

Week 01 21.10.29. 07:00 PM

---

+

o

•

# C Programming

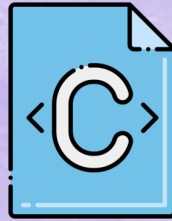
Second-half period / CMD-C\_Programming Study

Mentor : Jh9892

# INDEX

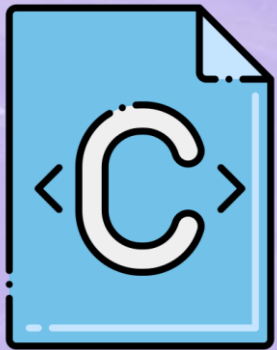
- 💧 Programming Language
- 💧 Feature of “C Language”
- 💧 Data Type & Calculating & Operation
- 💧 Input / Output
- 💧 Conditional Statement
- 💧 Loop Statement
- 💧 Assignment

# Programming Language





# Feature of “C Language”



Low - Level Language

Procedure-oriented Language

For H/W

So Fast

# Data Type

| 종류     | 자료형           | 메모리 크기 | 값의 범위   |
|--------|---------------|--------|---|
| 정수형 변수 | char(문자)      | 1 byte | -128~+128   |
|        | int           | 4 byte | -2147483648 ~ +2147483647                                 |
|        | unsigned int  | 4 byte | 0 ~ 4294967295  |
|        | long long int | 8 byte | -9,223,372,036,854,775,808 ~<br>9,223,372,036,854,775,807 |
|        | short         | 2 byte | -32768 ~ 32767  |
| 실수형 변수 | float         | 4 byte | $3.4 \cdot 10^{-37} \sim 3.4 \cdot 10^{38}$ (실수)          |
|        | double        | 8 byte | $1.7 \cdot 10^{-307} \sim 3.4 \cdot 10^{308}$ (실수)        |

# Calculating

| 서식 지정자 | 출력대상                | 출력형태              |
|--------|---------------------|-------------------|
| %d     | char, short, int    | 부호 있는 10진 정수      |
| %ld    | long                | 부호 있는 10진 정수      |
| %lld   | long long           | 부호 있는 10진 정수      |
| %u     | unsigned int        | 부호 없는 10진 정수      |
| %o     | unsigned int        | 부호 없는 8진 정수       |
| %x, %X | float, double       | 부호 없는 16진 정수      |
| %f     | float               | 10진수 방식의 부동소수점 실수 |
| %lf    | long double, double | 10진수 방식의 부동소수점 실수 |
| %c     | char, short, int    | 값에 대응하는 문자        |
| %s     | char*(문자열)          | 문자열               |
| %p     | void(주소값)           | 포인터 주소 값          |

# Calculating

| 산술 연산자 | 설명                                     |
|--------|--|
| +      | 왼쪽의 피연산자에 오른쪽의 피연산자를 더함.               |
| -      | 왼쪽의 피연산자에서 오른쪽의 피연산자를 뺌.               |
| *      | 왼쪽의 피연산자에 오른쪽의 피연산자를 곱함.               |
| /      | 왼쪽의 피연산자를 오른쪽의 피연산자로 나눔.               |
| %      | 왼쪽의 피연산자를 오른쪽의 피연산자로 나눈 후, 그 나머지를 반환함. |



# Operation

| 대입 연산자 | 설명   |
|--------|--|
| =      | 왼쪽의 피연산자에 오른쪽의 피연산자를 대입함.                        |
| +=     | 왼쪽의 피연산자에 오른쪽의 피연산자를 더한 후, 그 결과값을 왼쪽의 피연산자에 대입함. |
| -=     | 왼쪽의 피연산자에서 오른쪽의 피연산자를 뺀 후, 그 결과값을 왼쪽의 피연산자에 대입함. |
| *=     | 왼쪽의 피연산자에 오른쪽의 피연산자를 곱한 후, 그 결과값을 왼쪽의 피연산자에 대입함. |
| /=     | 왼쪽의 피연산자를 오른쪽의 피연산자로 나눈 후, 그 결과값을 왼쪽의 피연산자에 대입함. |
| %=     | 왼쪽의 피연산자를 오른쪽의 피연산자로 나눈 후, 그 나머지를 왼쪽의 피연산자에 대입함. |



# Operation

| 증감 연산자 | 설명                                  |
|--------|-------------------------------------|
| ++x    | 먼저 피연산자의 값을 1 증가시킨 후에 해당 연산을 진행함.   |
| x++    | 먼저 해당 연산을 수행하고 나서, 피연산자의 값을 1 증가시킴. |
| --x    | 먼저 피연산자의 값을 1 감소시킨 후에 해당 연산을 진행함.   |
| x--    | 먼저 해당 연산을 수행하고 나서, 피연산자의 값을 1 감소시킴. |

