

Do it! C언어 입문

20164389 멀티미디어공학과 박병주

김상엽 지음 [이지스 퍼블리싱]



☰목차

12장 : 배열과 문자열

13장 : 포인터

14장 : 표준 입력 함수



배열과 문자열



배열과 문자열

배열

배열이란?

```
short student1, student2, student3, student4, student5,  
student6, student7, student8, student9, student10
```

```
student1 = 0;  
student2 = 0;  
student3 = 0;  
student4 = 0;  
student5 = 0;  
student6 = 0;  
student7 = 0;  
student8 = 0;  
student9 = 0;  
student10 = 0;
```

C언어는 데이터를 그룹으로 묶어서 표현하는 배열을 사용함

-> short student[9]



배열과 문자열

배열

-> short student[9]

student[0], student[1], student[2], student[3], student[4]
, student[5], student[6], student[7], student[8], student[9]

short

자료형

student[9];

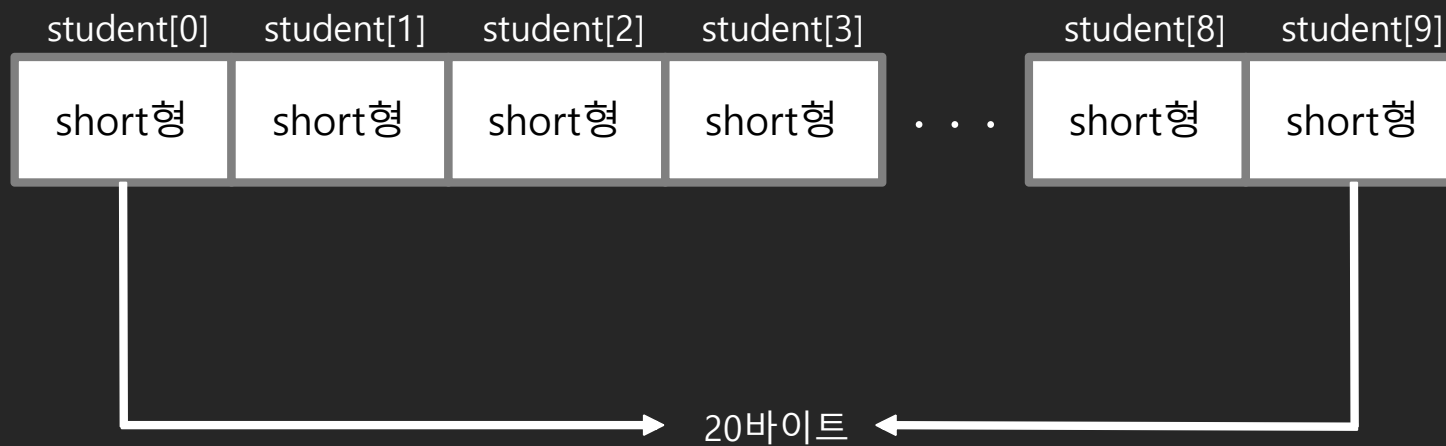
변수이름[요소개수];



배열과 문자열

배열

-> short student[9]



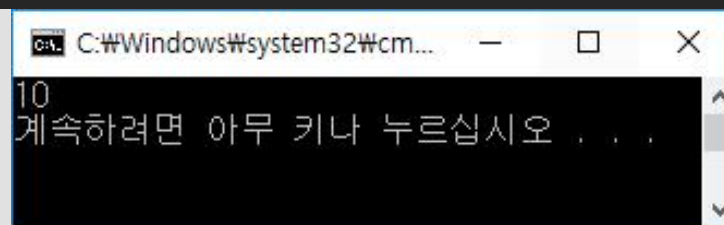
배열과 문자열

배열의 특정 요소에 값 대입하기

```
#include <stdio.h>

void main()
{
    short student[20];

    student[1] = 10;
    printf("%d\n", student[1]);
}
```



C:\Windows\system32\cmd...
10
계속하려면 아무 키나 누르십시오 . . .



