

C Programming (W4)



Welcome!!

Please check attendance individually.
(Mobile App)

Things to do today

01 | Codyssey C1-P1 : Evaluation

02 | Lecture Notes (Ch.1 ~3)
Standard input (W3) - CLI, Language, Debugger

03 | Let's practice Codyssey (C1-P2)

<https://www.aiexpo.co.kr/en/>

Change computer's time to korean time.

Select a project in Codyssey

002, 003 – Ch1.~3

Step for homework

- 01** | Read a problem in **Codysey** (Specifically Implementation Task, Constraints)
- 02** | Implement the solution of a problem in **VSC**
Run and make sure your solution satisfies implementation task & constraints
- 03** | Upload your solution into your **github**.
- 04** | Request the evaluation (30 minutes discussion) through **Codysey**
Then participate the evaluation of classmate's solution (30 minutes discussion)

Homework 진행 순서

01

Codyssey 문제 파악(특히 수행문제, 제약사항)

02

VSC 에서 Coding 구현
Terminal에서 실행 확인 및 Codyssey에 요청하는 수행문제, 제약사항 만족하는지 확인

03

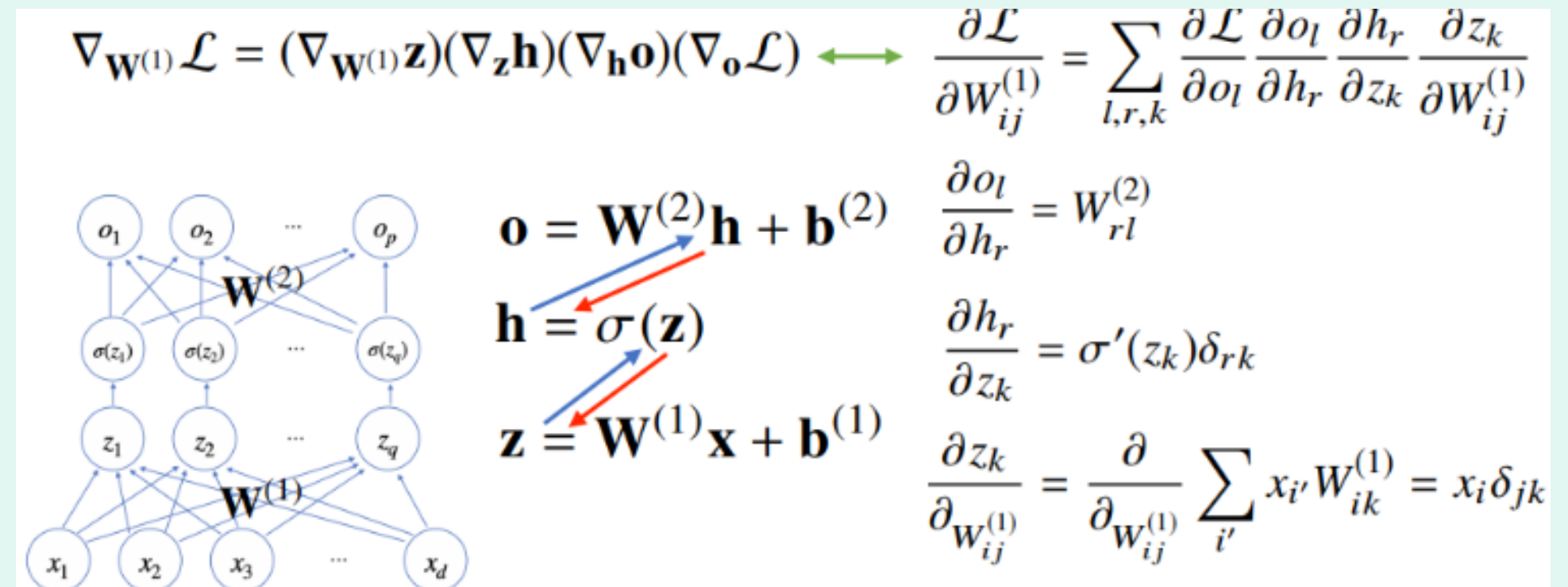
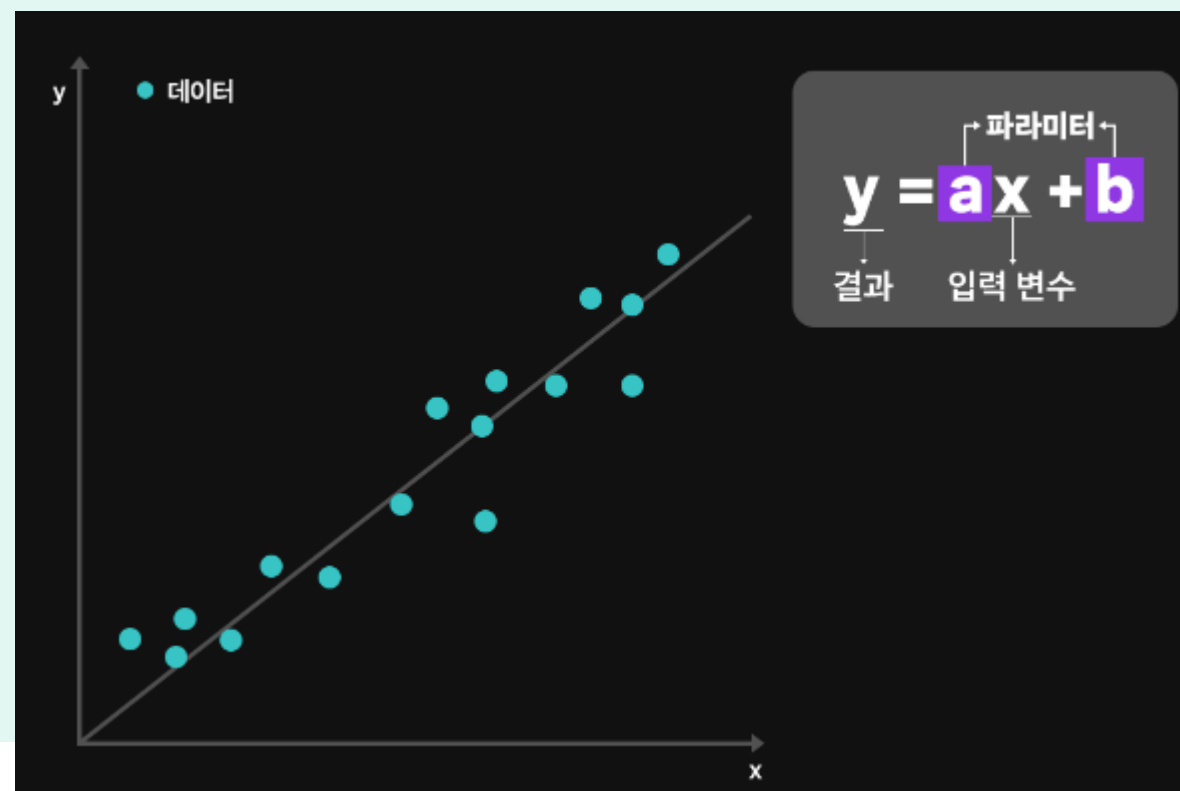
Github에 구현한 Source code 저장(Upload)

