

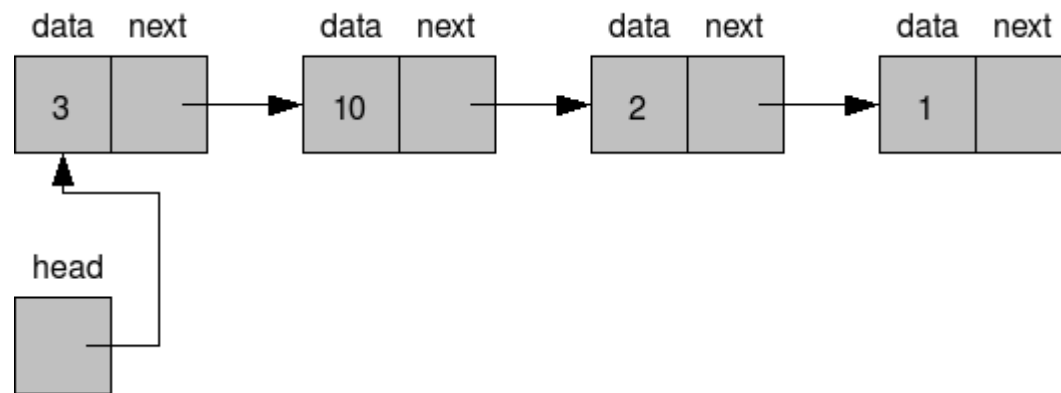
Linked List

HW10

- 1. String을 지원하는 Queue로 수정
- 2. main함수 작성
 - menQ, womenQ 두개의 Queue를 생성
 - 파일을 읽어서 menQ, womenQ에 데이터 추가
 - menQ혹은 womenQ에 데이터가 없을때 까지 매칭커플을 출력
 - 사람이 남은 Queue에 대해서 남은 갯수를 출력하고 맨 앞사람 이름을 출력한다.

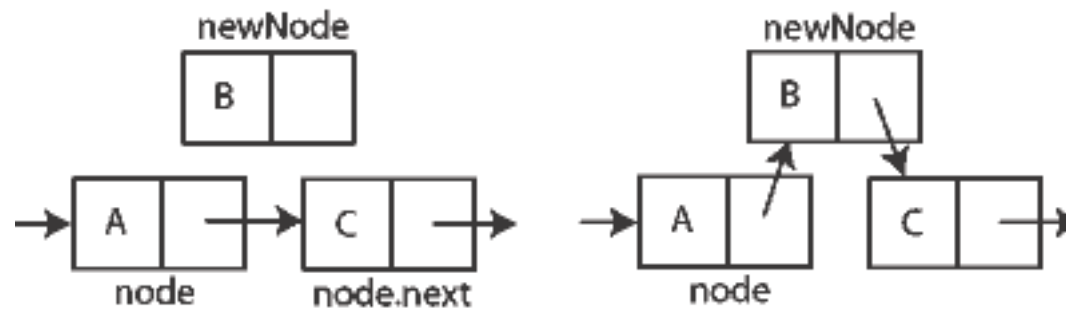
Linked List

- Insert/Delete at any place

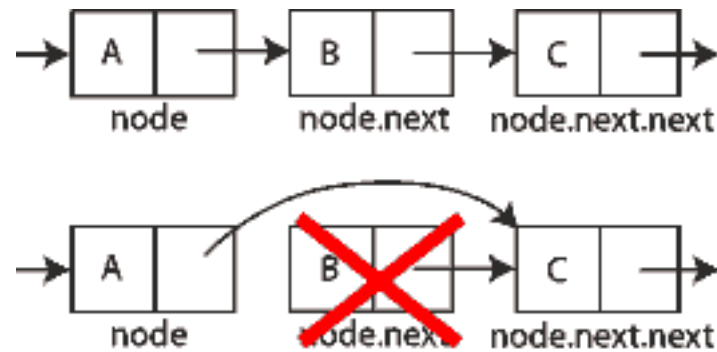


Linked List

- Insert



- Delete



class LList

```
class LList {  
    public:  
        LList();  
        void append(T v);  
        Node *find(T v);  
        void delete(T v);  
        void print();  
}  
class Node {  
    T data;  
    Node *next;  
    public:  
        Node(T v, Node *next);  
        T getData();  
        Node *next();  
        void setNext( Node *node );  
}
```

LList: isEmpty()

- Empty



Node *head;
head == NULL;