배열과 문자열

배열 초기화 하기

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

```
void main()
```

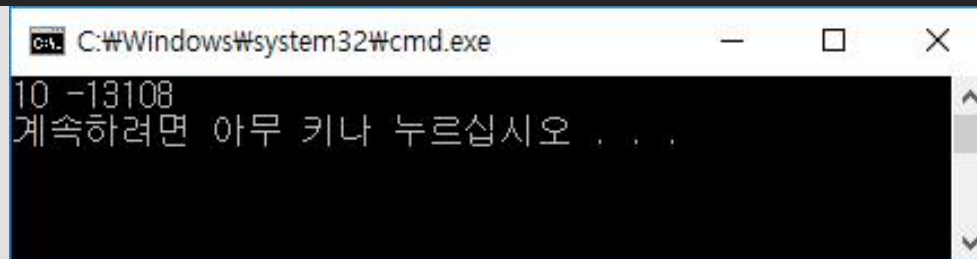
```
{
```

```
    short student[20];
```

```
    student[1] = 10;
```

```
    printf("%d %d\n", student[1], student[2]);
```

```
}
```



C:\Windows\system32\cmd.exe

```
10 -13108  
계속하려면 아무 키나 누르십시오 . . .
```



배열과 문자열


배열 초기화 하기

배열을 선언할 때는 []안에 상수를 써주어야 하지만,
특정 요소에 값을 대입할 때는 []안에 변수를 사용할 수 있음

```
#include <stdio.h>

void main()
{
    short student[20];
    for (int i = 0; i < 20; i++) student[i] = 5;

    for (int i = 0; i < 20; i++) printf("Student[%d] = %d\n", i, student[i]);
}
```



```
C:\Windows\system32\cmd.exe
Student[0] = 5
Student[1] = 5
Student[2] = 5
Student[3] = 5
Student[4] = 5
Student[5] = 5
Student[6] = 5
Student[7] = 5
Student[8] = 5
Student[9] = 5
Student[10] = 5
Student[11] = 5
Student[12] = 5
Student[13] = 5
Student[14] = 5
Student[15] = 5
Student[16] = 5
Student[17] = 5
Student[18] = 5
Student[19] = 5
계속하려면 아무 키나 누르십시오 . . .
```



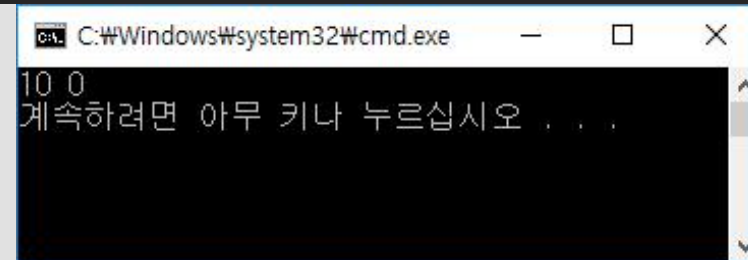
배열과 문자열

배열 초기화 하기

심표를 사용한 배열 초기화 문법

```
#include <stdio.h>

void main()
{
    short student[20] = {0,};
    student[1] = 10;
    printf("%d %d\n", student[1], student[2]);
}
```



```
short data[5] = {3,};
               = {3,0,0,0,0};
```

배열 크기 생략하기

```
short data[ ] = {1,2,3,4,5};
```



배열과 문자열

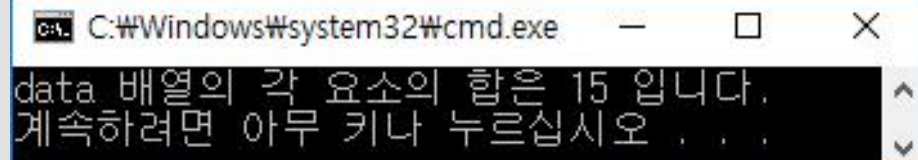
배열의 각 요소에 저장된 값 합하기

```
#include <stdio.h>

void main()
{
    char data[5] = { 1,2,3,4,5, };
    int result = 0, i;

    for (i = 0; i < 5; i++) {
        result = result + data[i];
    }

    printf("data 배열의 각 요소의 합은 %d 입니다. \n", result);
}
```



C:\Windows\system32\cmd.exe

data 배열의 각 요소의 합은 15 입니다.
계속하려면 아무 키나 누르십시오 . . .



