



7 주차 실습





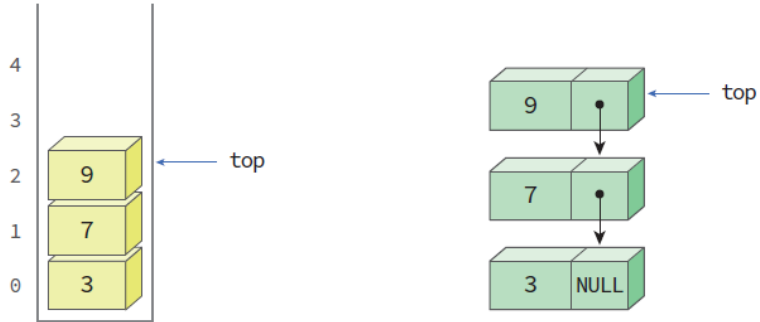
연결 리스트

```
#include <stdio.h>
#include <stdlib.h>
typedef int element;
typedef struct ListNode {
    element data;
    struct ListNode* link;
} ListNode;
void error(char* message){
    fprintf(stderr, "%s\n", message);
    exit(1);
}
ListNode* insert_first(ListNode* head,
int value){
    ListNode* p =
    (ListNode*)malloc(sizeof(ListNode));
    p->data = value;
    p->link = head;
    head = p;
    return head;
}
ListNode* insert(ListNode* head,
ListNode* pre, element value){
    ListNode* p =
    (ListNode*)malloc(sizeof(ListNode));
    p->data = value;
    p->link = pre->link;
    pre->link = p;
    return head;
}
```

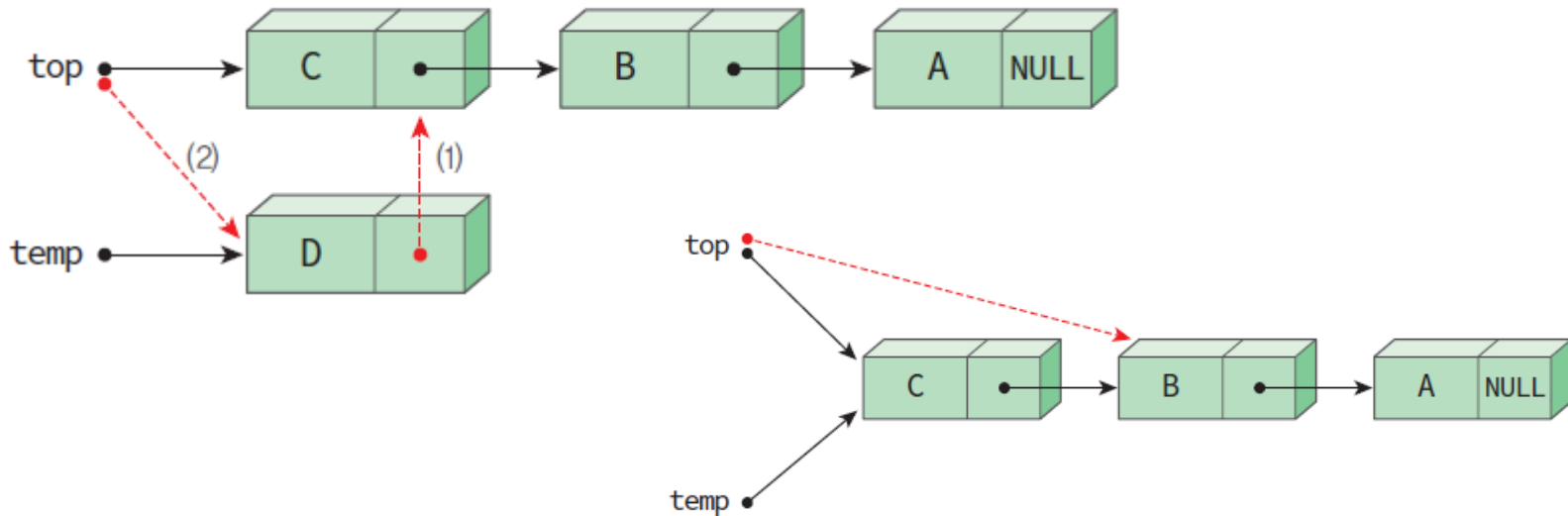
```
ListNode* delete_first(ListNode* head){
    ListNode* removed;
    if (head == NULL)
        return NULL;
    removed = head; // (1)
    head = removed->link; // (2)
    free(removed); // (3)
    return head; // (4)
}
ListNode* delete(ListNode* head, ListNode* pre){
    ListNode* removed;
    removed = pre->link;
    pre->link = removed->link; // (2)
    free(removed); // (3)
    return head; // (4)
}
void print_list(ListNode* head){
    for (ListNode* p = head; p != NULL; p = p->link)
        printf("%d->", p->data);
    printf("NULL\n");
}
```

실습 (partice_15.c)

 연결 리스트를 수정하여 스택을 구축 하시오



배열을 이용한 스택 연결 리스트를 이용한 스택



실습 (partice_15.c)

연결 리스트를 수정하여 스택을 구축 하시오

```
#include <stdio.h>
#include <stdlib.h>
typedef char element;
typedef struct StackNode {
    element data;
    struct StackNode* link;
} StackNode;

typedef struct {
    StackNode* top;
} LinkedStackType;

// 초기화 함수
void init(LinkedStackType* s){
    ...
}

// 공백 상태 검출 함수
int is_empty(LinkedStackType* s){
    ...
}

// 삽입 함수
void push(LinkedStackType* s, element item){
    ...
}
```

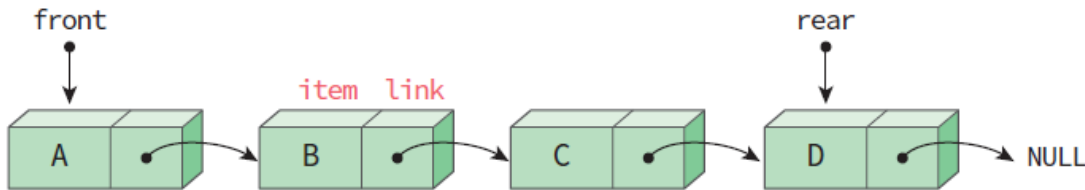
```
void print_stack(LinkedStackType* s){
    ...
}

// 삭제 함수
element pop(LinkedStackType* s){
    ...
}

int main(void) {
    char name[] = "Kwangseob Kim";
    LinkedStackType s;
    for (int i = 0; i < strlen(name); i++) {
        push(&s, name[i]);
    }
    while (!is_empty(&s)) {
        printf("%c", pop(&s));
    }
    return 0;
}
```

실습 (partice_16.c)

연결 리스트를 수정하여 큐를 만들어보시오



```
#include <stdio.h>
#include <stdlib.h>

typedef int element; // 요소의 타입
typedef struct QueueNode { // 큐의 노드의 타입
    element data;
    struct QueueNode* link;
} QueueNode;

typedef struct { // 큐 ADT 구현
    QueueNode* front, * rear;
} LinkedQueueType;

// 삽입 함수
void enqueue(LinkedQueueType* q, element data){
    ...
}
```

```
// 삭제 함수
element dequeue(LinkedQueueType* q){
    ...
}

int is_empty(LinkedQueueType* q) {
    ...
}

void init(LinkedQueueType* q) {
    ...
}

int main(void) {
    ... 피보나치 수열
}
```