



C Piscine

C 12

Summary: 이 문서는 42 C Piscine 과정 C12 모듈의 학습 주제입니다.

Contents

I	Foreword	2
II	Instructions	4
III	Exercice 00 : ft_create_elem	6
IV	Exercice 01 : ft_list_push_front	7
V	Exercice 02 : ft_list_size	8
VI	Exercice 03 : ft_list_last	9
VII	Exercice 04 : ft_list_push_back	10
VIII	Exercice 05 : ft_list_push_strs	11
IX	Exercice 06 : ft_list_clear	12
X	Exercice 07 : ft_list_at	13
XI	Exercice 08 : ft_list_reverse	14
XII	Exercice 09 : ft_list_foreach	15
XIII	Exercice 10 : ft_list_foreach_if	16
XIV	Exercice 11 : ft_list_find	17
XV	Exercice 12 : ft_list_remove_if	18
XVI	Exercice 13 : ft_list_merge	19
XVII	Exercice 14 : ft_list_sort	20
XVIII	Exercice 15 : ft_list_reverse_fun	21
XIX	Exercice 16 : ft_sorted_list_insert	22
XX	Exercice 17 : ft_sorted_list_merge	23

Chapter I

Foreword

스포일러 경고 다음 페이지를 읽지 마시오.

경고했습니다.

- 스타워즈에서 다크 베이더는 루크의 아버지다.
- 유주얼 서스펙트에서 버벌이 카이제 소저다.
- 파이트 클럽에서 타일러 더튼과 나레이터는 동일 인물이다.
- 식스 센스에서 브루스 윌리스는 처음부터 죽은 상태였다.
- 디 아더스에서 저택에 살고 있던 사람들이 사실 유령이다.
- 밤비에서 밤비의 엄마는 죽는다.
- 빌리지에서 괴물은 마을 사람들이고 영화의 시대 배경은 사실 현대다.
- 해리 포터에서 덤블도어는 죽는다.
- 흑성탈출에서 그들이 도착한 건 사실은 지구였다.
- 왕좌의 게임에서 롭 스타크와 조프리 바라테온은 결혼식날 죽는다.
- 트와일라잇에서 뱀파이어는 햇빛에 닿으면 반짝반짝 빛난다.
- 스타게이트 SG-1 시즌 1의 18번째 에피소드에서 오닐과 카터는 남극에 있다.
- 다크 나이트 라이즈에서 미란다 테이트가 탈리아 알 굴이다.
- 슈퍼 마리오 브라더스에서 공주는 다른 성에 있다.

Chapter II

Instructions

- 오직 이 페이지만 참고해야 합니다. 소문은 믿지 마세요.
- 파일 제출 전에 이 문서가 변경될 수도 있으니 주의하세요!
- 파일과 디렉토리에 대해 적절한 권한을 갖고 있는지 확인하세요.
- 모든 과제물을 제출할 때는 제출 절차를 따라야 합니다.
- 제출하신 과제물은 동료들끼리 서로 확인하고 평가하게 됩니다.
- 추가로, Moulinette라는 프로그램도 과제물을 확인하고 평가합니다.
- Moulinette는 아주 꼼꼼하고 간간하게 과제물을 평가합니다. 완전히 자동화된 프로그램이기 때문에 일체의 협상은 불가능합니다. 그러니 좋지 않은 평가를 받고 실망하고 싶지 않다면 최선을 다해 철저하게 과제를 수행하세요.
- Moulinette는 그다지 마음이 너그럽지 못합니다. 표준을 따르지 않는 코드는 이해하려고 노력조차 하지 않을 겁니다. Moulinette은 `norminette` 는 프로그램으로 파일이 표준을 따랐는지 확인합니다. 그러니까 `norminette` 검사를 통과하지 못하는 과제물을 제출한다는 건 어리석은 일이겠죠?
- Exercise는 난이도에 따라 쉬운 문제에서 어려운 문제 순으로 짜여 있습니다. 앞 단계 문제의 과제물이 완벽하게 작동하지 않으면 난이도가 더 높은 문제는 아무리 잘 완료했다 하더라도 평가에 반영되지 않습니다.
- 사용이 금지된 함수를 사용하는 것은 부정 행위로 간주됩니다. 부정 행위는 -42점을 받게 되며, 받은 점수는 절대 조정이 불가능합니다.
- 프로그램을 제출해야 하는 문제의 경우 `main()` 함수만 제출하면 됩니다.
- Moulinette은 `-Wall -Wextra -Werror` 플래그를 지정하여 컴파일하며 `gcc`를 사용합니다.
- 프로그램이 컴파일되지 않으면 0점을 받게 됩니다.
- Exercise에서 정한 파일 이외의 어떠한 파일도 디렉토리에 남겨 두어서는 안 됩니다.
- 질문이 있으신가요? 오른쪽 동료에게 물어보세요. 아니면 왼쪽 동료에게 물어보세요.
- 참고 가이드는 Google / man / the Internet / ...입니다.