배열과 문자열

배열

배열은 사용자 정의 자료형

char형은 1바이트 크기이지만 char three[3]이란
배열을 선언하면 3바이트 자료형이 만들어짐



배열과 문자열

문자열

배열로 문자열 표현하기

```
char data1 =  
'H';  
char data2 = 'e';  
char data3 = 'l';  
char data4 = 'l';  
char data5 = 'o';  
char data6 = '!';
```



```
char data[7] = { 'H', 'e', 'l', 'l', 'o', '!', 0};
```



```
char data[7] = {"Hello!"};
```

※ 문자열 끝에 자동으로 NULL이 포함됨



배열과 문자열

문자열

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

```
void main()
```

```
{
```

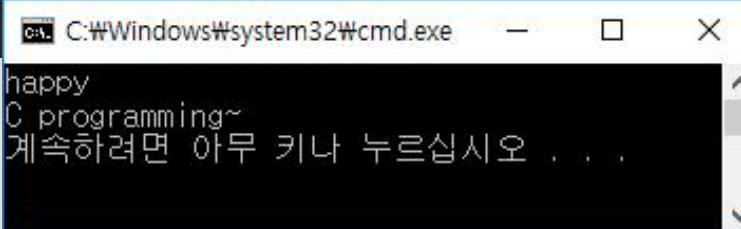
```
    char data[6] = { 'h','a','p','p','y',0 };
```

```
    char ment[] = { "C programming~" };
```

```
    printf("%s\n",data);
```

```
    printf("%s\n", ment);
```

```
}
```



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window shows the output of the C program: "happy", "C programming~", and "계속하려면 아무 키나 누르십시오 . . .".



배열과 문자열

저장된 문자열 길이 구하기

```
#include <stdio.h>

void main()
{
    char data[6] = "happy";
    int count = 0;
    while (data[count] != 0) {
        count++;
    }
    printf("문자열의 길이는 %d 입니다.\n", count);
}
```

C:\ 선택 C:\Windows\system32\cmd.exe

문자열의 길이는 5 입니다.
계속하려면 아무 키나 누르십시오 . . .



배열과 문자열

문자열을 다루는 함수

strlen 함수 : 문자열의 길이를 구하는 함수

strcpy 함수 : 문자열을 복사하는 함수

strcat 함수 : 문자열 뒤에 다른 문자열을 덧붙이는 함수



배열과 문자열

문자열 길이 출력하기

```
#include <stdio.h>
#include <string.h>
```

```
void main()
{
```

```
    int data_length;
```

```
    char data[10] = { 'h', 'a', 'p', 'p', 'y', 0 };
```

```
    data_length = strlen(data);
```

```
    printf("data length = %d\n", data_length);
```

```
}
```

C:\Windows\system32\cmd.exe

data length = 5

계속하려면 아무 키나 누르십시오 . . .



배열과 문자열

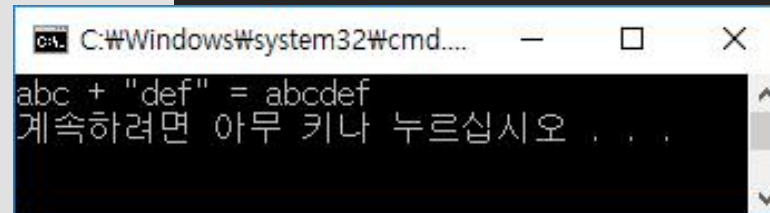
두 개의 문자열 합치기

```
#include <stdio.h>
#include <string.h>

void main( )
{
    char data[10] = { 'a','b','c',0 };
    char result[16];

    strcpy(result, data);
    strcat(result, "def");

    printf("%s + #\"def#\" = %s\n", data, result);
}
```



C:\Windows\system32\cmd...
abc + "def" = abcdef
계속하려면 아무 키나 누르십시오 . . .



