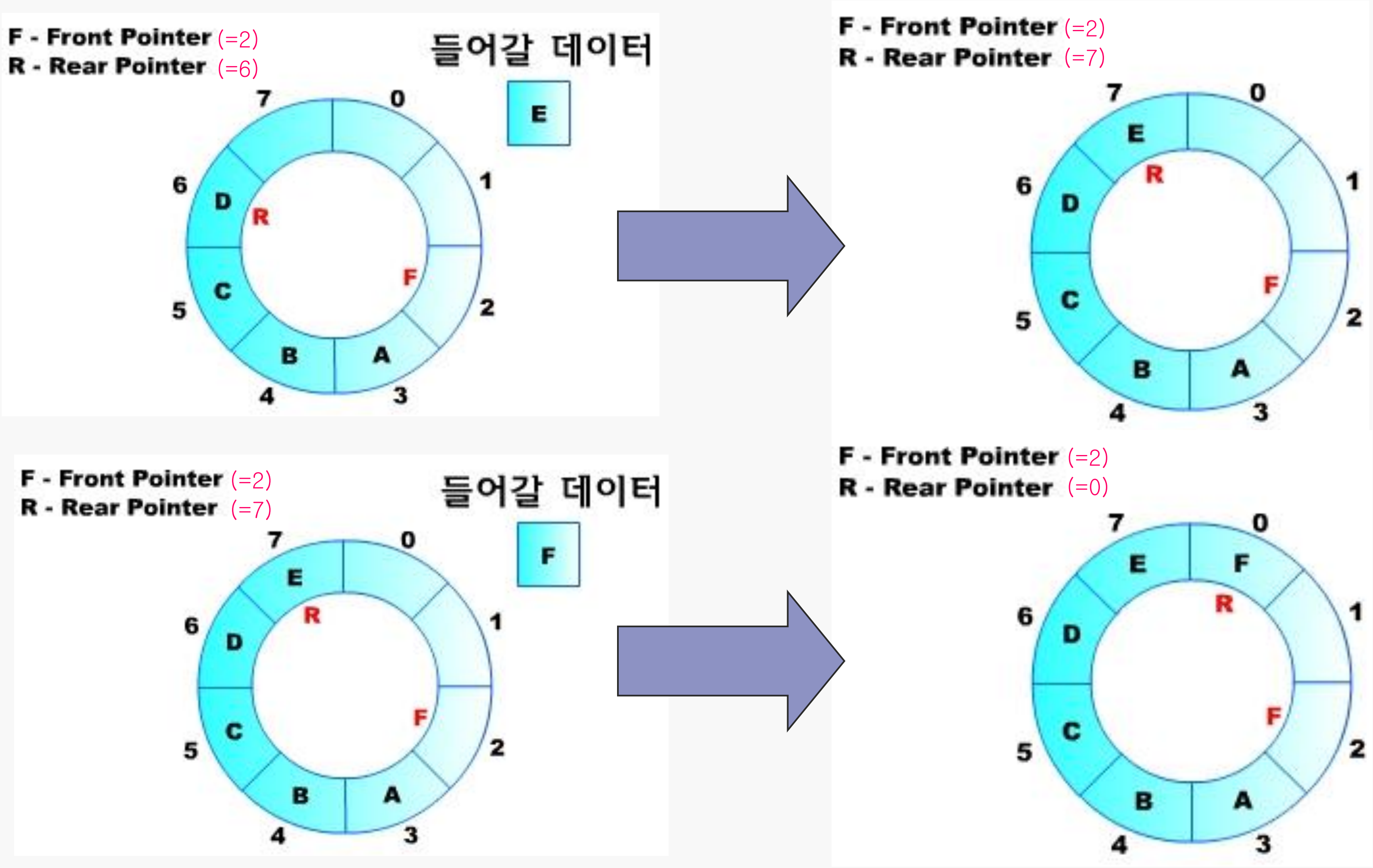
Project #6(Pjt06_uart_stdout_cir_queue)

- Circular Queue – insert



F - Front Pointer (=2)
R - Rear Pointer (=6)

들어갈 데이터

F - Front Pointer (=2)
R - Rear Pointer (=7)

