

Kuality

신입생 교육 - 4주차



목표



- Coding Convention
 - C언어의 Coding Convention
- 함수
 - 최대공약수 프로그램
 - 팩토리얼
 - 주의사항



Coding Convetion

- 새로운 코드를 작성하는 일 vs 기존 코드를 수정하는 일
 - 기존 코드를 수정하는일이 훨씬 어렵다
 - 특히, 남이 짠 코드일수록 더더욱
- 유지보수에 투자되는 비용을 최소화 하기 위하여 통일된 코드 작성
- 언어마다 존재
- 변수 네이밍, 함수 네이밍, 파일 네이밍, 들여쓰기 등

Coding Convetion



- <https://users.ece.cmu.edu/~eno/coding/CCodingStandard.html>

C Coding Standard

Adapted from <http://www.possibility.com/Cpp/CppCodingStandard.html> and NetBSD's style guidelines

For the C++ coding standards click [here](#)

Contents


1. [Names](#)
 - (important recommendations below)
 - [Include Units in Names](#)
 - [Structure Names](#)
 - [C File Extensions](#)
 - (other suggestions below)
 - [Make Names Fit](#)
 - [Variable Names on the Stack](#)
 - [Pointer Variables](#)
 - [Global Constants](#)
 - [Enum Names](#)
 - [#define and Macro Names](#)
2. [Formatting](#)
 - (important recommendations below)
 - [Brace / } Policy](#)
 - [Parens / \) with Key Words and Functions Policy](#)
 - [A Line Should Not Exceed 78 Characters](#)
 - [If Then Else Formatting](#)
 - [switch Formatting](#)
 - [Use of goto continue break and ?:](#)
 - (other suggestions below)
 - [One Statement Per Line](#)
3. [Documentation](#)
 - (important recommendations below)
 - [Comments Should Tell a Story](#)
 - [Document Decisions](#)
 - [Use Headers](#)
 - [Make Gotchas Explicit](#)
 - [Commenting function declarations](#)
 - (other suggestions below)
 - [Include Statement Documentation](#)
4. [Complexity Management](#)
 - [Layering](#)

함수




- 반복적으로 사용되는 코드를 모듈화하여 재사용 가능한 코드 블록으로 만드는 방법
- 구조를 단순화하고, 유지보수성을 높이는 데에 도움을 줍니다.

c

 Copy code

```
반환형 함수이름(매개변수1, 매개변수2, ...) {  
    함수의 기능  
    return 반환값;  
}
```

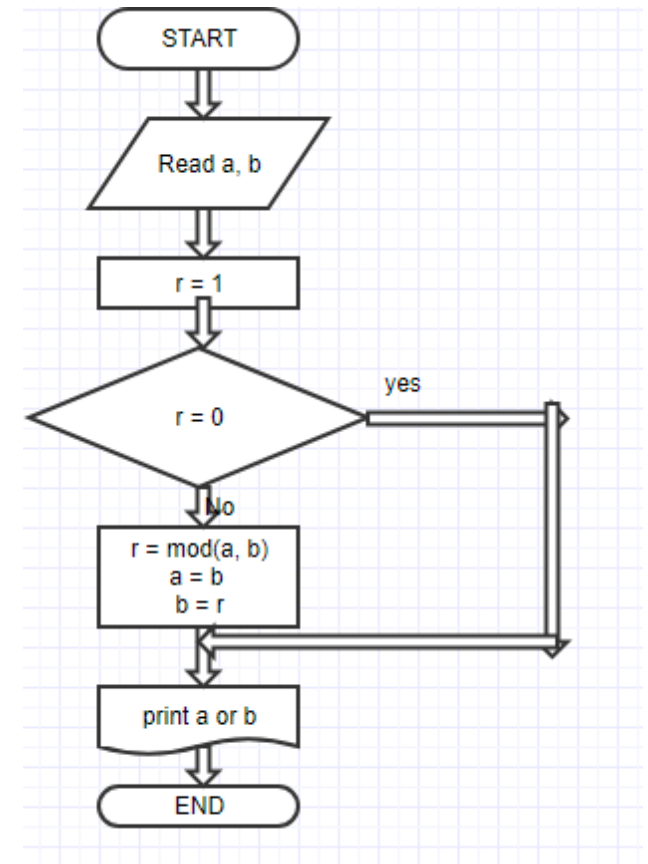
c

 Copy code

```
int add(int a, int b) {  
    int result = a + b;  
    return result;  
}
```