배열과 문자열

2차원 배열

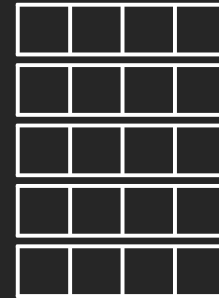
```
char data1, data2, data3, data4;  
=> char data[4];
```

```
char data[4];
```



```
char data1[4], data2[4], data3[4], data4[4], data5[4];  
=> char data[5][4];
```

```
char data[5][4];
```

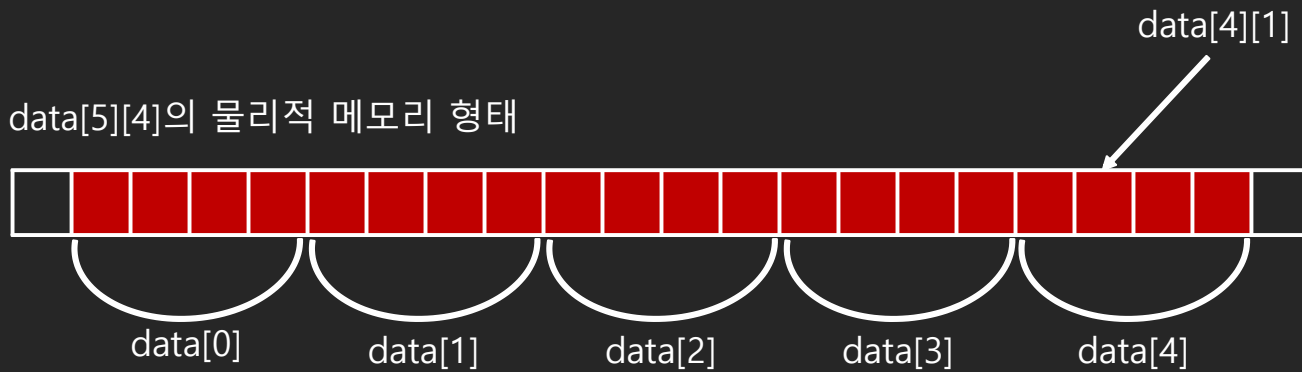


배열과 문자열

2차원 배열

char data[5][4]를 메모리에 나열하면?

data[5][4]의 물리적 메모리 형태



배열과 문자열

2차원 배열

char data[5][4]를 메모리에 나열하면?

data[5][4]의 논리적 메모리 형태

data[0][0]	data[0][1]	data[0][2]	data[0][3]
data[1][0]	data[1][1]	data[1][2]	data[1][3]
data[2][0]	data[2][1]	data[2][2]	data[2][3]
data[3][0]	data[3][1]	data[3][2]	data[3][3]
data[4][0]	data[4][1]	data[4][2]	data[4][3]



배열과 문자열

2차원 배열 초기화하기

```
char temp1[3]={1,2,3};  
char temp2[3]={4,5,6};
```

```
char temp[2][3] = { {1, 2, 3}, {4, 5, 6} };
```

