



C Piscine

C 05

*Summary:* 이 문서에서는 42 C Piscine 과정의 모듈 C 05에서 학습하는 주제를 다룹니다.

# Contents

I	Instructions	2
II	Foreword	4
III	Exercise 00 : ft_iterative_factorial	6
IV	Exercise 01 : ft_recursive_factorial	7
V	Exercise 02 : ft_iterative_power	8
VI	Exercise 03 : ft_recursive_power	9
VII	Exercise 04 : ft_fibonacci	10
VIII	Exercise 05 : ft_sqrt	11
IX	Exercise 06 : ft_is_prime	12
X	Exercise 07 : ft_find_next_prime	13
XI	Exercise 08 : The Ten Queens	14

# 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 프로그램이기 때문에 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 -42점을 받게 되며, 받은 점수는 절대 조정이 불가능합니다.
- 프로그램을 제출해야 하는 문제의 경우 main() 함수만 제출하면 됩니다.
- Moulinette은 -Wall -Wextra -Werror 플래그를 지정하여 컴파일하며 gcc를 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 확인해 보세요.
- 예시를 꼼꼼히 살펴보세요. Exercise에서 명시적으로 언급되지 않은 세부적인 사항에 대한 힌트를 얻을 수도 있습니다...
- 오딘의 힘으로, 토르의 힘으로! 열심히 고민해 보세요!!!



Norminette은 `-R CheckForbiddenSourceHeader` 플래그를 사용해 실행되어야 합니다. Moulinette도 이 플래그를 사용합니다.

# Chapter II

## Foreword

해리포터에는 다음과 같은 노래가 나옵니다:

당신은 내가 예쁘다고 생각하지 않을지도 몰라요.  
그러나 눈에 보이는 것으로 판단하지 마세요.  
나보다 더 멋진 모자를 찾을 수 있다면  
난 나를 그냥 먹어버릴 거예요.

까만 중절모를 써도 좋고,  
맵시 있고 높은 신사모자도 괜찮아요,  
난 호그와트의 기숙사를 배정하는 마법모자예요.  
나는 모든 모자들을 다 덮어 버릴 수 있어요.

당신 머리 속에 있는 모든 것을  
기숙사를 배정하는 마법모자는 모두 볼 수 있어요.  
나를 써봐요. 그러면 말해 줄게요.  
당신이 어디로 가야 하는지

당신은 그리핀도르에 속할지도 몰라요.  
정말 용감한 사람들이 모이는 곳이죠,  
용기와 대담성 그리고 기사도 정신은  
그리핀도르의 특징이죠.

당신은 후플루프에 속할지도 몰라요.  
그곳 사람들은 정의롭고 성실하죠,  
참을성 있는 후플루프 사람들은 진실하며  
노고를 마다하지 않아요.

현명하고 사려 깊은 래번클로에서는,  
지혜와 지식이 있는 사람들이  
서로 어울릴 수 있어요.

또 슬리데린에서는  
진정한 친구를 만나게 될 거예요.  
그곳의 재간꾼들은 목적 달성을 위해  
수단과 방법을 가리지 않아요.

그러니 날 써보세요! 겁내지 말고요!  
그리고 당황하지 말아요!  
마음 푹 놓고 내 손에 맡겨요.(내겐 손은 없지만요.)  
나는 생각하는 모자니까요!

안타깝게도 이번 과제는 해리 포터와는 아무 관련이 없습니다. 참 아쉽네요. 연습 문제  
가 마법으로 그냥 술술 풀리면 얼마나 좋을까요!

# Chapter III

## Exercise 00 : ft\_iterative\_factorial

	Exercise 00
	ft_iterative_factorial
Turn-in directory :	ex00/
Files to turn in :	ft_iterative_factorial.c
Allowed functions :	None

- 숫자를 반환하는 반복 함수를 작성하세요. 이 숫자는 매개변수로 주어지는 숫자에 따른 계승 연산의 결과입니다.
- 인자가 유효하지 않을 경우 함수는 0을 반환해야 합니다.
- 오버플로우는 처리해서는 안 됩니다. 함수의 반환값이 정의되지 않은 결과가 나옵니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
int ft_iterative_factorial(int nb);
```

# Chapter IV

## Exercise 01 : ft\_recursive\_factorial

	Exercise 01
	ft_recursive_factorial
Turn-in directory :	ex01/
Files to turn in :	ft_recursive_factorial.c
Allowed functions :	None

- 매개변수로 주어지는 숫자의 계승을 반환하는 재귀함수를 작성하세요.
- 인자가 유효하지 않을 경우 함수는 0을 반환해야 합니다.
- 오버플로우는 처리해서는 안 됩니다. 함수의 반환값이 정의되지 않은 결과가 나옵니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
int ft_recursive_factorial(int nb);
```

