

# Garbage List

[code] DS1901-HW07

**Problem** 다음 그림과 같이 수업시간에 다룬 multiple array 기반의 양방향 연결리스트와 free 리스트에서 head에 새 노드를 삽입하는 insert\_node\_array() 함수와 head의 노드를 삭제하는 delete\_node\_array() 함수를 구현하시오. (0-based 인덱스를 구현하므로 주의할 것)

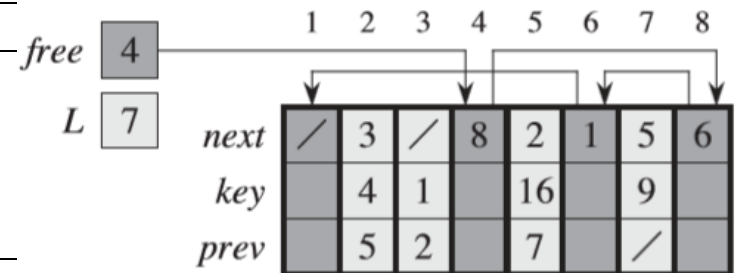
## [제약조건]

- 함수의 prototype은 다음과 같다.  
void insert\_node\_array(int \*next, int \*key, int \*prev, int n, int \*L, int \*free, int k)  
void delete\_node\_array(int \*next, int \*key, int \*prev, int n, int \*L, int \*free)  
next, key, prev : 연결리스트를 구현할 3배열 (오른쪽 그림 참조)  
n : 위 3 배열의 크기  
L, free : 연결리스트와 free리스트의 시작 인덱스  
k : 삽입하는 key 값
- 반드시 첫 번째 위치에 삽입, 삭제를 한다.
- 반드시 NIL은 -1로 표현한다.
- Overflow와 Underflow예외는 고려하지 않는다.
- 주의 제출하는 소스파일에는 main 함수는 포함하지 않는다.  
(다음 페이지 예처럼 mycode.c만 제출한다.)

## Submission

Due: 5월 31일 (금) 23시 59분 59초

19950001@ubuntu:~/DS\$ submit DS1901-HW07 mycode.c



# Self Test

```
19950001@ubuntu:~/DS$ sftest DS1901-HW07 mycode.c
```

```
MSG> Your source file was successfully compiled.
```

```
1 th. Testing 10 0 5 ---
```

```
Your answer is True : success.
```

```
2 th. Testing 10 5 0 ---
```

```
Your answer is True : success.
```

```
3 th. Testing 10 5 9 ---
```

```
Your answer is True : success.
```