

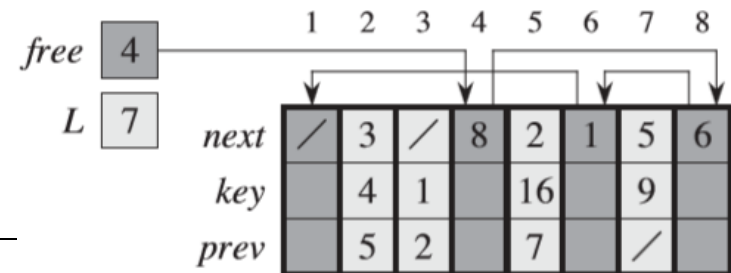
Garbage List

[code] DS2001-HW07

Problem 다음 그림과 같이 수업시간에 다룬 multiple-array 기반의 양방향 연결리스트와 free 리스트에서 head(리스트의 맨 처음)에 노드를 삽입하는 insert_node_head() 함수와 삭제하는 delete_node_head() 함수, tail(리스트의 맨 끝)에 노드를 삽입하는 insert_node_tail() 함수와 삭제하는 delete_node_tail() 함수를 구현하시오. (0-based 인덱스를 구현하므로 주의할 것)

[제약조건]

- 함수의 prototype은 다음과 같다.
void insert_node_head(int *next, int *key, int *prev, int n, int *L, int *free, int k)
void delete_node_head(int *next, int *key, int *prev, int n, int *L, int *free)
void insert_node_tail(int *next, int *key, int *prev, int n, int *L, int *free, int k)
void delete_node_tail(int *next, int *key, int *prev, int n, int *L, int *free)
- next, key, prev : 연결리스트를 구현하는 3개의 배열 (오른쪽 그림 및 수업내용 참조)
n : 위 3개 배열의 크기
L, free : 연결리스트와 free리스트의 시작 인덱스 (0-based)
k : 삽입하는 경우의 key 값
- **NIL**은 -1로 표현하며, Overflow와 Underflow에외는 고려하지 않는다.
- **주의) 제출하는 소스파일에는 main 함수는 포함하지 않는다.**
(다음 페이지 예처럼 mycode.c만 제출한다.)



Self Test

```
19950001@ubuntu:~/DS$ sftest DS2001-HW07 mycode.c
```

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

```
1 th. Testing 20 0 5 1 ---
```

```
Your answer is Correct : success.
```

```
2 th. Testing 20 5 0 1 ---
```

```
Your answer is Correct : success.
```

```
3 th. Testing 20 5 10 2 ---
```

```
Your answer is Correct : success.
```

Submission

Due: 6월 7일 (일) 23시 59분 59초

```
19950001@ubuntu:~/DS$ submit DS2001-HW07 mycode.c
```