
5장 연습문제 및 회문판별 프로그램(덱)

| 학과 컴퓨터공학과

| 학번 201821405

| 성명 장경원

| 제출일 5/23

5장 연습문제 1~8번

5장 연습문제 11번: 회문 판별 프로그램; 덱 사용

회문 판별 프로그램 소스코드

<Deque.cpp>

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream> // strlen 함수
#include <ctype.h> // tolower 함수

#define MAX_QUEUE_SIZE 50
typedef char element;
typedef struct {
    element data[MAX_QUEUE_SIZE];
    int front, rear;
}DequeType;

void init_deque(DequeType *q)
{
    q->front = q->rear = 0;
}

int is_empty(DequeType *q)
{
    return (q->front == q->rear);
}

int is_full(DequeType *q)
{
    return ((q->rear + 1) % MAX_QUEUE_SIZE == q->front);
}

void deque_print(DequeType *q)
{
    printf("DEQUE(front = %d rear = %d) = ", q->front, q->rear);
    if (!is_empty(q))
    {
        int i = q->front;
        printf("! ");
        do {
            i = (i + 1) % (MAX_QUEUE_SIZE);
            printf("%c | ", q->data[i]);
            if (i == q->rear)
```

```

        break;
    } while (i != q->front);
    printf("\n");
}

}

void add_rear(DequeType *q, element item)
{
    if (is_full(q))
        printf("큐가 포화상태입니다");
    q->rear = (q->rear + 1) % MAX_QUEUE_SIZE;
    memcpy(&(q->data[q->rear]), &item, sizeof(element));
}

element delete_front(DequeType* q)
{
    if (is_empty(q))
        printf("큐가 공백상태입니다");
    q->front = (q->front + 1) % MAX_QUEUE_SIZE;
    return q->data[q->front];
}

element get_front(DequeType* q)
{
    if (is_empty(q))
        printf("큐가 공백상태입니다");
    return q->data[(q->front + 1) % MAX_QUEUE_SIZE];
}

void add_front(DequeType* q, element val)
{
    if (is_full(q))
        printf("큐가 포화상태입니다");
    memcpy(&(q->data[q->front]), &val, sizeof(element));
    q->front = (q->front - 1 + MAX_QUEUE_SIZE) % MAX_QUEUE_SIZE;
}

element delete_rear(DequeType* q)
{
    int prev = q->rear;
    if (is_empty(q))
        printf("큐가 공백상태입니다");
    q->rear = (q->rear - 1 + MAX_QUEUE_SIZE) % MAX_QUEUE_SIZE;
    return q->data[prev];
}

```

```

element get_rear(DequeType* q)
{
    if (is_empty(q))
        printf("큐가 공백상태입니다");
    return q->data[q->rear];
}

bool checkStringNum(int num)
{
    if (num == 1) return false; // 1글자일시 불가능
    else
        return true;
}

int riddleSave(DequeType* q, char* string)
{
    int count = 0; // 문자열 길이 카운트하는 변수
    for (unsigned int i = 0; i <= strlen(string); i++)
    {
        if (string[i] >= 'A' && string[i] <= 'Z') // 대문자면 소문자로 바꿔서 push
        {
            add_rear(q, tolower(string[i]));
            count++; // 문자열 길이 증가
        }
        else if (string[i] >= 'a' && string[i] <= 'z') // 소문자면 바로 push
        {
            add_rear(q, string[i]);
            count++;
        }
        else
            continue;
    }
    return count;
}

bool checkPalindrome(DequeType* q, int count)
{
    if (count % 2 == 1)
        // 문자열 길이가 홀수일 시 길이 -1을 해준다: 가운데 값은 비교하지 않아도
        // 되기 때문에
        count--;
    for (int i = 1; i <= count / 2; i++) // 문자열의 길이의 절반만큼 반복
    {
        if (delete_front(q) == delete_rear(q)) // 앞에서 하나 뒤에서 하나 비교

```

```

        continue;
    else return false;
}

return true;
}

int get_count(DequeType* q)
{
    int extraFront = 0, extraRear = 0, count = 0;
    extraFront = q->front; // front, rear값에 영향을 주지 않도록 값만 복사
    extraRear = q->rear;
    for (; extraFront != extraRear; extraFront = (extraFront + 1) % MAX_QUEUE_SIZE)
        // front == rear 가 될때까지 증가
        count++;

    return count;
}

int main(void)
{
    char string[MAX_QUEUE_SIZE] = { 0, };
    int count = 0, count2 = 0;
    DequeType dequePalindrome;
    init_deque(&dequePalindrome);

    printf("문자열을 입력하시오: ");
    scanf_s("%[^\n]s", string, sizeof(string));
    count2 = riddleSave(&dequePalindrome, string);
    count = get_count(&dequePalindrome);
    printf("\n입력하면서 카운트 = %d", count2);
    printf("\nfront, rear를 통한 카운트 = %d\n\n", count);
    deque_print(&dequePalindrome);

