

1. 다음과 같이 정렬되지 않은 정수 데이터를 입력하면서 정렬된 Linked List를 만들고 실행예와 같이 수행되는 프로그램을 작성하라.

### (1) 실행 순서

① 아래와 같은 데이터를 입력받으면서 **정렬된 Linked List**를 만든다. 입력데이터는 정렬되지 않은 값으로 중복 가능하다. (출력의 모든 수는 %2d의 형식을 따라 간다.)

입력 값 (행/열수는 3/10)
50 80 30 20 19 90 30 55 77 30
99 45 55 89 91 10 20 66 38 59
22 55 88 22 66 29 50 95 78 83

- ② Linked List의 처음부터 끝까지 노드의 데이터를 출력한다.
- ③ 점수를 입력받고 그 이하인 노드를 Linked List에서 삭제한다.
- ④ Linked List의 처음부터 끝까지 노드의 데이터를 출력한다.

### (2) 구현 세부사항 - 반드시 linked list로 풀어야 함

① 구조체 정의문은 다음과 같다.

```
typedef struct listNode *listPointer;
typedef struct listNode {
    int data;
    listPointer link;
} listNode;
listPointer first = NULL;
```

② 함수 ( Linked list를만 지키면 편하게 구현해도 됨...)

- find : insert 위치를 찾는 함수를 새롭게 정의
- insert, printList : Program 4.2, 4.4를 참고하여 **수정**
- delete : Program 4.3을 그대로 사용

※delete 함수명이 C++의 예약어와 같아서 에러로 표시되나 실행에 문제는 없음. 다른 이름으로 정의해도 됨.

```
void insert(listPointer *first, listPointer x)
/* insert a new node with data = 50 into the chain
   first after node x */
{
    listPointer temp;
    MALLOC(temp, sizeof(*temp));
    temp->data = 50;
    if (*first) {
        temp->link = x->link;
        x->link = temp;
    }
    else {
        temp->link = NULL;
        *first = temp;
    }
}
```

Program 4.2: Simple insert into list

---

```

void delete(listPointer *first, listPointer trail,
            listPointer x)
/* delete x from the list, trail is the preceding node
and *first is the front of the list */
if (trail)
    trail→link = x→link;
else
    *first = (*first)→link;
free(x);
}

```

---

**Program 4.3:** Deletion from a list

---

```

void printList(listPointer first)
{
    printf("The list contains: ");
    for (; first; first = first→link)
        printf("%4d", first→data);
    printf("\n");
}

```

---

**Program 4.4:** Printing a list

### (3) 실행 예

```

50 80 30 20 19 90 30 55 77 30 //서버 자동 입력
99 45 55 89 91 10 20 66 38 59
22 55 88 22 66 29 50 95 78 83 //서버 자동 입력 종료
50 80 30 20 19 90 30 55 77 30 //사용자 입력 전체 출력
99 45 55 89 91 10 20 66 38 59
22 55 88 22 66 29 50 95 78 83 //사용자 입력 전체 출력 종료
50 //특정값 입력 - 노드 삭제
80 90 55 77 99 55 89 91 66 59 //사용자 결과 출력
55 88 66 95 78 83 //사용자 결과 출력 종료

```

[Linked Stacks] 다음과 같은 스택을 생성하고 실행하는 프로그램을 작성하라. 이를 위해, push, pop, stackEmpty 함수를 구현하여야 한다. (3점)

(1) 실행 순서

① 아래와같은 입력으로 부터 학번 순으로 미리 정렬된 데이터를 입력받으면서 순서대로 Linked Stack을 구현한다. (과목번호, 학번, 성적)의 쌍으로 데이터들이 입력되며 각 과목별로 스택에 저장된다.

(과목번호 : 1자리 수, 학번 : 1자리수, 성적 : 2자리수)

0	1	95
1	1	80
2	1	89
0	2	45
1	2	81
0	3	45
1	3	12
2	3	33
0	4	99
1	4	94
2	4	91
0	5	67
2	5	49

② 각 과목 별로 학번의 역순으로 학번과 노드의 데이터(학번, 성적)를 출력하라.

(출력형식, 과목번호 : %d, 학번 : %d, 성적 : %2d)

(2) 구현 세부사항 - 반드시 linked list를 활용한 stack으로 작성해야 함

```
#define MAX_STACKS 3
typedef struct {
    int id:           //학번
    int grade:        //성적
} element;

typedef struct stack *stackPointer;
typedef struct stack {
    element data;
    stackPointer link;
}Node;

stackPointer top[MAX_STACKS];

/*****
top[i] = NULL, 0 ≤ i < MAX_STACKS // initial condition
top[i] = NULL, iff the ith stack is empty // boundary condition
*****/
```

---

```

void push(int i, element item)
{
    /* add item to the ith stack */
    stackPointer temp;
    MALLOC(temp, sizeof(*temp));
    temp->data = item;
    temp->link = top[i];
    top[i] = temp;
}

```

---

**Program 4.5:** Add to a linked stack

```

element pop(int i)
{
    /* remove top element from the ith stack */
    stackPointer temp = top[i];
    element item;
    if (!temp)
        return stackEmpty();
    item = temp->data;
    top[i] = temp->link;
    free(temp);
    return item;
}

```

---

**Program 4.6:** Delete from a linked stack

### (3) 실행 예

```

0 1 95//서버 입력 시작
1 1 80
2 1 89
0 2 45
1 2 81
0 3 45
1 3 12
2 3 33
0 4 99
1 4 94
2 4 91
0 5 67
2 5 49
0 0 0//서버 입력 종료
0 5 67//사용자 출력 시작
0 4 99
0 3 45
0 2 45
0 1 95
1 4 94
1 3 12
1 2 81
1 1 80
2 5 49
2 4 91
2 3 33
2 1 89//사용자 출력 종료

```