < MAX\_QUEUE\_SIZE; i++)  {  if (i <= q->front || i > q->rear)  printf(" |");  else  printf("%d|", q->data[i]);  }  printf("\n");  }  int is\_full(QueueType\* q)  {  if (q->rear == MAX\_QUEUE\_SIZE - 1)  return 1;  else  return 0;  }  int is\_empty(QueueType\* q)  {  if (q->front == q->rear)  return 1;  else  return 0;  }  void enqueue(QueueType\* q, int item)  {  if (is\_full(q))  {  error("큐가 포화상태입니다.");  return;  }  q->data[++(q->rear)] = item;  }  int dequeue(QueueType\* q)  {  if (is\_empty(q))  {  error("큐가 공백상태입니다.");  return -1;  }  q->data[++(q->front)];  }  int main()  {  int item = 0;  QueueType q;  init\_queue(&q);  enqueue(&q, 10); queue\_print(&q);  enqueue(&q, 20); queue\_print(&q);  enqueue(&q, 30); queue\_print(&q);  dequeue(&q); queue\_print(&q);  dequeue(&q); queue\_print(&q);  dequeue(&q); queue\_print(&q);  return 0;  } |

교재 연습문제

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #define MAX\_QUEUE\_SIZE 100  // ==== 덱 코드 시작 =====  typedef int element;  typedef struct { // 큐 타입  element data[MAX\_QUEUE\_SIZE];  int front, rear;  } DequeType;  // 오류 함수  void error(char\* message)  {  fprintf(stderr, "%s\n", message);  exit(1);  }  // 초기화  void init\_deque(DequeType\* q)  {  q->front = q->rear = 0;  }  // 공백 상태 검출 함수  int is\_empty(DequeType\* q)  {  return (q->front == q->rear);  }  // 포화 상태 검출 함수  int is\_full(DequeType\* q)  {  return ((q->rear + 1) % MAX\_QUEUE\_SIZE == q->front);  }  // 원형큐 출력 함수  void deque\_print(DequeType\* q)  {  printf("DEQUE(front=%d rear=%d) = ", q->front, q->rear);  if (!is\_empty(q)) {  int i = q->front;  do {  i = (i + 1) % (MAX\_QUEUE\_SIZE);  printf("%d | ", q->data[i]);  if (i == q->rear)  break;  } while (i != q->front);  }  printf("\n");  }  // 삽입 함수  void add\_rear(DequeType\* q, element item)  {  if (is\_full(q))  error("큐가 포화상태입니다");  q->rear = (q->rear + 1) % MAX\_QUEUE\_SIZE;  q->data[q->rear] = item;  }  // 삭제 함수  element delete\_front(DequeType\* q)  {  if (is\_empty(q))  error("큐가 공백상태입니다");  q->front = (q->front + 1) % MAX\_QUEUE\_SIZE;  return q->data[q->front];  }  // 삽입 함수  void add\_front(DequeType\* q, element val)  {  if (is\_full(q))  error("큐가 포화상태입니다");  q->data[q->front] = val;  q->front = (q->front - 1 + MAX\_QUEUE\_SIZE) % MAX\_QUEUE\_SIZE;  }  element delete\_rear(DequeType\* q)  {  int prev = q->rear;  if (is\_empty(q))  error("큐가 공백상태입니다");  q->rear = (q->rear - 1 + MAX\_QUEUE\_SIZE) % MAX\_QUEUE\_SIZE;  return q->data[prev];  }  // ==== 덱 코드 끝 =====  int get\_count(DequeType\* q)  {  int count = q->rear - q->front;  if (count < 0) count += MAX\_QUEUE\_SIZE;  return count;  }  int main()  {  DequeType queue; int equal = 1;  char s[100] = "madam";  init\_deque(&queue);  for (int i = 0; i < (int)strlen(s); i++)  {  add\_rear(&queue, s[i]);  }  while ((get\_count(&queue) > 1) & equal)  {  char first = delete\_front(&queue);  char last = delete\_rear(&queue);  if (first != last)  equal = 0;  }  printf("제출자:김건희\n");  printf("<실행결과>\n");  if (equal)  printf("회문이다.\n");  else  printf("회문이 아니다.\n");  } |