04

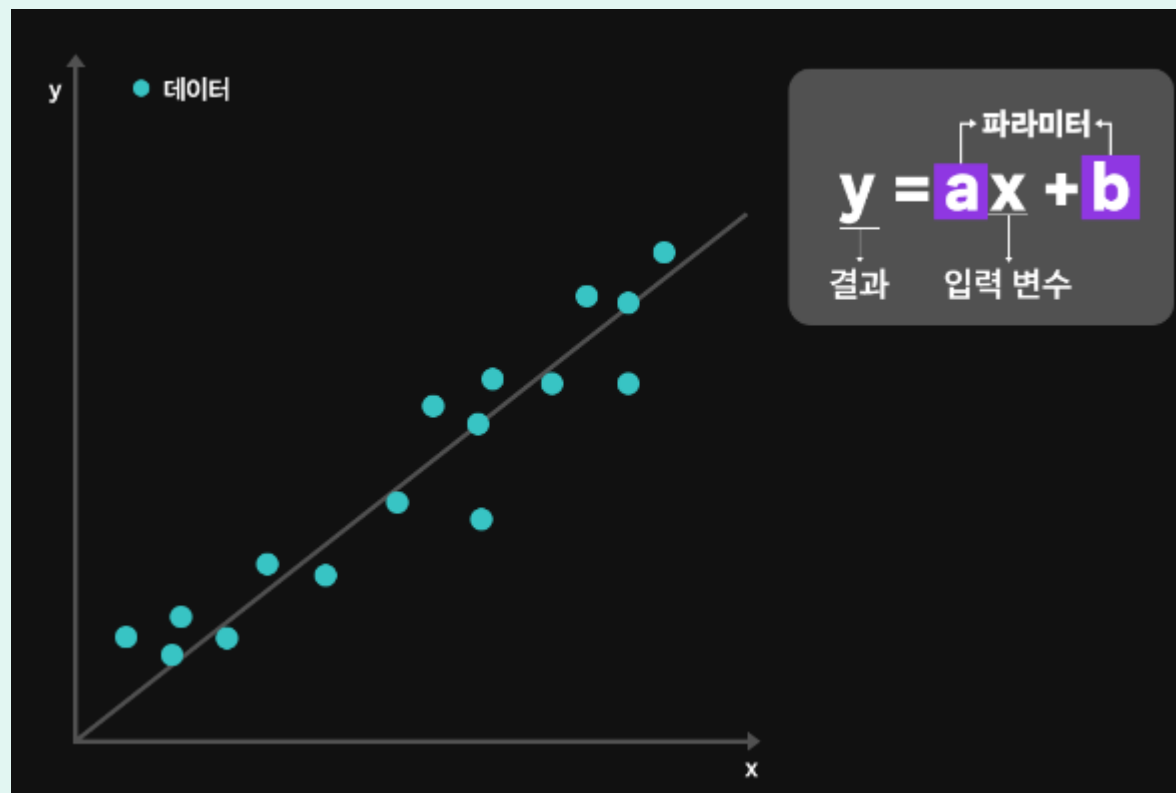
Codyssey에서 동료 “평가요청” (평가자에게 설명)
역할을 바꿔서 평가 요청한 동료의 Source code “평가참여”

AI (Artificial Intelligence), especially Deep Learning, can be viewed as a process of learning **a mathematical function (model) with numerous parameters.**

An AI model is a mathematical function that describes the complex relationship between input and output, and can be said to be **a process of learning data patterns** by optimizing numerous parameters. However, AI is not a simple "formula"; it is a process of finding the optimal function based on data.



AI(인공지능), 특히 딥러닝은 수많은 매개변수를 갖는 수학적 함수(모델)를 학습하는 과정으로 볼 수 있습니다. AI 모델은 입력과 출력 간의 복잡한 관계를 기술하는 수학적 함수이며, 수많은 매개변수를 최적화하여 데이터 패턴을 학습하는 과정이라고 할 수 있습니다. 그러나 AI는 단순한 "공식"이 아니라 데이터를 기반으로 최적의 함수를 찾는 과정입니다.



$$\nabla_{\mathbf{W}^{(1)}} \mathcal{L} = (\nabla_{\mathbf{W}^{(1)}} \mathbf{z})(\nabla_{\mathbf{z}} \mathbf{h})(\nabla_{\mathbf{h}} \mathbf{o})(\nabla_{\mathbf{o}} \mathcal{L}) \longleftrightarrow \frac{\partial \mathcal{L}}{\partial W_{ij}^{(1)}} = \sum_{l,r,k} \frac{\partial \mathcal{L}}{\partial o_l} \frac{\partial o_l}{\partial h_r} \frac{\partial h_r}{\partial z_k} \frac{\partial z_k}{\partial W_{ij}^{(1)}}$$

$$\mathbf{o} = \mathbf{W}^{(2)} \mathbf{h} + \mathbf{b}^{(2)} \quad \frac{\partial o_l}{\partial h_r} = W_{rl}^{(2)}$$

$$\mathbf{h} = \sigma(\mathbf{z}) \quad \frac{\partial h_r}{\partial z_k} = \sigma'(z_k) \delta_{rk}$$

$$\mathbf{z} = \mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)} \quad \frac{\partial z_k}{\partial W_{ij}^{(1)}} = \frac{\partial}{\partial W_{ij}^{(1)}} \sum_{i'} x_{i'} W_{ik}^{(1)} = x_i \delta_{jk}$$

집단지성 → 20년간 1인당 정보량 1,000만 배 증가



In a PC, a **shell** is a user interface that allows you to interact with the operating system. It can be **command-line-based** (CLI) or **graphical** (GUI):

