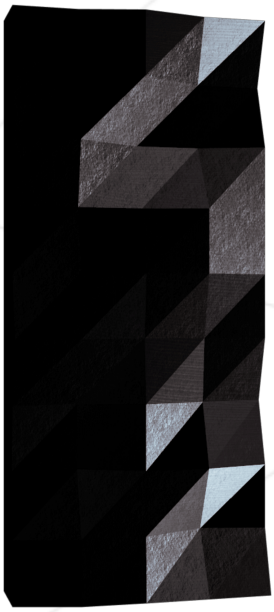

C Piscine

C 08

Summary: 이 문서에서는 42 C Piscine 과정의 모듈 C 08에서 학습하는 주제를 다룹니다.

Contents

I	**Instructions**	**2**
II	**Foreword**	**4**
III	**Exercise 00 : ft.h**	**5**
IV	**Exercise 01 : ft_boolean.h**	**6**
V	**Exercise 02 : ft_abs.h**	**8**
VI	**Exercise 03 : ft_point.h**	**9**
VII	**Exercise 04 : ft_strs_to_tab**	**10**
VIII	**Exercise 05 : ft_show_tab**	**12**

Chapter I

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 깐깐하게 과제물을 평가합니다. 완전히 자동화된 프로그램이기때문에 일체의 협상은 불가능합니다. 그러니 좋지 않은 평가를 받고 실망하고 싶지 않다면 최선을 다해 철저하게 과제를 수행하세요.
- Moulinette는 그다지 마음이 너그럽지 못하답니다. 표준을 따르지 않는 코드는 이해하려고 노력조차 하지 않을 겁니다. Moulinette는 norminette라는 프로그램으로 파일이 표준을 따랐는지 확인합니다. 그러니까 Norminette의 확인을 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 `-42`점을 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 `0`점을 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 확인해 보세요.
- 예시를 꼼꼼히 살펴보세요. Exercise에서 명시적으로 언급되지 않은 세부적인 사항에 대한 힌트를 얻을 수도 있습니다...
- 오딘의 힘으로, 토르의 힘으로! 열심히 고민해 보세요!!!

Chapter II

Foreword

오리너구리에 대해 위키백과에는 다음과 같이 설명되어 있습니다 :

오리너구리(학명: Ornithorhynchus anatinus) 또는 오리주둥이는 오스트레일리아와 태즈메이니아 섬 토종의 반수서성 단공류 포유류의 일종이다. 가시두더지 4종과 함께 현존하는 다섯뿐인 단공류이며, 가장 원시적인 포유류인 동시에 난생의 번식 방법을 택하고 있는 극소수의 포유류 중 하나이다. 또한 오리너구리속, 오리너구리과에서도 유일하게 현재까지 명맥을 잇고 있는 종이자 모식종으로, 같은 오리너구리과에 속하는 것으로 여겨지는 다른 여러 종은 모두 화석으로만 발견된다.

오리를 닮은 부리, 비버를 닮은 꼬리, 수달을 닮은 발을 가진 다소 우스꽝스러운 외모에 알을 낳는 생태까지 겹쳐, 유럽 박물학자들은 오리너구리를 처음 보았을 때 당황했고 일부는 조작된 가짜라고 보기도 했다. 포유류 가운데서는 매우 드물게도 독성 물질을 지니고 있는데, 수컷 오리너구리의 뒷발 며느리발톱과 연결된 독샘을 통해서 분출되며, 인간이 여기에 베일 경우 찌르는 듯한 극심한 고통을 수반한다. 이러한 오리너구리의 독특한 특징 탓에 오리너구리는 진화생물학의 중요 연구 주제이며 오스트레일리아의 대표적인 상징이 되었다. 국가 행사의 마스코트로 등장하기도 했으며 20센트 동전 뒷면에도 새겨져 있다. 오스트레일리아 뉴사우스웨일즈주의 상징동물이기도 하다.

20세기 초까지 모피를 얻고자 남획당했으나, 현재는 모든 서식지에서 법적으로 보호받고 있다. 인공번식으로 개체 수를 불리는 노력을 기울였음에도 큰 효과는 보지 못했고 환경 오염이 오리너구리 생존에 위협이 되고 있기는 하지만, 아직까지 개체 수의 폭락이나 눈에 띄는 위협은 보이지 않는다.

이 과제는 오리너구리와는 손톱만큼도 관련이 없습니다.

