

# 컴퓨터공학 All in One

---

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)  
제 10강 - 문자

# 학습 목표

## 문자

- 1) C언어에서 문자를 처리하는 방법에 대해서 학습합니다.
- 2) 문자와 버퍼의 관계를 바르게 이해하고 활용합니다.

# 문자

## 아스키 코드

- 1) C 프로그램의 문자는 아스키 코드 (Ascii Code)를 따릅니다.
- 2) 아스키 코드는 0~127중의 1바이트로 구성되며 주요 문자를 출력하도록 해줍니다.

0	48
A	65
a	97

# 문자

## 아스키 코드

- 1) 캐릭터형 자체에 숫자를 넣어서 처리할 수 있습니다.
- 2) 문자 입출력에서 형식 지정자로 %c를 씁니다.

```
#include <stdio.h>

int main(void) {
    char a = 65;
    printf( "%c\n", a);
    system( "pause");
}
```

# 문자

## 문자 입출력 함수

- 1) 이러한 문자는 getchar()를 이용해서 입력을 받는 방식을 이용할 수 있습니다.
- 2) getchar()는 단 하나의 문자를 입력 받습니다.

```
#include <stdio.h>

int main(void) {
    char a = getchar();
    printf("%c\n", a);
    system("pause");
}
```

# 문자

## 문자

- 1) 문자열을 처리할 때 버퍼의 개념이 많이 사용됩니다.
- 2) 버퍼(Buffer)란 임시적으로 특정한 데이터를 저장하기 위한 목적으로 사용됩니다.
- 3) C 프로그램은 기본적으로 사용자가 의도하지 않아도 자동으로 버퍼를 이용해 입출력을 처리합니다.

# 문자

## 문자와 버퍼

### 1) 입력 버퍼로 인해 흔히 발생하는 오류

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main()
{
    int a;
    char c;
    scanf("%d", &a);
    printf("%d\n", a);
    scanf("%c", &c);
    printf("%c\n", c);
    system("pause");
    return 0;
}
```

# 문자

## 문자와 버퍼

1) 남아 있는 입력 버퍼를 항상 지울 수 있습니다.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main() {
    int a; char c;
    scanf("%d", &a);
    printf("%d\n", a);
    int temp;
    // 한 자씩 받아서 파일의 끝이거나 개행 문자를 만나면 입력을 멈추므로 항상 입력 버퍼를 비웁니다.
    while ((temp = getchar()) != EOF && temp != '\n') { }
    scanf("%c", &c);
    printf("%c\n", c);
    system("pause");
    return 0;
}
```



# 배운 내용 정리하기

## 문자

- 1) C언어에서의 문자는 내부적으로 아스키 코드 구조를 따릅니다.
- 2) C언어에서 문자 입출력의 내부에는 버퍼(Buffer)가 존재합니다.