temp[0][0]=1	temp[0][1]=2	temp[0][2]=3
temp[1][0]=4	temp[1][1]=5	temp[1][2]=6



배열과 문자열

2차원 배열 초기화하기

```
#include <stdio.h>

void main()
{
    char data[3][4] = { {0,0,2,0},{1,1,0,0},{2,1,0,2} };
    int x, y;

    for (y = 0; y < 3; y++) {
        for (x = 0; x < 4; x++) {
            printf("%d행 %d행에", y + 1, x + 1);
            if (data[y][x] == 1) printf(" 검은 돌이 놓여 있습니다.\n");
            else if (data[y][x] == 2) printf(" 흰 돌이 놓여 있습니다.\n");
            else printf("는 돌이 놓여 있지 않습니다.\n");
        }
    }
}
```

```
C:\Windows\system32\cmd.exe
1행 1행에 는 돌이 놓여 있지 않습니다.
1행 2행에 검은 돌이 놓여 있습니다.
1행 3행에 흰 돌이 놓여 있습니다.
1행 4행에 는 돌이 놓여 있지 않습니다.
2행 1행에 검은 돌이 놓여 있습니다.
2행 2행에 검은 돌이 놓여 있습니다.
2행 3행에 검은 돌이 놓여 있지 않습니다.
2행 4행에 검은 돌이 놓여 있지 않습니다.
3행 1행에 흰 돌이 놓여 있습니다.
3행 2행에 검은 돌이 놓여 있습니다.
3행 3행에 는 돌이 놓여 있지 않습니다.
3행 4행에 흰 돌이 놓여 있습니다.
계속하려면 아무 키나 누르십시오...
```



포인터



포인터

포인터란?

포인터 : 메모리의 주소를 저장하고 있는 메모리

포인터 변수임을
나타내는 기호

short *ptr;

자료형

포인터 변수 이름



포인터

변수가 위치한 메모리 주소 출력

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

```
void main()
```

```
{
```

```
    short birthday;
```

```
    short *ptr;
```

```
    ptr = &birthday;
```

```
    printf("birthday 변수의 주소는 %p입니다.\n", ptr);
```

```
}
```

C:\Windows\system32\cmd.exe

birthday 변수의 주소는 010FFC9C입니다.
계속하려면 아무 키나 누르십시오 . . .



포인터

다른 변수의 주소를 사용하여
포인터로 값 대입

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

```
void main()  
{
```

```
    short birthday;
```

```
    short *ptr;
```

```
    ptr = &birthday;
```

```
    *ptr = 0x0412;
```

```
    printf("birthday = %d (0x%04X)\n", birthday, birthday);
```

```
}
```

C:\Windows\system32\cmd.exe

birthday = 1042 (0x0412)

계속하려면 아무 키나 누르십시오 . . .



포인터

ptr과 *ptr의 차이점

ptr=

포인터 변수의 값 변경
(가리키는 대상의 주소)

*ptr=

포인터가 가리키는
대상의 값 변경



포인터

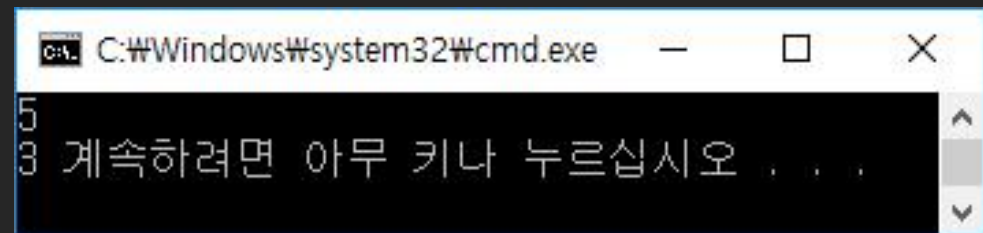
다른 함수의 변수 주소 받기

```
#include <stdio.h>

void Test(short *ptr)
{
    short soft = 0;
    soft = *ptr;
    *ptr = 3;

    printf("%d %n", soft);
}

void main()
{
    short tips = 5;
    Test(&tips);
    printf("%d ", tips);
}
```



```
C:\Windows\system32\cmd.exe
5
3 계속하려면 아무 키나 누르십시오 . . .
```



포인터

Swap함수

```
#include <stdio.h>

void Swap(int *pa, int *pb)
{
    int temp = *pa;
    *pa = *pb;
    *pb = temp;
}

void main()
{
    int start = 96, end = 5;

    printf("before : start = %d, end = %d\n", start, end);
    if (start > end) {
        Swap(&start, &end);
    }
    printf("after : start = %d, end = %d\n", start, end);
}
```

C:\Windows\system32\cmd.exe

before : start = 96, end = 5
after : start = 5, end = 96
계속하려면 아무 키나 누르십시오 . . .