- 인트라넷의 포럼에서 ‘C Piscine’ 파트를 참조하거나 Slack의 Piscine 채널을 확인해보세요.
- 예시를 꼼꼼히 살펴보세요. Exercise에서 명시적으로 언급되지 않은 세부적인 사항에 대한 힌트를 얻을 수도 있습니다...
- 오딘의 힘으로, 토르의 힘으로! 열심히 고민해 보세요!!!
- 이번 과제에서는 다음 struct를 사용하게 됩니다. :

```
typedef struct          s_list
{
    struct s_list      *next;
    void               *data;
    t_list;
}
```

- 이 struct를 ft_list.h 파일에 넣은 다음 각 과제물을 제출할 때 함께 제출해야 합니다.
- Exercise 01 부터는 제공된 ft_create_elem를 사용하게 되므로 준비하세요. (ft_list.h 파일에 프로토타입을 넣어 두면 편할 수도 있습니다...).

Chapter III

Exercice 00 : ft_create_elem


	Exercise 00
	ft_create_elem
	Turn-in directory : <i>ex00/</i>
	Files to turn in : <i>ft_create_elem.c</i> , <i>ft_list.h</i>
	Allowed functions : <i>malloc</i>

- `t_list`형 새로운 요소를 생성하는 함수 `ft_create_elem`을 작성하세요.
- 이 함수는 `data`를 주어진 인자에 할당하고 `next`를 `NULL`로 할당해야 합니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
t_list *ft_create_elem(void *data);
```

Chapter IV

Exercice 01 : ft_list_push_front


	Exercise 01
ft_list_push_front	
Turn-in directory : <i>ex01/</i>	
Files to turn in : <i>ft_list_push_front.c, ft_list.h</i>	
Allowed functions : <i>ft_create_elem</i>	

- `t_list`형 새로운 요소를 목록의 맨 앞에 추가하는 함수 `ft_list_push_front`를 작성하세요.
- 이 함수는 `data`를 주어진 인자에 할당해야 합니다.
- 필요하다면 함수가 목록의 맨 앞에 있는 포인터를 업데이트할 것입니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
void ft_list_push_front(t_list **begin_list, void *data);
```


Chapter V

Exercice 02 : ft_list_size


	Exercise 02
	ft_list_size
Turn-in directory : <i>ex02/</i>	
Files to turn in : <i>ft_list_size.c, ft_list.h</i>	
Allowed functions : None	

- 목록에 있는 요소의 개수를 반환하는 함수 `ft_list_size`를 작성하세요.
- 프로토타입은 다음과 같이 선언합니다.

```
int ft_list_size(t_list *begin_list);
```

Chapter VI

Exercice 03 : ft_list_last


	Exercise 03
	ft_list_last
Turn-in directory : <i>ex03/</i>	
Files to turn in : <i>ft_list_last.c, ft_list.h</i>	
Allowed functions : None	

- 목록의 마지막 요소를 반환하는 함수 `ft_list_last` 를 작성하세요.
- 프로토타입은 다음과 같이 선언합니다.:

```
t_list *ft_list_last(t_list *begin_list);
```

Chapter VII

Exercice 04 : ft_list_push_back


	Exercise 04
ft_list_push_back	
Turn-in directory : <i>ex04/</i>	
Files to turn in : <i>ft_list_push_back.c, ft_list.h</i>	
Allowed functions : <i>ft_create_elem</i>	

- t_list형 새로운 요소를 목록의 맨 끝에 추가하는 함수 ft_list_push_back을 작성하세요.
- 이 함수는 data를 주어진 인자에 할당해야 합니다.
- 필요하다면 함수가 목록의 맨 앞에 있는 포인터를 업데이트할 것입니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
void ft_list_push_back(t_list **begin_list, void *data);
```

Chapter VIII

Exercice 05 : ft_list_push_strs


	Exercise 05
ft_list_push_strs	
Turn-in directory : <i>ex05/</i>	
Files to turn in : <i>ft_list_push_strs.c, ft_list.h</i>	
Allowed functions : <i>ft_create_elem</i>	

- `strs`에서 요소가 가리키는 모든 문자열을 포함하는 새로운 목록을 생성하는 함수 `ft_list_push_strs` 를 작성하세요.
- `size`는 `strs`의 크기입니다.
- 첫 번째 요소가 목록의 맨 끝에 있어야 합니다.
- 목록의 첫 번째 링크의 주소가 반환됩니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
t_list *ft_list_push_strs(int size, char **strs);
```