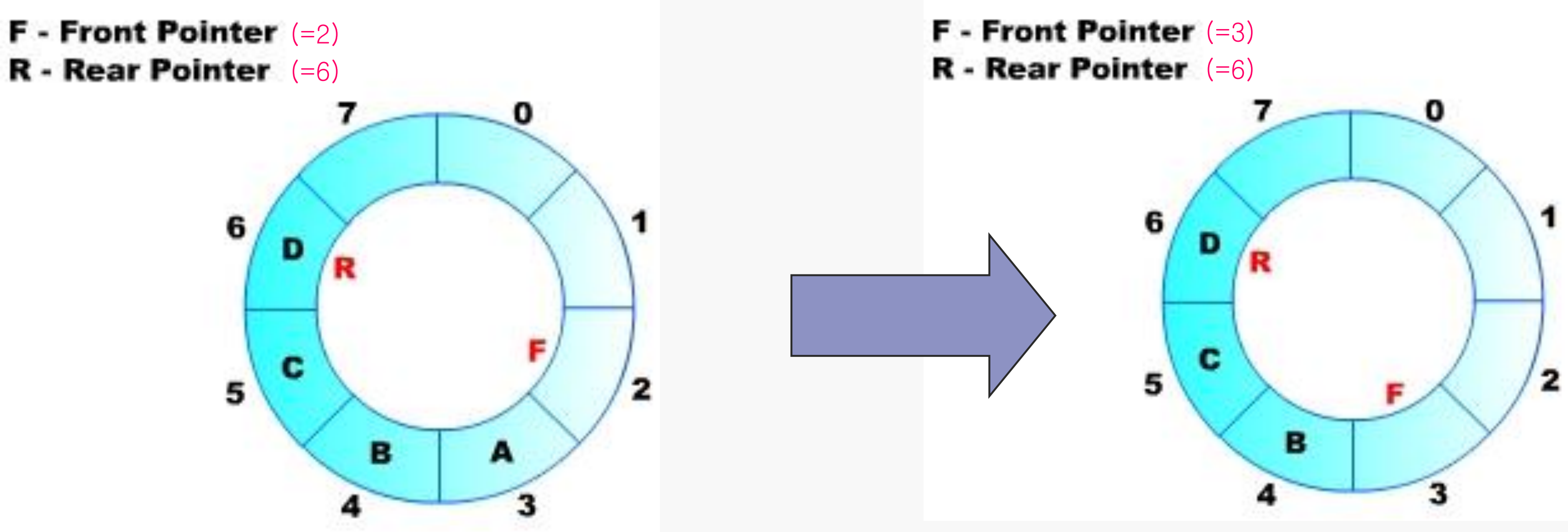
F - Front Pointer (=2)
R - Rear Pointer (=7)

들어갈 데이터

F - Front Pointer (=2)
R - Rear Pointer (=0)

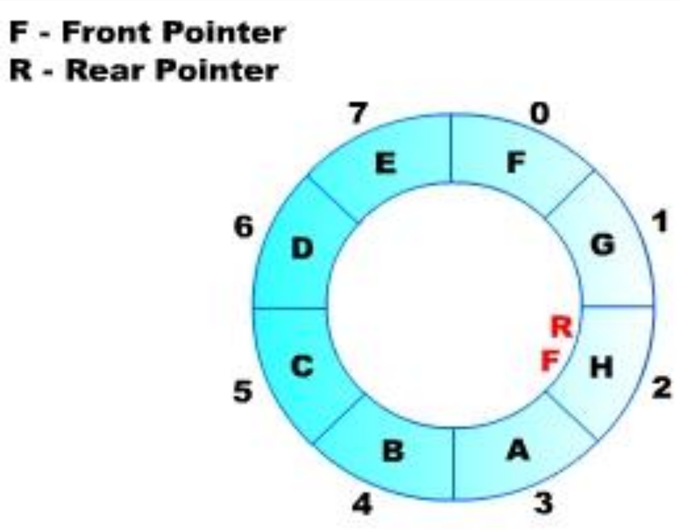
Project #6(Pjt06_uart_stdout_cir_queue)