포인터

포인터와 const 키워드

int * const p; - 주소 변경 불가
const int *p; - 대상의 값 변경 불가

```
int data = 5, temp = 0;  
int *const p = &data;  
*p = 3;  
p = &temp; 오류발생
```

```
int data = 5;  
const int *p = &data;  
*p = 3; 오류발생
```

const int * const p; - 주소, 대상의 값 변경 불가

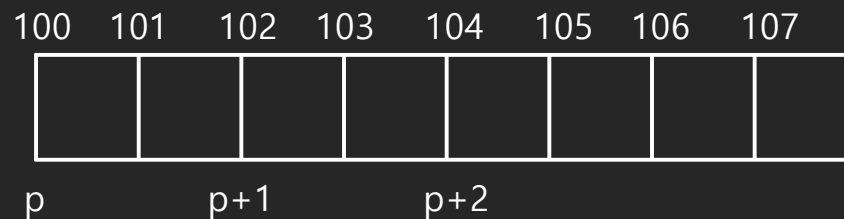


포인터

포인터의 주소 연산

```
short data = 0;  
short *p =  
&data;  
p = p + 1;
```

포인터 변수에 저장된 주소 값을 1만큼 증가시킴



```
char *p1 = (char *)100;  
short *p2 = (short *)100;  
int *p3 = (int *)100;  
double *p4 = (double *)100;  
p1++; //101번지  
p2++; //102번지  
p3++; //104번지  
p4++; //108번지
```



포인터

void 포인터

```
int data = 0;  
void *p = &data;  
*p = 5;
```



```
int data = 0;  
void *p = &data;  
*(int*)p = 5;
```



포인터

void 포인터의 활용

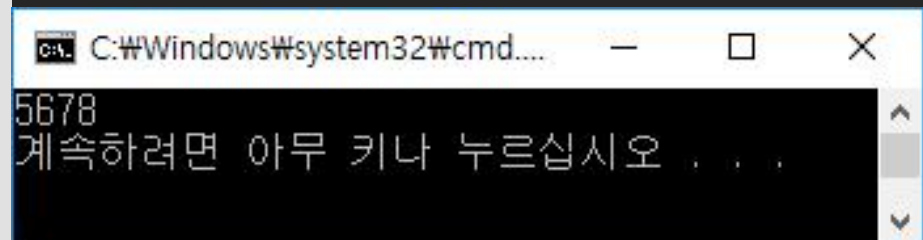
```
#include <stdio.h>

int GetData(void *p_data, char type)
{
    int result = 0;

    if (type == 1) result = *(char *)p_data;
    else if (type == 2) result = *(short *)p_data;
    else if (type == 4) result = *(int *)p_data;

    return result;
}

void main()
{
    int data = 0x12345678;
    printf("%X\n", GetData(&data, 2));
}
```



C:\Windows\system32\cmd... 5678
계속하려면 아무 키나 누르십시오 . . .



표준 입력 함수



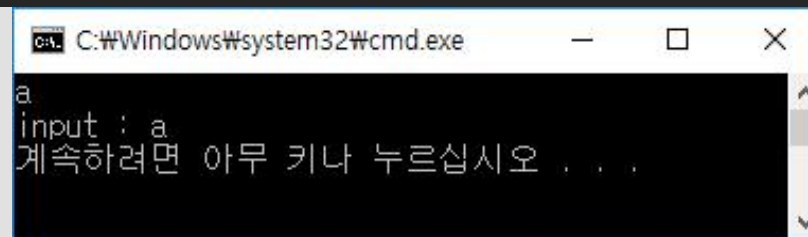
표준입력함수

문자를 한 개를 입력 받는 getchar 함수

```
#include <stdio.h>

void main()
{
    int input_data;
    input_data = getchar();

    printf("input : %c\n", input_data);
}
```



```
C:\Windows\system32\cmd.exe
a
input : a
계속하려면 아무 키나 누르십시오 . . .
```