Chapter IX

Exercice 06 : ft_list_clear


	Exercise 06
	ft_list_clear
Turn-in directory : <i>ex06/</i>	
Files to turn in : <i>ft_list_clear.c, ft_list.h</i>	
Allowed functions : <i>free</i>	

- 목록에서 모든 링크를 삭제하고 해제하는 함수 `ft_list_clear` 를 작성하세요.
- `free_fct`로 각 data를 해제합니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
void ft_list_clear(t_list *begin_list, void (*free_fct)(void *));
```

Chapter X

Exercice 07 : ft_list_at


	Exercise 07
ft_list_at	
Turn-in directory : <i>ex07/</i>	
Files to turn in : <i>ft_list_at.c, ft_list.h</i>	
Allowed functions : None	

- 목록의 N번째 요소를 반환하는 함수 `ft_list_at` 를 작성합니다. 목록의 첫 번째 요소는 `nbr`이 0일 때입니다.
- 오류가 발생하면 널 포인터를 반환해야 합니다.
- 프로토타입은 다음과 같이 선언합니다.

```
t_list *ft_list_at(t_list *begin_list, unsigned int nbr);
```

Chapter XI

Exercice 08 : ft_list_reverse


	Exercise 08
ft_list_reverse	
Turn-in directory : <i>ex08/</i>	
Files to turn in : <i>ft_list_reverse.c</i>	
Allowed functions : None	

- 목록에 있는 요소의 순서를 뒤바꾸는 함수 `ft_list_reverse` 를 작성하세요. 각 요소의 값은 그대로여야 합니다.
- 이 함수에서 저희는 저희가 자체적으로 작성한 `ft_list.h`를 사용할 것입니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
void ft_list_reverse(t_list **begin_list);
```

Chapter XII

Exercice 09 : ft_list_foreach

	Exercise 09
	ft_list_foreach
	Turn-in directory : <i>ex09/</i>
	Files to turn in : <i>ft_list_foreach.c, ft_list.h</i>
	Allowed functions : None

- 인자로 주어진 함수를 목록의 각 요소에 적용하는 함수 `ft_list_foreach`를 작성하세요.
- `f`는 목록의 순서와 동일한 순서로 적용되어야 합니다.
- 프로토타입은 다음과 같이 선언합니다.:


```
void ft_list_foreach(t_list *begin_list, void (*f)(void *));
```

- `f`가 가리키는 함수는 다음과 같이 사용될 것입니다.:

```
(*f)(list_ptr->data);
```


Chapter XIII

Exercice 10 : ft_list_foreach_if

	Exercise 10
ft_list_foreach_if	
Turn-in directory : ex10/	
Files to turn in : ft_list_foreach_if.c, ft_list.h	
Allowed functions : None	

- 인자로 주어진 함수를 목록에 있는 일부 요소에 적용하는 함수 ft_list_foreach_if 를 작성하세요.
- data_ref로 cmp를 호출할 때 cmp가 0을 반환할 때만 요소에 함수를 적용합니다.
- f는 목록과 동일한 순서로 적용되어야 합니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
void ft_list_foreach_if(t_list *begin_list, void (*f)(void *), void  
*data_ref, int (*cmp)())
```

- f와 cmp가 가리키는 함수는 다음과 같이 사용될 것입니다.:


```
(*f)(list_ptr->data);  
(*cmp)(list_ptr->data, data_ref);
```



예를 들어, cmp 함수는 ft_strcmp일 수 있습니다...

Chapter XIV

Exercice 11 : ft_list_find

	Exercise 11
	ft_list_find
	Turn-in directory : ex11/
	Files to turn in : ft_list_find.c, ft_list.h
	Allowed functions : None

- data_ref로 cmp한 결과와 비교하여 첫 번째 요소의 데이터로 cmp한 결과가 0을 반환할 때 첫 번째 요소의 데이터의 주소를 반환하는 함수 ft_list_find 를 작성하세요.
- 프로토타입은 다음과 같이 선언합니다.:


```
t_list *ft_list_find(t_list *begin_list, void *data_ref, int (*cmp)());
```

- cmp가 가리키는 함수는 다음과 같이 사용될 것입니다.:

```
(*cmp)(list_ptr->data, data_ref);
```

Chapter XV

Exercice 12 : ft_list_remove_if

	Exercise 12
ft_list_remove_if	
Turn-in directory : ex12/	
Files to turn in : ft_list_remove_if.c, ft_list.h	
Allowed functions : free	

- data_ref로 cmp한 결과와 비교하여 요소의 데이터로 cmp한 결과가 0을 반환하는 모든 요소를 목록에서 지우는 함수 ft_list_remove_if 를 작성하세요.
- 지워야 하는 요소의 데이터는 free_fct를 사용하여 해제해야 합니다.
- 프로토타입은 다음과 같이 선언합니다.:


```
void ft_list_remove_if(t_list **begin_list, void *data_ref, int (*cmp)(), void (*free_fct)(void *))
```

- cmp 및 free_fct가 가리키는 함수는 다음과 같이 사용될 것입니다.:

```
(*cmp)(list_ptr->data, data_ref);  
(*free_fct)(list_ptr->data);
```

Chapter XVI

Exercice 13 : ft_list_merge


	Exercise 13
ft_list_merge	
Turn-in directory : <i>ex13/</i>	
Files to turn in : <i>ft_list_merge.c, ft_list.h</i>	
Allowed functions : None	

- begin2라는 목록의 요소를 begin1이라는 목록의 끝에 배치하는 함수 `ft_list_merge`를 작성하세요.
- 요소 생성은 허용되지 않습니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
void ft_list_merge(t_list **begin_list1, t_list *begin_list2);
```

Chapter XVII

Exercice 14 : ft_list_sort

	Exercise 14
	ft_list_sort
	Turn-in directory : ex14/
	Files to turn in : ft_list_sort.c, ft_list.h
	Allowed functions : None

- 데이터를 함수와 비교하여 두 개의 요소를 비교하는 방법으로 목록의 요소를 오름차순으로 정렬하는 함수 ft_list_sort를 작성하세요.
- 프로토타입은 다음과 같이 선언합니다.:

```
void ft_list_sort(t_list **begin_list, int (*cmp)());
```

- cmp가 가리키는 함수는 다음과 같이 사용될 것입니다.:


```
(*cmp)(list_ptr->data, list_other_ptr->data);
```



예를 들면 cmp는 ft_strcmp일 수 있습니다.

Chapter XVIII

Exercice 15 : ft_list_reverse_fun


	Exercise 15
ft_list_reverse_fun	
Turn-in directory : <i>ex15/</i>	
Files to turn in : <i>ft_list_reverse_fun.c, ft_list.h</i>	
Allowed functions : None	

- 목록에 있는 요소의 순서를 뒤바꾸는 함수 `ft_list_reverse_fun` 을 작성하세요.
- 프로토타입은 다음과 같이 선언합니다.:

```
void ft_list_reverse_fun(t_list *begin_list);
```

Chapter XIX

Exercice 16 : ft_sorted_list_insert

	Exercise 16
ft_sorted_list_insert	
Turn-in directory : ex16/	
Files to turn in : ft_sorted_list_insert.c, ft_list.h	
Allowed functions : ft_create_elem	

- 새로운 요소를 생성한 다음 그 요소를 정렬된 목록에 삽입하여 오름차순으로 정렬되도록 하는 함수 ft_sorted_list_insert를 작성하세요.
- 프로토타입은 다음과 같이 선언합니다.:


```
void ft_sorted_list_insert(t_list **begin_list, void *data, int (*cmp)());
```

- cmp가 가리키는 함수는 다음과 같이 사용될 것입니다.:

```
(*cmp)(list_ptr->data, list_other_ptr->data);
```

Chapter XX

Exercice 17 : ft_sorted_list_merge

	Exercise 17
ft_sorted_list_merge	
Turn-in directory : ex17/	
Files to turn in : ft_sorted_list_merge.c, ft_list.h	
Allowed functions : None	

- 정렬된 목록 begin2의 요소를 또다른 정렬된 목록인 begin1에 통합하는 함수 ft_sorted_list_merge를 작성하세요. 이때 begin1은 이전과 마찬가지로 오름차순으로 정렬되어야 합니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
void ft_sorted_list_merge(t_list **begin_list1, t_list *begin_list2, int (*cmp)());
```

- cmp가 가리키는 함수는 다음과 같이 사용될 것입니다.:

```
(*cmp)(list_ptr->data, list_other_ptr->data);
```