```c
//Implent a circular array queue
#include <stdio.h>
#define MAX 5

struct caq
{
    int array[MAX];
    int front;
    int rear;
};

void push(caq *cir, int x)
{
    if((cir->rear + 1)%MAX == cir->front)
        return;
    else
    {
        if(cir->rear == MAX-1)
        {
            cir->rear = (cir->rear+1)%MAX;
            cir->array[cir->rear] = x;
            return;
        }
        cir->array[cir->rear] = x;
        cir->rear = (cir->rear+1)%MAX;
    }
}

int pop(caq *cir)
{
    if(cir->rear == cir->front)
        return -1;
    int rr = cir->array[cir->front];
    cir->front = (cir->front+1)%MAX;
    return rr;
}

void display(caq *cir)
{
    if(cir->rear == cir->front)
        return;
    if(cir->front < cir->rear)
    {
        for(int i=cir->front; i<cir->rear ; i++)
        {
            printf("%6d", cir->array[i]);
        }
    }
    else
    {
        for(int i=cir->front ; i<MAX-1 ; i++)
        {
            printf("%6d", cir->array[i]);
        }
        for(int i=0 ; i<=cir->rear ; i++)
        {
            printf("%6d", cir->array[i]);
        }
    }
    printf("\n");
}

int main(int argc, char const *argv[])
{
    caq cir;
    cir.front = cir.rear = 0;
    push(&cir, 1);
    push(&cir, 2);
    push(&cir, 3);
```

```
    push(&cir, 4);
    display(&cir);
    printf("After removing 1 element from queue\n");
    pop(&cir);
    display(&cir);
    push(&cir, 6);
    printf("After adding 1 element to queue\n");
    display(&cir);
}
/*OUTPUT
     1     2     3     4
After removing 1 element from queue
     2     3     4
After adding 1 element to queue
     2     3     4     6
*/
```