

자료 구조 실습과제 #02

조용주
ycho@smu.ac.kr

실습과제 #02 – 리스트 추상 데이터 타입(ADT)

▣ 서울시 남녀 인구 자료(출처: <https://seoul.data.go.kr>)

자치구	남자	여자
종로구	76876	82120
중구	65776	68859
용산구	118835	125810
성동구	146424	154081
광진구	173000	187109
동대문구	175322	181692
중랑구	197118	202444
성북구	214658	232398
강북구	151466	160103
도봉구	159420	167941
노원구	254188	272844
은평구	231998	252045
송파구	325080	348846

자치구	남자	여자
서대문구	153152	169800
마포구	178977	202881
양천구	224379	233402
강서구	282686	303215
구로구	215424	217064
금천구	126141	121694
영등포구	202969	204398
동작구	193718	207854
관악구	255211	254592
서초구	205282	223743
강남구	260378	283677
강동구	228044	235954

실습과제 #02 – 리스트 추상 데이터 타입(ADT)

- main() 함수에서는 인구 정보를 파일에서 읽어 리스트를 구성한 후 화면에 출력
- 리스트에서 가장 첫 번째 데이터인 종로구, 마지막 데이터인 강동구, 중간 데이터인 은평구 요소 삭제 후 출력
- 다시 리스트에 종로구, 강동구, 은평구 요소 추가(원래 위치에 삽입) 후 출력
- LstIterate() 함수를 이용해서 남자, 여자 인구 합계를 구한 후에 출력
- LstIterate() 함수를 이용해서 화면에 출력

실습과제 #02 – 리스트 추상 데이터 타입(ADT)

□ 데이터

- LstElement – PopulationData

□ 연산

연산 종류	설명
List* LstCreate(void)	리스트 생성 후 반환(head = tail = NULL)
int LstEmpty(const List* list)	리스트에 요소가 없으면 1, 있으면 0을 반환
int LstPushBack(List* list, const LstElement* e)	리스트 마지막에 요소 추가
int LstPushFront(List* list, const LstElement* e)	리스트 앞에 요소 추가

실습과제 #02 – 리스트 추상 데이터 타입(ADT)

연산 종류	설명
<code>void LstPrint(List*, PrintFunc print)</code>	리스트 요소를 화면에 출력
<code>int LstGetEntry(List* list, int pos, LstElement* e)</code>	pos 위치의 요소를 e로 반환. pos < 0 또는 pos > len이면 0(오류) 반환
<code>// e1과 e2가 동일하면 1을 반환하고 아니면 0을 반환</code> <code>typedef void (*PrintFunc)(const LstElement* e);</code>	

실습과제 #02 – 리스트 추상 데이터 타입(ADT)

연산 종류	설명
<code>int LstInsertBefore(List* list, ListNode* next, const LstElement* e)</code>	next가 리스트에 있다면, e를 next 이전 노드로 삽입
<code>int LstInsertAfter(List* list, ListNode* previous, const LstElement* e)</code>	previous가 리스트에 있다면, e를 previous 다음 노드로 삽입
<code>int LstInsert(List* list, int pos, const LstElement* e)</code>	pos 위치에 e를 삽입. pos는 0부터 시작. $pos < 0$ 또는 $pos > len - 1$ 이면 추가하지 않고 0(오류) 반환

실습과제 #02 – 리스트 추상 데이터 타입(ADT)

연산 종류	설명
<code>int LstDelete(List* list, ListNode* node)</code>	주어진 노드를 리스트에서 삭제
<code>void LstClear(List* list)</code>	리스트의 모든 요소를 삭제. 삭제 후, List의 head를 NULL로 초기화
<code>ListNode* LstFindNode(List* list, LstElement* e, CompareFunc compare)</code>	CompareFunc를 이용해서 리스트의 각 요소와 e를 비교하고, 제일 처음 발견되는 동일한 값을 가지고 있는 노드를 반환
<pre>// e1과 e2가 동일하면 1을 반환하고 아니면 0을 반환 typedef int (*CompareFunc)(const LstElement* e1, const LstElement* e2);</pre>	

실습과제 #02 – 리스트 추상 데이터 타입(ADT)