# Operation

| 비교 연산자 | 설명                                    |
|--------|---------------------------------------|
| ==     | 왼쪽의 피연산자와 오른쪽의 피연산자가 같으면 1을 반환함.      |
| !=     | 왼쪽의 피연산자와 오른쪽의 피연산자가 같지 않으면 1을 반환함.   |
| >      | 왼쪽의 피연산자가 오른쪽의 피연산자보다 크면 1을 반환함.      |
| >=     | 왼쪽의 피연산자가 오른쪽의 피연산자보다 크거나 같으면 1을 반환함. |
| <      | 왼쪽의 피연산자가 오른쪽의 피연산자보다 작으면 1을 반환함.     |
| <=     | 왼쪽의 피연산자가 오른쪽의 피연산자보다 작거나 같으면 1을 반환함. |

# Operation

| 논리 연산자 | 설명  |
|--------|---|
| &&     | 논리식이 모두 참이면 1을 반환함. (논리 AND 연산)           |
|        | 논리식 중에서 하나라도 참이면 1을 반환함. (논리 OR 연산)       |
| !      | 논리식의 결과가 참이면 0을, 거짓이면 1을 반환함. (논리 NOT 연산) |

| 비트 연산자 | 설명  |
|--------|---|
| &      | 대응되는 비트가 모두 1이면 1을 반환함. (비트 AND 연산)                   |
|        | 대응되는 비트 중에서 하나라도 1이면 1을 반환함. (비트 OR 연산)               |
| ^      | 대응되는 비트가 서로 다르면 1을 반환함. (비트 XOR 연산)                   |
| ~      | 비트를 1이면 0으로, 0이면 1로 반전시킴. (비트 NOT 연산)                 |
| <<     | 지정한 수만큼 비트들을 전부 왼쪽으로 이동시킴. (left shift 연산)            |
| >>     | 부호를 유지하면서 지정한 수만큼 비트를 전부 오른쪽으로 이동시킴. (right shift 연산) |



# Operation

## 삼항 연산자(ternary operator)

삼항 연산자는 다른 언어에는 존재하지 않는 C언어만의 독특한 연산자입니다.

이 연산자는 C언어에서도 유일하게 피연산자를 세 개나 가지는 조건 연산자입니다.

삼항 연산자의 문법은 다음과 같습니다.

### 문법

조건식 ? 반환값1 : 반환값2

1. 주소 연산자(&)

2. 참조 연산자(\*)

주소 연산자(&)는 변수의 이름 앞에 사용하여, 해당 변수의 주소값을 반환합니다.

'&'기호는 앰퍼샌드(ampersand)라고 읽으며, 번지 연산자라고도 불립니다.

참조 연산자(\*)는 포인터의 이름이나 주소 앞에 사용하여, 포인터에 가리키는 주소에 저장된 값을 반환합니다.

# Input / Output

```
# include <stdio.h>

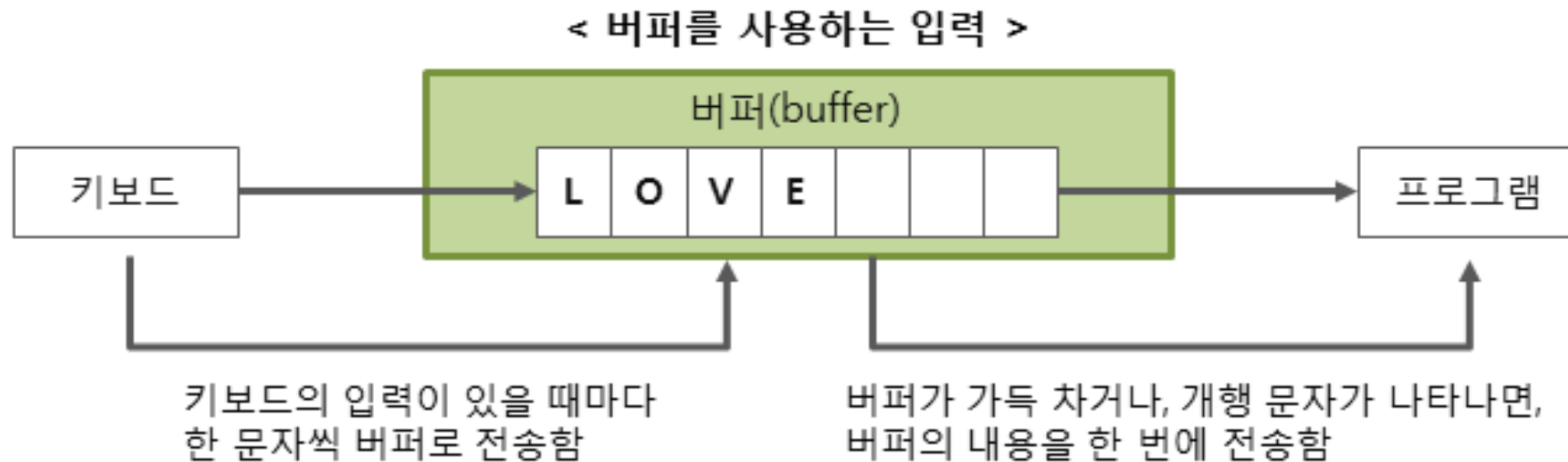
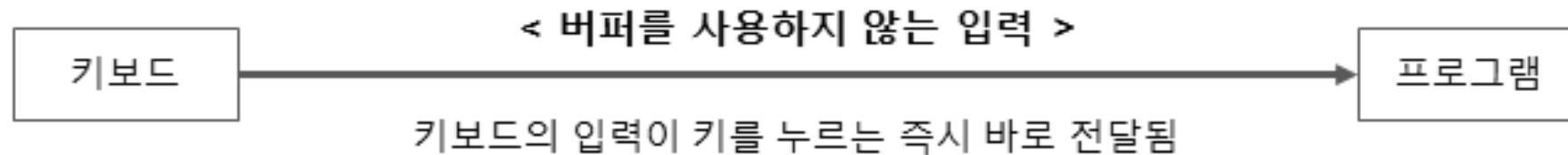
int main(){
    int test;
    scanf("%d", &test);
    printf("%d", test);

    return 0;
}
```