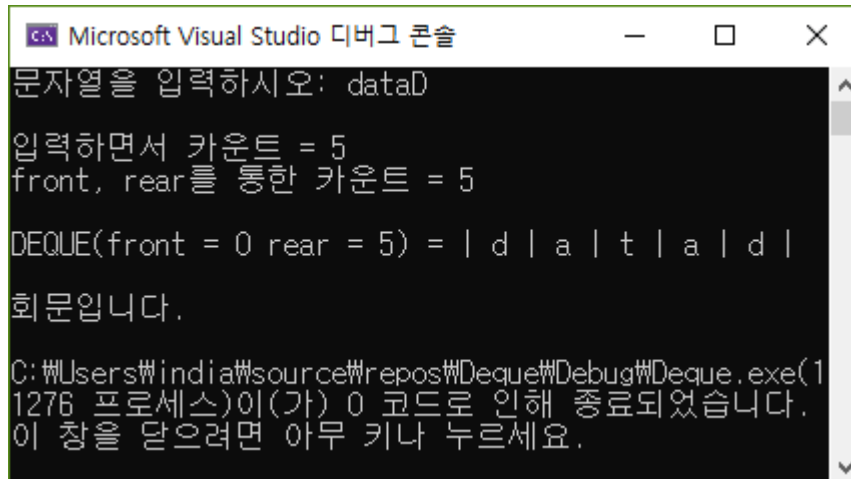
    if (checkStringNum(count)) // 문자열의 길이가 1 일경우 거짓 == 종료
    {
        if (checkPalindrome(&dequePalindrome, count)) // 회문 확인
        {
            printf("\n회문입니다.\n");
        }
        else
            printf("\n회문이 아닙니다.\n");
    }
    else
        printf("\n회문 일 수 없습니다.\n");
}

```

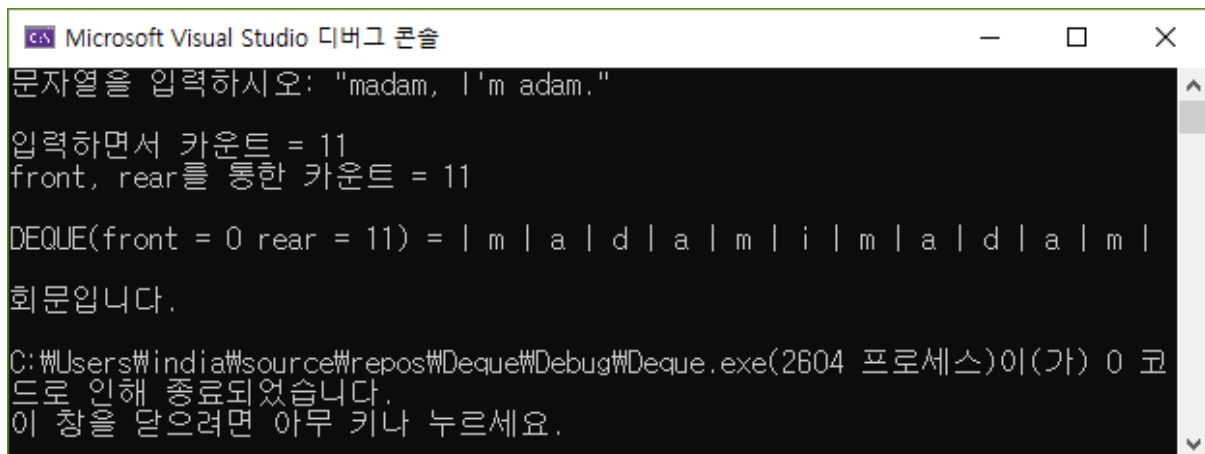
}

입출력 창

문자열 입력 받고 문자를 대문자면 소문자로, 소문자면 바로 텍에 저장, 그리고 두 가지의 경우로 문자열 길이 카운트(1. 기존 문자열을 소문자로 텍에 저장하면서 카운트 2. 저장된 문자열을 토대로 텍에 있는 front와 rear를 통한 카운트)하고 해당 결과 출력한 다음에 텍에 저장된 내용도 출력된다. 그 후 회문인지 아닌지 결과값이 나온다.



```
Microsoft Visual Studio 디버그 콘솔
문자열을 입력하시오: dataD
입력하면서 카운트 = 5
front, rear를 통한 카운트 = 5
DEQUE(front = 0 rear = 5) = | d | a | t | a | d |
회문입니다.
C:\Users\windia\source\repos\Deque\Debug\Deque.exe(1276 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```



```
Microsoft Visual Studio 디버그 콘솔
문자열을 입력하시오: "madam, I'm adam."
입력하면서 카운트 = 11
front, rear를 통한 카운트 = 11
DEQUE(front = 0 rear = 11) = | m | a | d | a | m | i | m | a | d | a | m |
회문입니다.
C:\Users\windia\source\repos\Deque\Debug\Deque.exe(2604 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

프로그램 후기

이전 과제(4장 연습문제 16번)에서 짰 palindrome.cpp에서 쓰던 함수를 수정하여 코드를 짰다. 이전 과제의 코드보다 더 쉽고 연산횟수가 줄어들어서 더 좋은 프로그램이 된 것 같다. 그리고, **5장 연습문제 8번**에서 생각한 논리대로 텍에서도 구현을 해보았다. 원형큐와 텍의 구조가 유사해서 구현하기 쉬웠던 것 같다.