1. Command-Line Shell (CLI) – This is a text-based interface where users enter commands to perform tasks. Examples:

- **Bash** (Linux, macOS)
- **PowerShell** (Windows)
- **Command Prompt (cmd.exe)** (Windows)
- **Zsh, Fish** (Unix-like systems)

2. Graphical Shell (GUI) – This provides a visual interface with icons, windows, and menus. Examples:

- **Windows Explorer** (Windows shell)
- **GNOME/KDE/Xfce** (Linux desktop environments)
- **macOS Finder** (Mac shell)

Codyyssey 구조

Subproject	Learning Course	Problem	Week	Essential
Step 1: Audition for Cody Enter	Course 1: Wasteland with Value, Magratea Standard Input/Output	01 Question 1 Introduce myself	1	O
		02 Question 2 Project Kick-Off!	2	O
		03 Question 3 Children who became Milliways candidates	3	O
		04 Question 4 Who will choose the one who will hold the key to destiny?	4	O
		05 Question 5 The hand that determines destiny		-
		06 Question 6 The one chosen to embark on a journey of adventure		-
		07 Question 7 So that no one is forgotten		-
		08 Question 8 Certification test		-
		09 Question 9 From the cradle to the stage		-
Step 2: 8-Step Training Program	Course 2: Sprouts Blooming in the Wasteland Multidimensional Arrays	01 Question 1 8-step training program	5	O
		02 Question 2 Dumbass, the problem is physical strength!	6	O
		03 Question 3 My Basic Workout Routine	6	O
		04 Question 4 A Flexitarian Who Dreams of Being Vegan		-
		05 Question 5 The Conflict Between Hunger and Fullness		-
		06 Question 6 Escaping Obsessive-Compulsive Disorder		-
		07 Question 7 Workout Routine for K-POP Idols		-
		08 Question 8 A Talent with Both Civil and Military Skills		-
		09 Question 9 Starbucks' Jazz Album		-
	Course 3: Wounds Heal in the Gardener's Hands Structures	01 Question 1 A Fight with Yourself	7	O
		02 Question 2 Facing Trauma	8	O
		03 Question 3 There is No Way to Escape Your Own Ghosts	8	O
		04 Question 4 Friends Who Go Together to Go Far		-
		05 Question 5 A Day Organized in 3 Questions		-
		06 Question 6 Peak-End Law		-
		07 Question 7 Between Coolness and Passion		-
		08 Question 8 Standing at the Starting Point of a Race		-
		09 Question 9 Anti-Hunger Games		-
	Course 4: Temperature of Language Pointers	01 Question 1 How Good is My Korean?	9	O
		02 Question 2 Consonants and Vowels	9	O
		03 Question 3 Basic Grammar Learned through Puzzles	9	O
		04 Question 4 End-of-the-Sentence Linking		-
		05 Question 5 Dictionary Game		-
		06 Question 6 Dialects and Standard Language		-
		07 Question 7 Dictation		-
		08 Question 8 Measurement is the Key to Improvement		-
	Course 5: Temperature of Sound	01 Question 1 Facing the Present	10	O
		02 Question 2 Finding My Voice	10	O
		03 Question 3 Those Who Realized the Principle	10	O
		04 Question 4 Singing a Song		-

Requirements list (C1-P1)

1. Development Environment & Standards

- Use a cross-platform development tool (e.g., VS Code, CLion, Code::Blocks).
- Follow the ANSI C standard syntax.
- Use GCC 9.x compiler.
- Only use the C standard library (stdio.h, stdlib.h, etc.).
- Follow ANSI C coding style (proper indentation, comments, and readable code).

2. Project Structure

- The project folder must be named Magrathea.
- All source code must include comments for clarity.
- The source code must compile successfully without errors or warnings.
- Code must not contain unnecessary or unrelated parts.
- All logic must be implemented within main() (no functions outside main())

3. Problem specific requirements

- Print Arthur and the team members' basic information to the console, following the specified format:
- Each sentence in the "Introduction" section should break onto a new line after a period (.).



Common to all
problems

Requirements list (C1-P2)

1. User Input
 - Prompt the user to enter the **current date** in the "yyyy-mm-dd" format.
 - Prompt the user to enter their **name**.
2. Processing the Input
 - Display the message **“The input has been processed successfully.”**
 - Ensure the entered values (name and date) are incorporated into the splash screen output.