# Input / Output

## 버퍼(buffer)

지금까지 우리가 사용한 printf(), scanf(), puts() 함수 등은 C언어의 표준 입출력 함수입니다.  
이러한 표준 입출력 함수를 사용할 때에는 버퍼(buffer)라는 임시 메모리 공간을 사용하게 됩니다.





# Conditional Statement

```
# include <stdio.h>

int main(){
    char ch;

    puts("당신의 성별을 약자로 적어주세요 : ");
    puts("(남성=M, 여성=F)");
    scanf("%c", &ch);

    if (ch=='m' | ch=='M'){
        printf("당신은 남성입니다.\n");
    }
    else if(ch=='f' | ch=='F'){
        printf("당신은 여성입니다.\n");
    }
    else{
        printf("성별을 잘못 입력하셨습니다.\n");
    }

    return 0;
}
```

# Loop Statement

```
# include <stdio.h>

int main(){
    int i = 0;
    int num = 5;

    while (i < num){
        printf("while 문 이 %d 번째 반복 수행 중 입니다 .\n", i + 1);
        i++; // 이 부분을 삭제 하면 무한 루프에 빠지게 됨
    }
    return 0;
}
```

# Loop Statement

```
# include <stdio.h>

int main(){
    int i = 0;
    int num = 3;

    do{
        printf("do/while 문 이 %d 번째 반복 수행 중 입니다 .\n", i + 1);
        i++;
    }while (i < num);

    return 0;
}
```



# Loop Statement

```
# include <stdio.h>

int main(){
    int i = 0;
    int num = 3;

    for (i = 0; i < num; i++){
        printf("for 문 이 %d 번째 반복 수행 중 입니다 .\n", i + 1);
    }

    return 0;
}
```

# Assignment

## Q. Palindrome(회문)여부를 검출

Palindrome은 회문이라 불리며 정방향에서 읽는 것과 역방향에서 읽는 것이 동일한 낱말, 숫자, 문자열을 뜻한다.

검출하기 위해 사용되는 낱말, 숫자, 문자열의 모음을 '구문' 이라고 칭하고

구문사이에 속해 있는 띄어쓰기나 특수문자, 문장부호는 무시한다.

하나의 문자열을 입력 받아서 그 문자열이 회문인지 아닌지 검출하는 프로그램을 작성하시오.

(단, 문자열 입력 시, 소문자로 된 알파벳 a~z로만 이루어진 7자의 문자열이 입력되며,

공백과 띄어쓰기, 특수문자, 문장부호는 고려하지 않도록 한다.)

### 예시 01

입력 : racecar

출력 : this is palindrome!!

### 예시 02

입력 : raceccr

출력 : this is not palindrome...

C:\Users\User\Desktop\C\_Study\week01\ans\ans.exe

```
racecar  
this is paliIndrome!!
```

```
-----  
Process exited after 3.805 seconds with return value 0  
계속하려면 아무 키나 누르십시오 . . .
```

C:\Users\User\Desktop\C\_Study\week01\ans\ans.exe

```
raceccr  
this is not palindrome...
```

```
-----  
Process exited after 29.47 seconds with return value 0  
계속하려면 아무 키나 누르십시오 . . .
```