

# Data Structure

Practical Class - Week 05

# Exercise01 - Skeleton 1

- LinkedList Stack 을 구현해 봅시다.
- LinkedList Stack 의 top 부터 bottom 까지의 items 를 출력하는 **printStack()** 함수 제공
- **Main Function** 제공
- Main Function 의 역할
  1. printstack Function 출력
  2. Push 값 수동 입력
  3. Pop 이 되는 과정 출력

- ❖ 예시답안이 있으므로 출력할 때 4 3 1 7 5 가 아닌 9 5 4 3 2 로 코드를 수정하여 제출 해야함!
- ❖ 불이행시 감점!!

```
45 // print all items in stack from top to bottom
46 void printStack() {
47     stackNode* p = top;
48     printf("\n STACK [ ");
49     while (p) {
50         printf("%d ", p->data);
51         p = p->link;
52     }
53     printf("] ");
54 }
55
56 void main(void) {
57     element item;
58     top = NULL;
59     printf("\n** Linked Stack **\n");
60     printStack();
61     push(4);    printStack();
62     push(3);    printStack();
63     push(1);    printStack();
64     push(7);    printStack();
65     push(5);    printStack();
66
67     item = pop(); printStack();
68     printf("\t pop  => %d", item);
69
70     item = pop(); printStack();
71     printf("\t pop  => %d", item);
72
73     item = pop(); printStack();
74     printf("\t pop  => %d", item);
75
76     item = pop(); printStack();
77     printf("\t pop  => %d", item);
78
79     item = pop(); printStack();
80     printf("\t pop  => %d", item);
81
82     getchar();
83 }
```

# Exercise01 – Skeleton 2

---

- LinkedList Stack 을 구현하기 위한 기본 **Library, Structure, isEmpty Function** 제공

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  typedef int element;
6
7  typedef struct stackNode {
8      element data;
9      struct stackNode *link;
10 } stackNode;
11
12 stackNode* top;           // define pointer top for pointing a top node
13
14 // isEmpty
15 int isEmpty() {
16     if (top == NULL) return 1;
17     else return 0;
18 }
```

## Exercise01 - Problem 01

---

```
20 // push
21 void push(element item) {
22
23
24
25
26 }
27
28 // pop
29 element pop() {
30     element item;
31     stackNode* temp = top;
32
33     if (top == NULL) { // if stack is empty
34         printf("\n\n Stack is empty !\n");
35         return 0;
36     }
37     else {
38
39
40
41
42     }
43 }
```

1) 주어진 조건에 맞게 LinkedList Stack push 함수를 작성하시오.

1) 주어진 조건에 맞게 LinkedList Stack push 함수를 작성하시오.

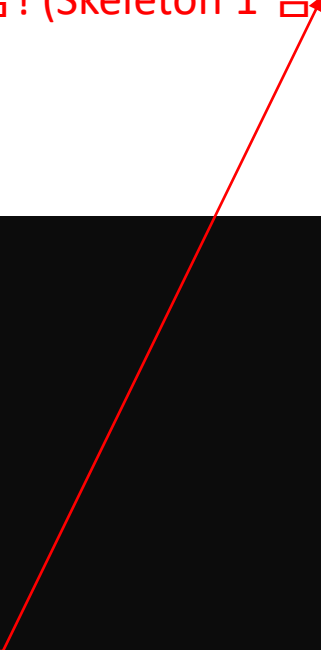
## Exercise01 - Output Example

---

- ❖ 예시답안이 있으므로 출력할 때 4 3 1 7 5 가 아닌 9 5 4 3 2 로 코드를 수정하여 제출 해야함! (Skeleton 1 참조)
- ❖ 불이행시 감점!!

```
** Linked Stack **

STACK [ ]
STACK [ 4 ]
STACK [ 3 4 ]
STACK [ 1 3 4 ]
STACK [ 7 1 3 4 ]
STACK [ 5 7 1 3 4 ]
STACK [ 7 1 3 4 ]      pop => 5
STACK [ 1 3 4 ]        pop => 7
STACK [ 3 4 ]          pop => 1
STACK [ 4 ]            pop => 3
STACK [ ]              pop => 4
```



# Exercise02 - Skeleton 1

- LinkedList Queue 을 구현해 봅시다.
- LinkedList Queue 의 **printLQ funtion** 제공
- LinkedList Queue 의 **main function** 제공
- 실질적으로 enqueue 출력과 Dequeue 를 출력

- ❖ 예시답안이 있으므로 출력할 때 C A B D E 가 아닌 J A G U U 로 코드를 수정하여 제출 해야함!
- ❖ 불이행시 감점!!