Chapter III

Exercise 00 : ft.h

	Exercise 00
	ft.h
Turn-in directory : $ex00/$	
Files to turn in : `ft.h`	
Allowed functions : `None`	

- `ft.h` 파일을 작성하세요.
- 이 파일은 다음과 같이 함수의 모든 프로토타입을 포함합니다. :

```
void    ft_putchar(char c);
void    ft_swap(int *a, int *b);
void    ft_putstr(char *str);
int     ft_strlen(char *str);
int     ft_strcmp(char *s1, char *s2);
```

Chapter IV

Exercise 01 : ft_boolean.h

	Exercise 01
	ft_boolean.h
Turn-in directory : $ex01/$	
Files to turn in : `ft_boolean.h`	
Allowed functions : `None`	

- `ft_boolean.h` 파일을 작성하세요. 이 파일은 다음 main을 적절히 컴파일하여 실행합니다. :

```
#include "ft_boolean.h"

void        ft_putstr(char *str)
{
        while (*str)
                write(1, str++, 1);
}

t_bool        ft_is_even(int nbr)
{
        return ((EVEN(nbr)) ? TRUE : FALSE);
}

int                main(int argc, char **argv)
{
        (void)argv;
        if (ft_is_even(argc - 1) == TRUE)
                ft_putstr(EVEN_MSG);
        else
                ft_putstr(ODD_MSG);
        return (SUCCESS);
}
```

- 이 프로그램은 다음을 출력해야 합니다.

```
I have an even number of arguments.
```

- 또는

```
I have an odd number of arguments.
```

- 필요한 경우 끝에 줄바꿈이 추가됩니다.

Norminette은 *-R CheckDefine* 플래그를 사용해 실행되어야 합니다.
Moulinette도 이 플래그를 사용합니다.

Chapter V

Exercise 02 : ft_abs.h

	Exercise 02
	ft_abs.h
Turn-in directory : $ex02/$	
Files to turn in : `ft_abs.h`	
Allowed functions : `None`	

- 인자를 절대값으로 교체하는 매크로 ABS를 작성하세요. :

```
#define ABS(Value)
```

Norminette은 *-R CheckDefine* 플래그를 사용해 실행되어야 합니다.
Moulinette도 이 플래그를 사용합니다.

Chapter VI

Exercise 03 : ft_point.h

	Exercise 03
	ft_point.h
Turn-in directory : $ex03/$	
Files to turn in : `ft_point.h`	
Allowed functions : `None`	

- 다음 main 함수를 컴파일할 파일 `ft_point.h`를 작성하세요. :

```
#include "ft_point.h"

void        set_point(t_point *point)
{
        point->x = 42;
        point->y = 21;
}

int                main(void)
{
        t_point                point;

        set_point(&point);
        return (0);
}
```

Chapter VII

Exercise 04 : ft_strs_to_tab

	Exercise 04
ft_strs_to_tab	
Turn-in directory : $ex04/$	
Files to turn in : `ft_strs_to_tab.c`	
Allowed functions : `malloc, free`	

- 문자열 배열을 인자로 취해 이 배열의 크기를 출력하는 함수를 작성하세요.
- 프로토타입은 다음과 같이 작성합니다 :

```
struct s_stock_str *ft_strs_to_tab(int ac, char **av);
```

- 이 함수는 av의 각 항을 구조로 바꿀 것입니다.
- 구조는 저희가 제공해드릴 파일 `ft_stock_str.h`에 다음과 같이 정의되어 있습니다. :

```
typedef struct s_stock_str
{
  int  size;
  char *str;
  char *copy;
}            t_stock_str;
```

  - `size`는 문자열의 길이입니다;
  - `str`은 문자열(string)입니다;
  - `copy`는 문자열의 사본(copy)입니다;
- av의 순서를 지켜야 합니다.

- 반환된 배열이 메모리를 할당하고 마지막 항의 `str`이 0으로 선언되면 배열의 끝을 나타내게 됩니다.
- 오류가 발생하면 널(null) 포인터가 출력되어야 합니다.
- 여러분이 작성한 함수를 저희의 `ft_show_tab`(다음 Exerciese)로 테스트할 것입니다. 이 함수로 작동되도록 만들어 보세요!

Chapter VIII

Exercise 05 : ft_show_tab

	Exercise 05
	ft_show_tab
Turn-in directory : $ex05/$	
Files to turn in : `ft_show_tab.c`	
Allowed functions : `write`	

- 전 장에서 작성한 함수에 의해 생성된 배열의 요소를 표시하는 함수를 작성하세요.
- 프로토타입은 다음과 같이 선언해야 합니다. :

```
void ft_show_tab(struct s_stock_str *par);
```

- 구조는 전 장의 연습 문제와 같으며, `ft_stock_str.h` 파일에 정의되어 있습니다.
- 각 요소에 대해 다음과 같이 출력하겠습니다. :
  - 문자열 + '\n'
  - 크기 + '\n'
  - 문자열의 사본(수정되었을 수 있음) + '\n'
- 저희 `ft_strs_to_tab` (전 장의 연습 문제)으로 테스트할 것입니다. 이 함수에 따라 작동되도록 만들어 보세요!