연산 종류	설명
<code>void LstIterate(List* list, IterateFunc iterate, void* p)</code>	리스트의 각 요소에 대해 IterateFunc를 호출. IterateFunc 함수를 호출할 때 세 번째 매개 변수인 p를 인자로 전달
<code>void LstDestroy(List* list)</code>	리스트 요소 삭제 및 동적 할당된 메모리 해제

```
// 리스트의 각 요소와 LstIterate()에 전달된 매개
// 변수인 p를 전달 받고 e에 대해서 처리
typedef void (*IterateFunc)(const LstElement* e,
void* p);
```


List.h

```
#include "LstElementType.h"

typedef struct _Node {
    LstElement element;
    struct _Node* next;
} ListNode;

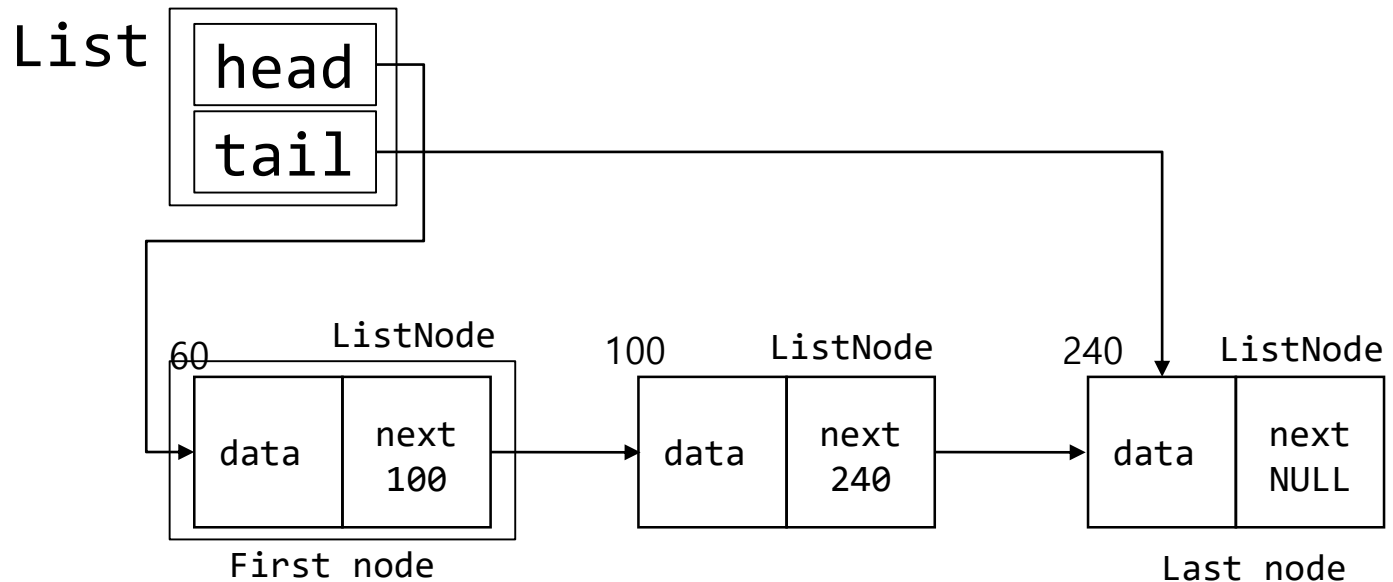
typedef struct {
    ListNode* head; // 가장 첫 번째 노드를 가리킴
    ListNode* tail; // 가장 마지막 노드를 가리킴
} List;

typedef void (*PrintFunc)(const LstElement* e);

// e1과 e2가 동일하면 1을 반환하고 아니면 0을 반환
typedef int (*CompareFunc)(const LstElement* e1,
const LstElement* e2);
```

실습과제 #02

- List.c를 구현
- main() 함수에서는 LstIterate() 함수를 이용해서 서울시 남녀 인구(남녀별 합계)와 화면에 출력하는 코드를 작성
- List.h에 있는 함수 설명 읽어볼 것



실습과제 #04

- main() 함수가 포함된 MP02.c 파일과 List.h, ReadPopulationData.c는 제공
- List.c파일은 구현할 것
- 출력 예시
 - output.txt 참조
- 제출 방법
 - 솔루션 폴더를 전체 zip파일 형태로 압축해서 제출