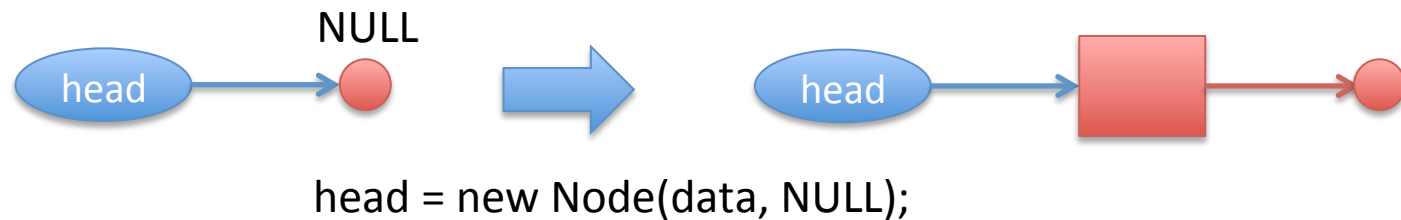
- Non-Empty



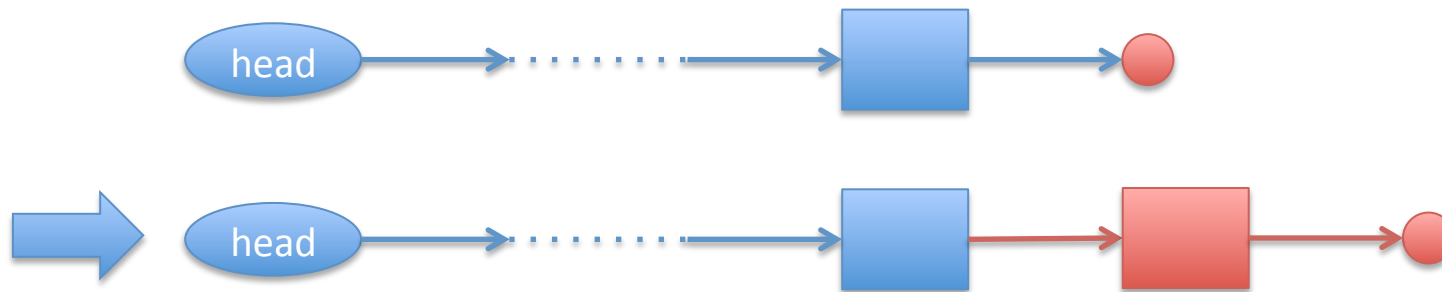
head != NULL;

LList: Append()

- Empty



- Non-Empty



find last node
`last_node->setNext(new Node(data, NULL))`

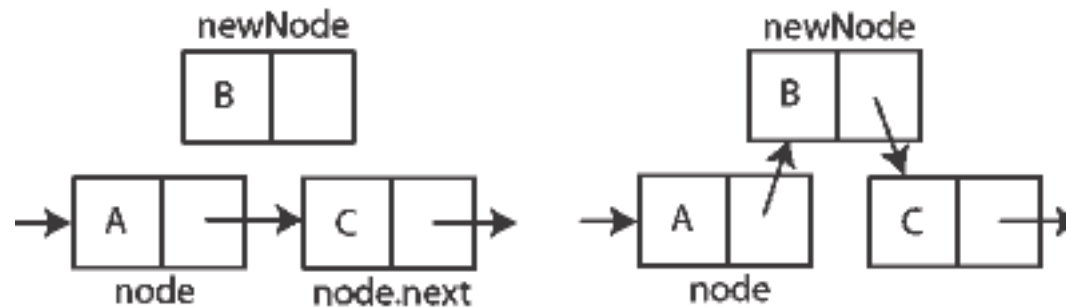
LList: Print()

```
Node *n = head;
while( n ) {
    cout << n->data << "→";
    n = n->next();
}
cout << "NULL" << endl;
```


LList: Find(data)

```
Node *n = head;
while( n ) {
    if( n->getData() == data )
        return n;
    n = n->next();
}
return NULL;
```

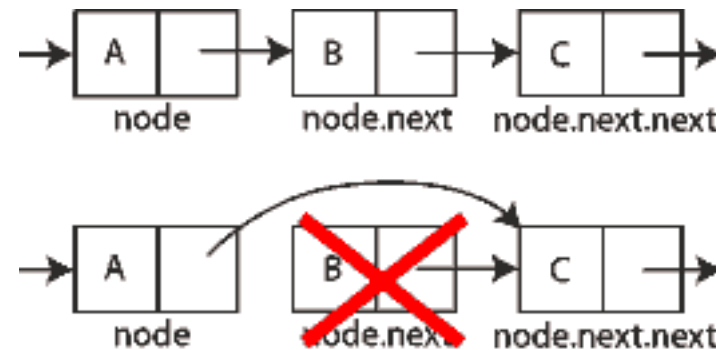
LList: Insert(n, data)



- Insert after node n
- `newnode = new Node(data, n->next());`
- `n->setNext(newnode);`
- or
- `n->setNext(new Node(data, n->next() ));`

LList: Delete(data)

- Find previous node and delete



- Find matching node, but store previous node
- `prev_node->setNext( found_node->next() );`
- `delete found_node;`