```
// print all items in the queue
void printLQ(LQueueType *LQ) {
    QNode *temp = LQ->front;
    printf(" Linked Queue : [");
    while (temp) {
        printf("%3c", temp->data);
        temp = temp->link;
    }
    printf(" ] ");
}

void main(void) {
    LQueueType *LQ = createLinkedListQueue(); // create a linked queue
    element data;

    printf("\n ***** Queue ***** \n");
    printf("\n enqueue C >>"); enqueue(LQ, 'C'); printLQ(LQ);
    printf("\n enqueue A >>"); enqueue(LQ, 'A'); printLQ(LQ);
    printf("\n enqueue B >>"); enqueue(LQ, 'B'); printLQ(LQ);
    printf("\n enqueue D >>"); enqueue(LQ, 'D'); printLQ(LQ);
    printf("\n enqueue E >>"); enqueue(LQ, 'E'); printLQ(LQ);

    printf("\n dequeue >>"); data = dequeue(LQ); printLQ(LQ);
    printf("\ndequeued data: %c", data);
    printf("\n dequeue >>"); data = dequeue(LQ); printLQ(LQ);
    printf("\ndequeued data: %c", data);
    printf("\n dequeue >>"); data = dequeue(LQ); printLQ(LQ);
    printf("\ndequeued data: %c", data);
    printf("\n dequeue >>"); data = dequeue(LQ); printLQ(LQ);
    printf("\ndequeued data: %c", data);
    printf("\n dequeue >>"); data = dequeue(LQ); printLQ(LQ);
    printf("\ndequeued data: %c", data);
    getchar();
}
```

## Exercise02 - Skeleton 2

---

- LinkedList Queue 를 구현하기 위한 **기본 Library, structure**
- 그리고 **linked queue** 를 **create** 하는 함수 제공
- LinkedList Queue 를 구현하기 위한 **isEmpty** 함수 제공

```
1  #include <stdio.h>
2  #include <malloc.h>
3
4  typedef char element;    // define element as character
5  typedef struct QNode {   // define linked queue node using structure
6      element data;
7      struct QNode *link;
8  } QNode;
9
10 typedef struct {          // define pointer front and rear
11     QNode *front, *rear;
12 } LQueueType;
13
14 // create linked queue
15 LQueueType *createLinkedQueue() {
16     LQueueType *LQ;
17     LQ = (LQueueType *)malloc(sizeof(LQueueType));
18     LQ->front = NULL;
19     LQ->rear = NULL;
20     return LQ;
21 }
22
23 // is empty
24 int isEmpty(LQueueType *LQ) {
25     if (LQ->front == NULL) {
26         printf(" Linked Queue is empty! ");
27         return 1;
28     }
29     else return 0;
30 }
```

## Exercise02 - Problem 01

---

```
32 // add item to the rear of the queue
33 void enqueue(LQueueType *LQ, element item) {
```

(1) 주어진 조건에 맞게 LinkedList Queue를 구현하기  
위한 enqueue 함수를 완성하시오.

```
45 }
46
47 // dequeue
48 element dequeue(LQueueType *LQ) {
```

(1) 주어진 조건에 맞게 LinkedList Queue를 구현하기  
위한 dequeue 함수를 완성하시오.

```
59 }
60 }
```



## Exercise02 - Output Example

---

❖ 예시답안이 있으므로 출력할 때 C A B D E 가 아닌 J A G U U 로 코드를 수정하여 제출 해야함! (Skeleton 1 참조)

❖ 불이행시 감점!!

\*\*\*\*\* Queue \*\*\*\*\*

```
enqueue C >> Linked Queue : [ C ]
enqueue A >> Linked Queue : [ C A ]
enqueue B >> Linked Queue : [ C A B ]
enqueue D >> Linked Queue : [ C A B D ]
enqueue E >> Linked Queue : [ C A B D E ]
dequeue >> Linked Queue : [ A B D E ]      dequeued data: C
dequeue >> Linked Queue : [ B D E ]        dequeued data: A
dequeue >> Linked Queue : [ D E ]          dequeued data: B
dequeue >> Linked Queue : [ E ]            dequeued data: D
dequeue >> Linked Queue : [ ]              dequeued data: E
```