- Circular Queue – delete

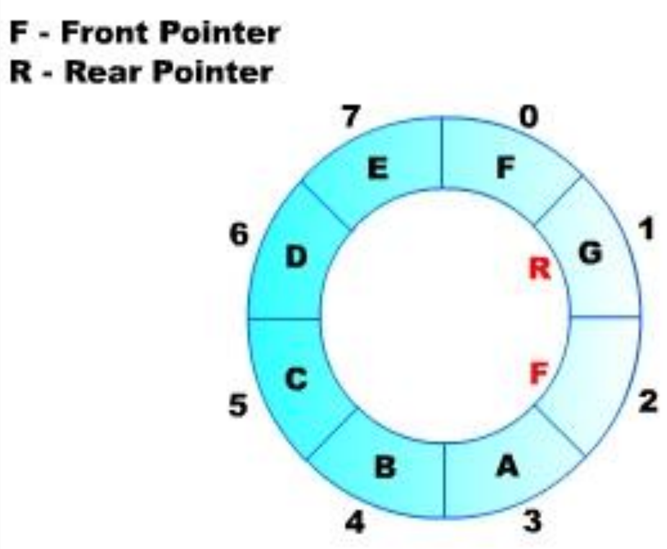


Project #6(Pjt06_uart_stdout_cir_queue)

- Circular Queue – full & empty



empty (F==R)



full (R+1==F)

reference ➔ http://blog.naver.com/limhj5798/70016683282

Project #6(Pjt06_uart_stdout_cir_queue)

- Data Structure for UART0 device driver(Circular Queue)

```
#define Q_SIZE  256

char q[Q_SIZE]
int  f, r;
```

```
void  q_init()
{
      f = r = 0;
}
```

```
int  q_insert(char ch)
{
      if ((r + 1) % Q_SIZE == f)
            return(0);     // full
      r = (r + 1) % Q_SIZE;
      q[r] = ch;
      return(1);
}
```

```
int  q_delete()
{
      if (r == f)
            return(0);       // empty
      f = (f + 1) % Q_SIZE;
      return(q[f]);
}
```

Project #6(Pjt06_uart_stdout_cir_queue)

- UART0 output device driver based on I/O interrupt

```
#include <stdio.h>
#include <avr/io.h>
#include <compat/deprecated.h>

FILE   Mystdout = FDEV_SETUP_STREAM (uart_putchar, NULL,_FDEV_SETUP_WRITE);
char   uart_busy;
```

```
void  uart_init()
{
    stdout = &Mystdout;
    uart_busy = 0;        // false
    q_init();
    UBRR0H = 0x00; UBRR0L = 0x07;
    sbi(UCSR0A, U2X0);    // UCSR)A |= (1 << U2X0);
    sbi(UCSR0B, TXEN0);
    sbi(UCSR0B, TXCIE0);
}
```

```
#include <stdio.h>
#include <avr/io.h>
#include <util/delay.h>

int  uart_putchar(char ch, FILE *stream)
{
     if (ch == ‘₩n’)
         uart_putchar(‘₩r’, stream);

     cli();  /* clear interrupt flag */
     if (!uart_busy) {
         UDR0 = ch;
         uart_busy = 1;
     }
     else {
         while(q_insert(ch) == 0) {
              sei();
              _delay_us(100);
              cli();
         }
     }
     sei();   /* set interrupt flag */

     return(1);
 }
```

```
#include <avr/interrupt.h>

ISR(USART0_TX_vect)
{
  char  ch;

  if ((ch = q_delete()) == 0)
     uart_busy = 0;
  else
     UDR0 = ch;
}
```

```
main()
{
   uart_init();
   sei();
   app_prime(2000);
   while(1);
}
```

```
int is_prime(int  n)
{
   int  i;
   for (i = 2; i <= n/2; i++)
      if ((n % i) == 0)
         return(0);
   return(1);
}
app_prime(int  t)
{
   int  n, count = 0;
   for (n = 2; n <= t; n++) {
     if  (is_prime(n)) {
        count++;
        printf(“%d  is a prime ”
                “number !!!₩n”, n);
     }
   }
   printf(“count=%d₩n”, count);
}
```