3. Splash Screen Output

- After processing, display the following splash screen format

```
+++++
[Magrathea ver 0.1]
Magrathea, where a shining planet is created in a wasteland with no grass,
a place where unseen potential is discovered and gems are polished by the hands of experts,
Welcome to Magrathea.
+++++
[User]: [name]           [Execution Time]: [date]
=====
```

4. Bonus 1: Delay Before Display

- After the input has been processed, **clear the screen after 3 seconds** and then display the splash screen.
- Display a right-angled triangle and an inverted right-angled triangle made of * characters on the left and right edges of the splash screen.



Requirements list (C1-P1)

개발 환경

크로스 플랫폼을 지원하는 개발 도구를 사용한다.
 이 과정에 사용하는 C 언어 문법은 ANSI C 표준을 따른다.
 ANSI C 표준을 준수하는 C 언어 컴파일러로 gcc 9.x를 사용한다.
 C언어 표준 라이브러리만 사용한다.
 ANSI C 표준 코딩 스타일을 따른다. (https://www.gnu.org/prep/standards/html_node/Writing-C.html)

공통 제약 사항

크로스 플랫폼을 지원하는 개발 도구를 사용한다.
 이 과정에 사용하는 C 언어 문법은 ANSI C 표준을 따른다.
 ANSI C 표준을 준수하는 C 언어 컴파일러로 gcc 9.x를 사용한다.
 C언어 표준 라이브러리만 사용한다.
 ANSI C 표준 코딩 스타일을 따른다.
 프로젝트 폴더의 이름은 Magrathea로 한다.
 작성한 소스 코드는 반드시 주석을 포함해야 한다.
 소스 코드는 어떠한 에러와 경고 없이 성공적으로 컴파일 되어야 한다.
 소스 코드에 문제 해결과 관련 없는 코드는 포함하지 않아야 한다.
 모든 코드는 main() 함수 내에 작성한다.
 솔루션이 다루는 C 언어의 범위는 정수와 실수, 상수와 변수, 데이터 입력/출력, 연산자, 조건문(선택문), 반복문, 1 차원 배열, 문자 및 문자열이다.

공통 사항

[문제 요구사항]

1. GCC 9.x 이상 사용

- gcc -version 명령어를 사용하여 컴파일러 버전 확인

2. 프로젝트 멤버 정보 출력

- 주어진 형식에 맞춰 팀원 정보를 printf() 함수 사용하여 출력
- 출력 형식 유지 (줄바꿈, 구분선, 정렬 고려)

[제약 사항]

- 모든 코드는 main() 함수 내에 작성한다.
- 제출할 파일의 이름은 about_me.c다.
- 표준 출력 함수만 사용해야 한다.
- 아서와 팀원의 기초 정보는 다음과 같다 (도표 확인)

Requirements list (C1-P2)

1. printf()를 사용하여 마그라테아 스플래시 화면에 사용할 인상적인 문구를 출력한다.
2. 사용자로부터 날짜, 이름을 입력 받는다.
 - 날짜 형식: "yyyy-mm-dd"
 - 이름: 한글
 - 입력 받은 후 “** 입력이 정상적으러 처리되었습니다**”를 출력한다.
3. 2가지 입력 후 Splash 화면 구현 (화면을 지우는 기능 system(cls))
 - After processing, display the following splash screen format

```

+++++
                        [마그라테아 ver 0.1]

      풀 한 포기 없는 황무지에서 반짝이는 행성을 만들내는 곳 마그라테아,
      사람들이 보지 못하는 잠재력을 찾고 전문가의 손길을 더해 보석을 빛는 곳,
      마그라테아에 오신걸 환영합니다.

+++++
[사용자]: 아서 덴트                      [실행 시간]:2023년 8월 20일
=====
  
```

[제약 사항]

- 코드는 splash.c 파일에 작성해야 한다.
- 스플래시 화면은 프로그램을 가동한 사람의 이름과 현재 날짜를 "yyyy-mm-dd" 형식으로 입력 받아야 한다.
- 입력과 출력은 표준 입력 및 출력 함수만 사용한다.
- 표준 라이브러리는 stdio.h와 stdlib.h만 사용한다.

Homework

1. Finish Step 1, Course 1, Problem 2
2. Read Step 1, Course 1, Problem 3, 4

See you next week!

DO NOT miss the classes

