

Dash – problem solving

Summary: this document is the subject for the dash @ 42Seoul.

Contents

1 Foreword

2 Objective

3 Instructions

4 Exercise 00 : forward_print

5 Exercise 01 : backward_print

6 Exercise 02 : memoization

Chapter 1

Foreword

이 프로젝트는 문제를 해결하는 것에 초점을 맞추었으며, 문제를 다양하게 바라보는 관점을 기르는 것을 목표로 합니다.

Chapter 2

Objective

문제해결


Chapter 3

Instructions

- 빌드 옵션은 `c99 -Wall -Wextra -Werror` 을 넣어주세요.
- 전역변수 사용을 강권합니다.
- 각 문제 별 제한 사항이 있으니 RedBox를 잘 읽어 주세요.
- `norm`은 지키지 않습니다.
- `scanf`를 사용할 수 있습니다.

Chapter 4

Exercise 00 : forward_print

	Exercise 00
forward_print	
Turn-in directory : ex00/	
Files to turn in : forward_print.c	
Allowed function : write	



반복문과 변수 선언을 금지합니다.

매개변수로 주어진 문자열을 정방향으로 출력 하시오.


void forward_print(char *msg);

허용함수: write

hint: recursion..?

Chapter 5

Exercise 01 : backward_print

	Exercise 01
backward_print	
Turn-in directory : ex01/	
Files to turn in : backward_print.c	
Allowed function : write	



반복문과 변수 선언을 금지합니다.


매개변수로 주어진 문자열을 역방향으로 출력하시오.

```
void backward_print(char *msg);
```

허용함수: write

Chapter 6

Exercise 02 : memoization

	Exercise 02
memoization	
Turn-in directory : ex02/	
Files to turn in : memoization.c	
Allowed function : write	



반복문 사용을 금지합니다.

이 챕터에서는 Memoization 기술을 배워 볼 수 있습니다.

이 문제의 이름은 왜 Dynamic Programming인가..?

:thinking_face: 리차드 벨만 사진

(메모 하는 기술인데, 그냥 멋있어서 다이나믹 프로그래밍이라고 했다고 한다...)

<https://stackoverflow.com/questions/35959100/explanation-on-fibonacci-recursion>

N번째 fibonacci수는 $Fib(N) = Fib(N - 1) + Fib(N - 2)$ 로 구할 수 있다.

위의 내용을 보면 알겠지만, 중복되는 과정이 존재한다. 이를 메모하면서 중복연산을 줄이는 방법이 메모기법이다.

N번째 fibonacci 수를 구할 수 있는 함수를 작성하시오.

```
long long fib(int index);
```

$1 \leq N \leq 90$