함수

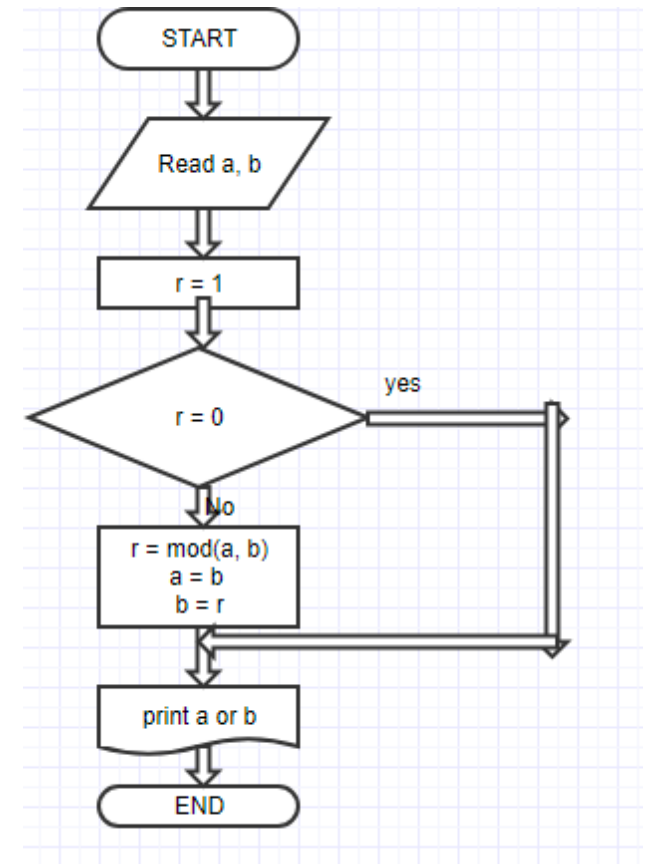
- 최대 공약수 프로그램 작성
- 함수가 없는 코드 : <https://pastebin.com/6UV4WJKk>
- 10개의 최대 공약수를 구한다면..?
- 1000개면?



함수



- 최대 공약수 프로그램 작성
- 함수가 있는 코드 : <https://pastebin.com/e6ELbK5U>



함수

- N! 팩토리얼을 구하는 함수 작성
- 함수의 원형 : `int factorial(int n)`



함수

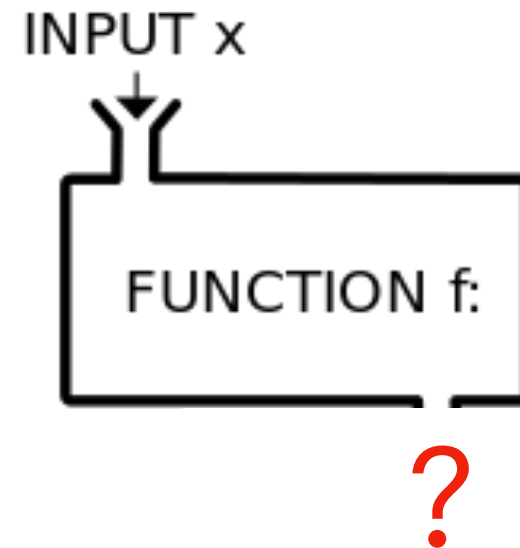


- N! 팩토리얼을 구하는 함수 작성
- 함수의 원형 : `int factorial(int n)`
- <https://pastebin.com/GVS6xwHc>

함수



- 주의사항
 - void 함수는 큰 의미가 없다
 - Input이 있는데 Output이 없는 함수
 - print 계열
 - main도 함수!
 - 원형은, `int main(int argc, char **argv)`



함수



- 프로그램의 진입점 : main
- main 함수의 원형 : `int main(int argc, char **argv)`
- argc : Argument의 개수
- argv: Argument의 배열
- Argument란?
- https://en.cppreference.com/w/c/language/main_function

함수



```
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf("argc = %d\n", argc);
    for(int ndx = 0; ndx != argc; ++ndx)
        printf("argv[%d] --> %s\n", ndx, argv[ndx]);
    printf("argv[argc] = %p\n", (void*)argv[argc]);
}
```



과제

• 필수 내용

- 모두의 C언어 5장까지 학습
- 학점을 문자로(A, A+, C, D- 등)으로 입력하여 평균 학점을 계산하는 프로그램을 자유롭게 구성
 - 필요할경우, strcmp 함수에 대해 찾아보기

• 선택 내용

- switch ~ case의 동작 원리에 대해 어셈블리 구현 찾아보기

• 제출 시각

- 2023년 4월 6일 수요일 23:59 까지

• 제출 방법

- GitHub kuality-2023-1 Repository 에 올리기
- VS의 경우 프로젝트 이름은 week04-assignment