# Chapter V

## Exercise 02 : ft\_iterative\_power

	Exercise 02
	ft_iterative_power
Turn-in directory :	ex02/
Files to turn in :	ft_iterative_power.c
Allowed functions :	None

- 어떤 수의 거듭제곱 값을 반환하는 반복함수를 작성하세요. n이 0보다 작으면 0이 반환됩니다. 오버플로우는 처리해서는 안 됩니다.
- 0의 0제곱은 1을 반환하기로 합니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
int ft_iterative_power(int nb, int power);
```

# Chapter VI

## Exercise 03 : ft\_recursive\_power

	Exercise 03
	ft_recursive_power
Turn-in directory :	ex03/
Files to turn in :	ft_recursive_power.c
Allowed functions :	None

- 어떤 숫자의 거듭제곱 값을 반환하는 함수를 작성하세요.
- 오버플로우는 처리해서는 안 됩니다. 함수의 반환값이 정의되지 않은 결과가 나옵니다.
- 0의 0제곱은 1을 반환하기로 합니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
int ft_recursive_power(int nb, int power);
```

# Chapter VII

## Exercise 04 : ft\_fibonacci

	Exercise 04
	ft_fibonacci
Turn-in directory :	ex04/
Files to turn in :	ft_fibonacci.c
Allowed functions :	None

- 피보나치 수열의 n번째 항을 반환하는 함수 `ft_fibonacci`를 작성합니다. 이때 첫번째 항은 인덱스 0으로 합니다. 이 피보나치 수열은 0, 1, 1, 2와 같이 시작되는 것으로 하겠습니다.
- 오버플로우는 처리해서는 안 됩니다. 함수의 반환값이 정의되지 않은 결과가 나옵니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
int ft_fibonacci(int index);
```

- 당연히, `ft_fibonacci`는 재귀함수여야 합니다.
- 인덱스가 0보다 작으면 함수는 -1을 반환해야 합니다.

# Chapter VIII

## Exercise 05 : ft\_sqrt

	Exercise 05
	ft_sqrt
Turn-in directory :	ex05/
Files to turn in :	ft_sqrt.c
Allowed functions :	None

- 어떤 수의 제곱근이 존재하면 그 제곱근을, 제곱근이 무리수일 경우에는 0을 반환하는 함수를 작성하세요.
- 프로토타입은 다음과 같이 선언합니다.:

```
int ft_sqrt(int nb);
```

# Chapter IX

## Exercise 06 : ft\_is\_prime

	Exercise 06
	ft_is_prime
Turn-in directory :	ex06/
Files to turn in :	ft_is_prime.c
Allowed functions :	None

- 매개변수로 주어진 수가 소수이면 1을, 그렇지 않으면 0을 반환하는 함수를 작성하세요.
- 프로토타입은 다음과 같이 선언합니다.:

```
int ft_is_prime(int nb);
```



0과 1은 소수가 아닙니다.

# Chapter X

## Exercise 07 : ft\_find\_next\_prime

	Exercise 07
	ft_find_next_prime
Turn-in directory :	ex07/
Files to turn in :	ft_find_next_prime.c
Allowed functions :	None

- 인자로 주어진 수보다 크거나 같은 다음 소수를 반환하는 함수를 작성하세요.
- 프로토타입은 다음과 같이 선언합니다.:

```
int ft_find_next_prime(int nb);
```

# Chapter XI

## Exercise 08 : The Ten Queens

	Exercise 08
	The Ten Queens
Turn-in directory :	<i>ex08/</i>
Files to turn in :	<i>ft_ten_queens_puzzle.c</i>
Allowed functions :	<i>write</i>

- 10X10 체스판에서 퀸 10개가 갈 수 있는 모든 위치를 표시하고 모든 가능한 수의 숫자를 반환하는 함수를 작성하세요. 단, 각 퀸은 1번 움직여서 다른 퀸을 잡을 수 없습니다.
- 이 문제를 풀기 위해서는 recursive를 활용해야 합니다.
- 프로토타입은 다음과 같이 선언합니다.:

```
int ft_ten_queens_puzzle(void);
```

- 다음과 같이 표시되어야 합니다:

```
$>./a.out | cat -e  
0257948136$  
0258693147$  
...  
4605713829$  
4609582731$  
...  
9742051863$  
$>
```

- 순서는 왼쪽에서 오른쪽입니다. 첫번째 자릿수는 1열 첫번째 퀸의 위치를 나타냅니다 (인덱스는 0부터 시작). n번째 자릿수는 n번째 열의 n번째 퀸의 위치를 나타냅니다.
- 반환값은 표시된 모든 방법의 총 개수여야 합니다.