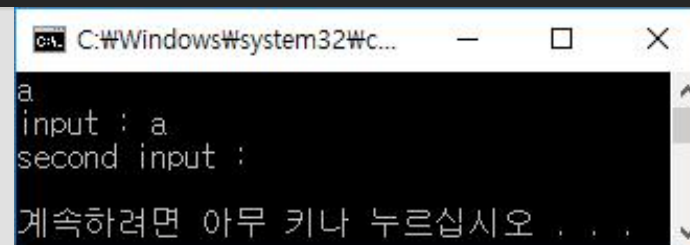


표준입력함수

문자를 한 개를 입력 받는 getchar 함수

```
#include <stdio.h>

void main()
{
    int input_data;
    input_data = getchar();
    printf("input : %c\n", input_data);
    input_data = getchar();
    printf("second input : %c\n", input_data);
}
```



```
cmd. C:\Windows\system32\cmd.exe
a
input : a
second input :
계속하려면 아무 키나 누르십시오 . . .
```

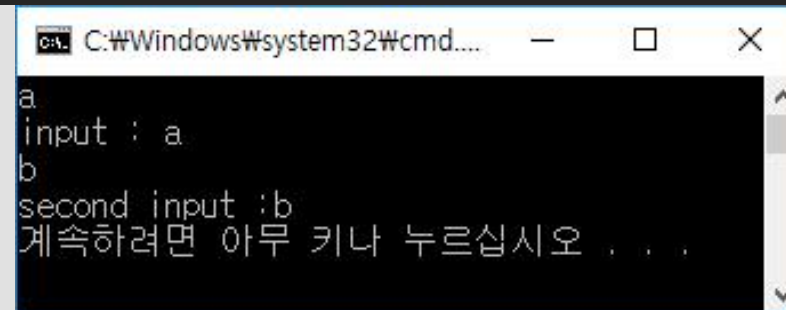


표준입력함수

문자를 한 개를 입력 받는 getchar 함수

```
#include <stdio.h>

void main()
{
    int input_data;
    input_data = getchar();
    getchar();
    printf("input : %c\n", input_data);
    input_data = getchar();
    getchar();
    printf("second input : %c\n", input_data);
}
```



```
C:\Windows\system32\cmd...
a
input : a
b
second input : b
계속하려면 아무 키나 누르십시오 . . .
```

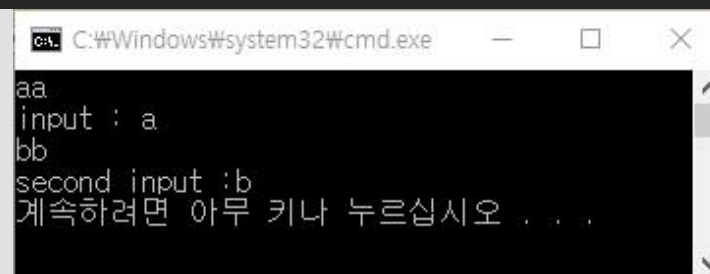


표준입력함수

문자를 한 개를 입력 받는 getchar 함수

```
#include <stdio.h>

void main()
{
    int input_data;
    input_data = getchar();
    rewind(stdin);
    printf("input : %c\n", input_data);
    input_data = getchar();
    rewind(stdin);
    printf("second input : %c\n", input_data);
}
```



```
C:\Windows\system32\cmd.exe
aa
input : a
bb
second input : b
계속하려면 아무 키나 누르십시오 . . .
```

