

K G 아 이 티 뱅 크

C 언 어

C L A N G U A G E

표준입력함수

## 표준입력함수

❖ `getchar()` : 단일문자를 입력을 받아 단일문자변수에 저장

➤ 입력한 것은 무엇이 되었든 문자로 인식하여 저장

➤ 한글은 단일문자로 취급이 안되서 저장하지 못함

❖ 사용예시

```
문자변수 = getchar();  
char ch = getchar();
```

< 파일이름 : getcharEX.c >

**실습문제1. 아래의 조건을 만족하는 코드를 작성하세요.**

**조건**

문자를 입력해 화면에 출력합니다.

**결과**

문자 입력 >> A

입력한 문자 : A

## 표준입력함수

❖ `gets_s( )` : 문자열을 문자열변수에 저장

➤ 입력한 것은 무엇이 되었든 문자열로 인식하여 저장

➤ 단일문자를 넣어도 문자열로 저장하기 때문에 주의

❖ 사용예시

```
gets_s(문자열변수, 크기);  
char word1[100];  
gets_s(word1, 100);
```

< 파일이름 : getsEX.c >

### 실습문제2.

자기 이름, 나이, 꿈을 `gets_s`로 입력을 받아 출력하세요.

#### 결과

이름 : (이름)  
나이 : (나이)  
꿈 : (꿈)

### ❖ scanf\_s( ) : c언어의 만능 입력 함수

- 서식문자로 값을 인식해 변수의 자료형에 맞춰 저장하는 함수
- 저장할 공간(메모리주소)을 요구하기에 변수에 &를 붙임
  - ✓ 단, 문자열은 구조적인 특징으로 붙이지 않음
- 서식문자 외의 다른 모든 문자의 사용은 비권장
- 여러 개를 입력을 받을 때는 서식문자와 변수를 맞춰서 준비
  - ✓ 여러 개는 띄어쓰기로 구분하며 엔터로 입력을 종료함
- 문자나 문자열 입력시 반드시 크기도 같이 넣어줘야 함

#### ❖ scanf\_s의 사용1

```
char ch1  
char word1[100];  
int num1;  
double dnum1;
```

#### ❖ scanf\_s의 사용2

```
scanf_s("%c", &ch1, 1);  
scanf_s("%s", word1, 100);  
scanf_s("%d", &num1);  
scanf_s("%lf", &dnum1);
```

## 표준입력함수

❖ scanf\_s는 서식문자에 맞는 자료형을 엄격하게 요구함

➤ 매칭이 잘못되면 비정상적인 값이 입력되어 저장

서식문자	대응자료형	받는 값
%c	char	문자
%hd	short	2바이트 정수
%d	int	4바이트 정수
%s	char 변수명[x]	문자열
%f	float	단정밀도 실수
%lf	double	배정밀도 실수
%o	unsigned int	무부호 8진수
%x	unsigned int	무부호 16진수

## 표준입력함수

< 파일이름 : scanfEX1.c >

**실습문제3. 아래의 조건을 만족하는 코드를 작성하세요.**

### 조건

1. 두 정수를 입력을 받아 합연산 결과를 출력하세요.
2. 두 실수를 입력을 받아 곱연산 결과를 출력하세요.

\* 실수는 소수점 두자리까지 제한합니다.

### 결과

-- 입력구간 --

정수1 입력 :

정수2 입력 :

실수1 입력 :

실수2 입력 :

-- 출력구간 --

(정수1) + (정수2) = (결과1)

(실수1) \* (실수2) = (결과2)

## 표준입력함수

< 파일이름 : scanfEX2.c >

**실습문제4. 아래의 조건을 만족하는 코드를 작성하세요.**

### 조건

자기 이름, 나이, 꿈, 혈액형을  
scanf\_s로 입력을 받아 출력하세요.

### 결과

```
-- 입력구간 --
이름 입력      :
나이 입력      :
혈액형 입력     :
꿈 입력         :
-- 출력구간 --
이름   : (이름)
나이   : (나이)
혈액형 : (혈액형)
꿈      : (꿈)
```



## 표준입력함수

### ❖ 키보드로 입력한 값을 인식해 지정한 메모리주소에 저장

- 키보드 : 입력하는 자료를 문자로써 전달하는 느린 장치
- 컴퓨터 : 다양한 출처의 자료를 해석해 처리하는 빠른 장치

### ❖ 한쪽이 너무 빠르거나 너무 느리면 정상적인 처리가 안됨

#### ➤ 속도를 일정수준으로 완화시키는 임시공간 : 버퍼

- ① scanf\_s 실행시 OS에 요청하여 키보드와 컴퓨터를 연결
- ② 연결된 두 장치의 처리속도를 맞추기 위해 버퍼를 설정
- ③ 버퍼에 임시 저장된 문자들을 인식해 값으로 저장



### ❖ 임시 공간에 ASCII 코드 등으로 입력한 값을 저장

- ANSI에서 표준화한 정보교환용 부호체계 ASCII가 기준
  - 그 외 각 국가별 부호체계가 다수 존재
- 해당 부호체계를 기준으로 입력 버퍼에 저장
- 입력의 경우 **키보드로 입력한 모든 것이 저장됨**
  - **엔터, 띄어쓰기, 탭 등 대부분을 포함**

### ❖ 입력함수에서 읽어 들이는 방식은 공백문자를 기준

- 공백문자(' ', 'Wn', 'Wt') 를 기준으로 입력된 내용을 구분
- 입력함수보다 **많은 내용**을 입력하면 나머지는 버림
- 입력함수보다 **적은 내용**을 입력하면 **입력대기**로 진입
- **단일문자를 입력을 받을 경우 공백문자도 하나의 값으로 취급**

### ❖ 해결책으로 사용할 수 있는 방법

1. scanf\_s( )로 받는 값의 종류를 하나로 통일하기
  - 1) 단순한 내용이지만 아주 중요한 작업
2. 다른 값을 입력을 받을 때 버퍼를 초기화
  - 1) fflush( stdin ) : 버퍼 초기화( VS2015이후 미지원 )
  - 2) rewind( stdin ) : 버퍼 초기화
3. 의도적인 버퍼 조작 : getchar( )

#### ❖ 코드1. 값 종류 통일

```
scanf_s("%c%c", &ch1, 1, &ch2, 1);  
scanf_s("%d%d", &num1, &num2);
```

#### ❖ 코드2. 초기화

```
scanf_s("%c", &ch1, 1);  
rewind(stdin);  
scanf_s("%c", &ch2, 1);
```

#### ❖ 코드3. 의도적인 조작

```
scanf_s("%c", &ch1, 1);  
getchar();  
scanf_s("%c", &ch2, 1);
```

## 표준입력함수

< 파일이름 : scanfEX3.c >

**실습문제5. 아래의 조건을 만족하는 코드를 작성하세요.**

### 조건

1. 세 정수를 입력을 받아 합연산 결과를 출력하세요.
  2. 입력을 받은 세 정수의 평균을 출력하세요.
- \* 실수는 소수점 4자리까지만 출력합니다.

### 결과

-- 입력구간 --

정수 3개 입력 >>

-- 출력구간 --

입력된 3개의 정수의 합 : (결과1)

입력된 3개의 정수의 평균 : (결과2)