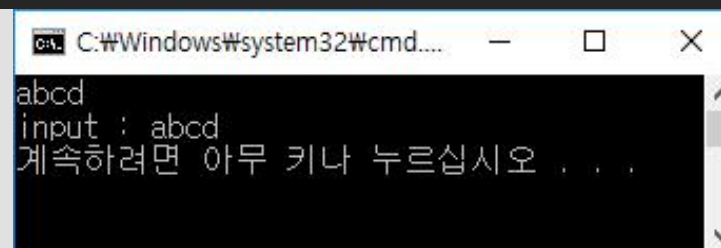


표준입력함수

문자열을 입력 받는 gets 함수

```
#include <stdio.h>

void main()
{
    char input_string[10];
    gets(input_string);
    printf("input : %s\n", input_string);
}
```



```
C:\Windows\system32\cmd...
abcd
input : abcd
계속하려면 아무 키나 누르십시오...
```

※입력 버퍼가 남지 않는다.



표준입력함수

문자열을 숫자로

```
#include <stdio.h>

int ArrayToInteger(char string[])
{
    int count = 0, num = 0;

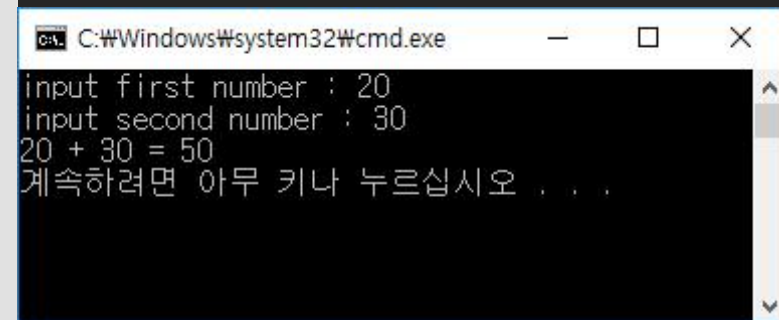
    while (string[count] != 0) {
        num = num * 10 + string[count] - '0';
        count++;
    }
    return num;
}

void main()
{
    int first_num, second_num;
    char first_string[16], second_string[16];

    printf("input first number : ");
    gets(first_string);
    printf("input second number : ");
    gets(second_string);

    first_num = ArrayToInteger(first_string);
    second_num = ArrayToInteger(second_string);

    printf("%d + %d = %d\n", first_num, second_num, first_num + second_num);
}
```



C:\Windows\system32\cmd.exe

```
input first number : 20
input second number : 30
20 + 30 = 50
계속하려면 아무 키나 누르십시오 . . .
```



표준입력함수

만능 표준 입력 함수 scanf

```
int data;  
scanf("%d", &data);
```

입력형식 지정
키워드

입력 받은 값을
저장할 변수의 주소

※엔터키와 공백문자를 구분하지 못한다.

문자열에 공백이 포함된다면 gets 함수가 더 편리



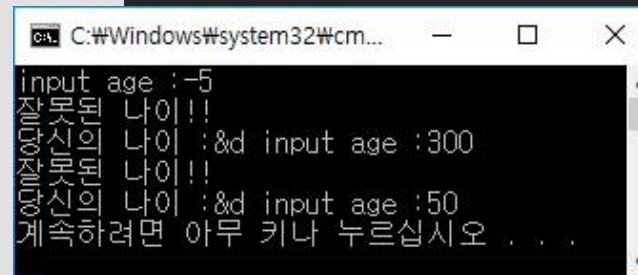
표준입력함수

scanf함수의 사용

```
#include <stdio.h>

void main()
{
    int num = 0;
    while (1) {
        printf("input age :");
        scanf("%d", &num);

        if (num > 0 && num <= 130) {
            break;
        }
        else {
            printf("잘못된 나이!!\n");
        }
        printf("당신의 나이 :&d ", num);
    }
}
```



```
C:\Windows\system32\cmd.exe
input age :-5
잘못된 나이!!
당신의 나이 :&d input age :300
잘못된 나이!!
당신의 나이 :&d input age :50
계속하려면 아무 키나 누르십시오 . . .
```





감사합니다.

The End.