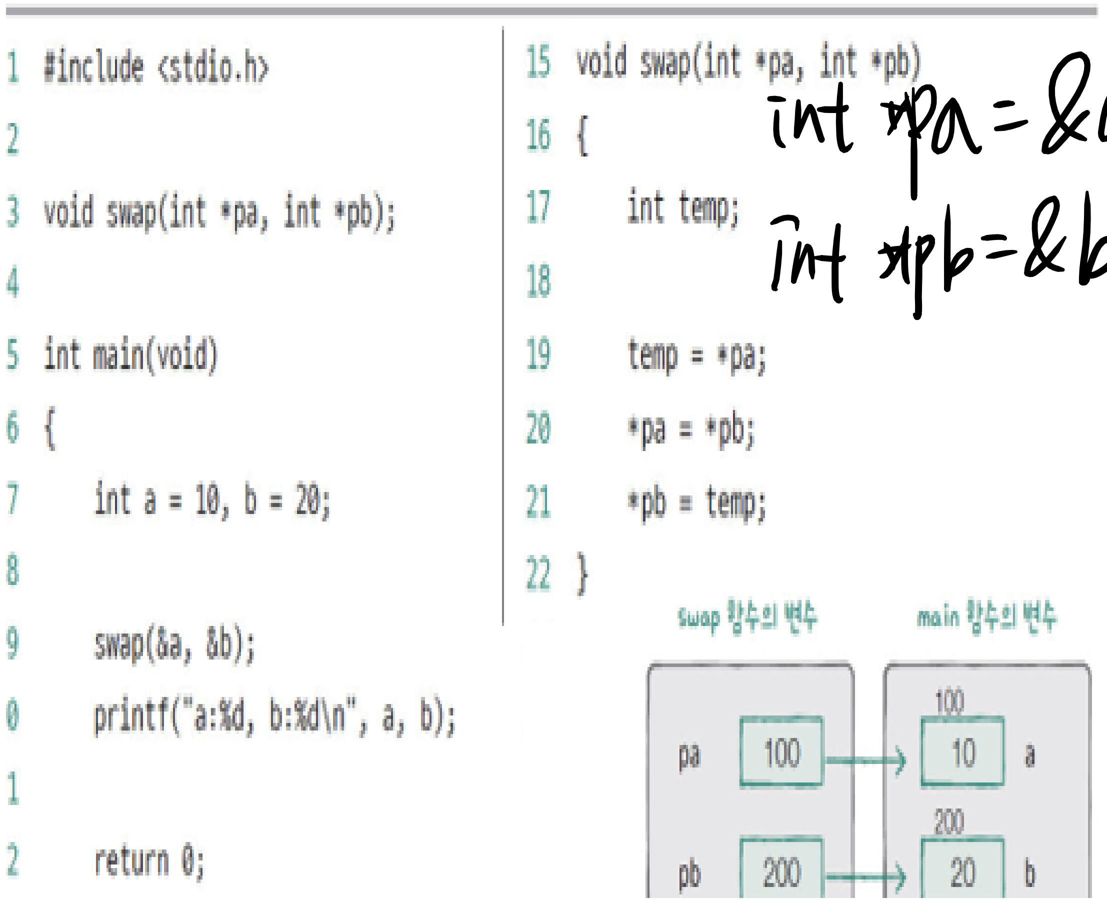
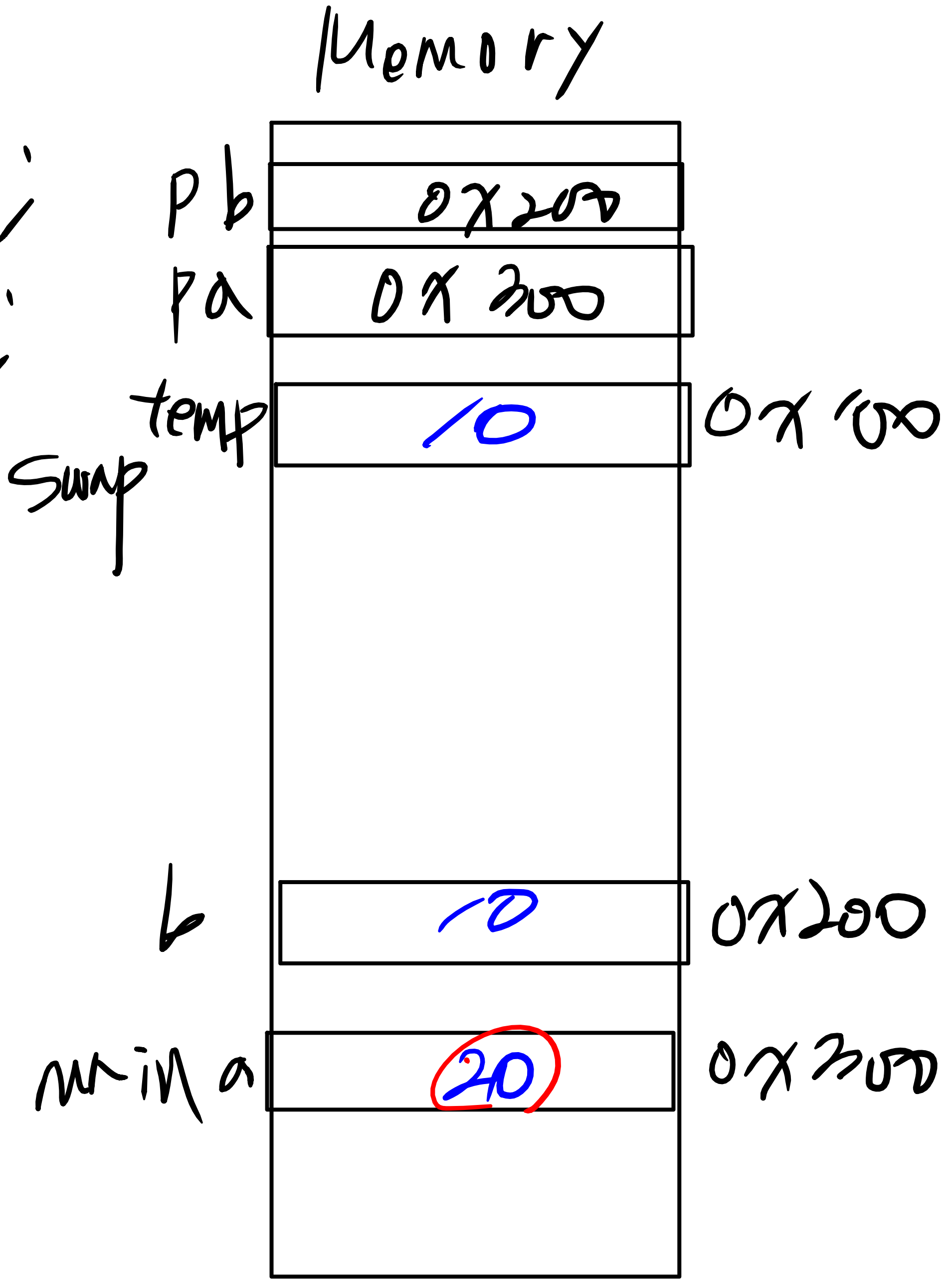


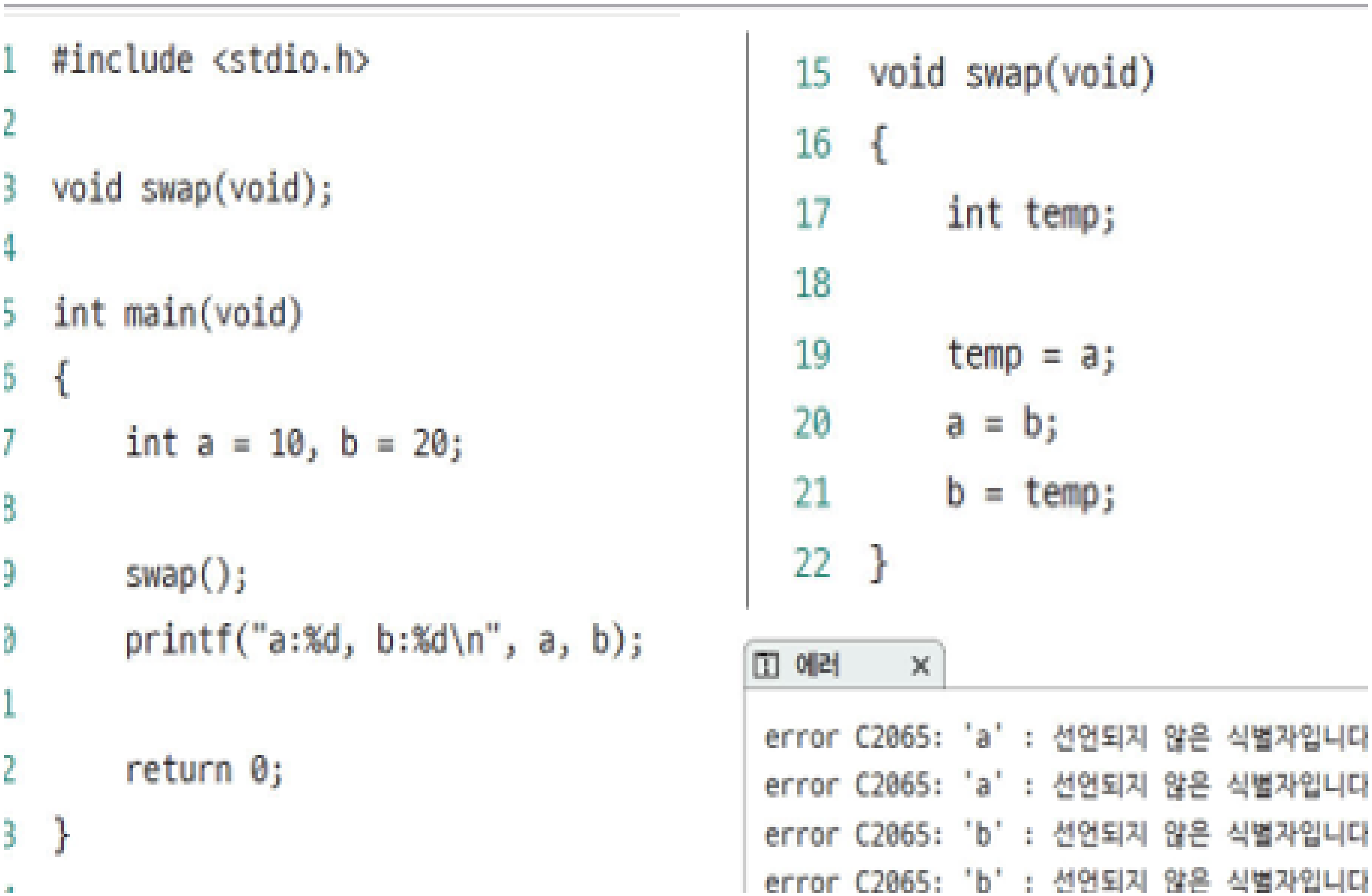
```c
#include <stdio.h>

void swap(int *pa, int *pb);

int main(void)
{
    int a = 10, b = 20;

    swap(&a, &b);
    printf("a:%d, b:%d\n", a, b);

    return 0;
```

```c
void swap(int *pa, int *pb)
{
    int temp;

    temp = *pa;
    *pa = *pb;
    *pb = temp;
}
```

int *pa=&a;

int *pb=&b;

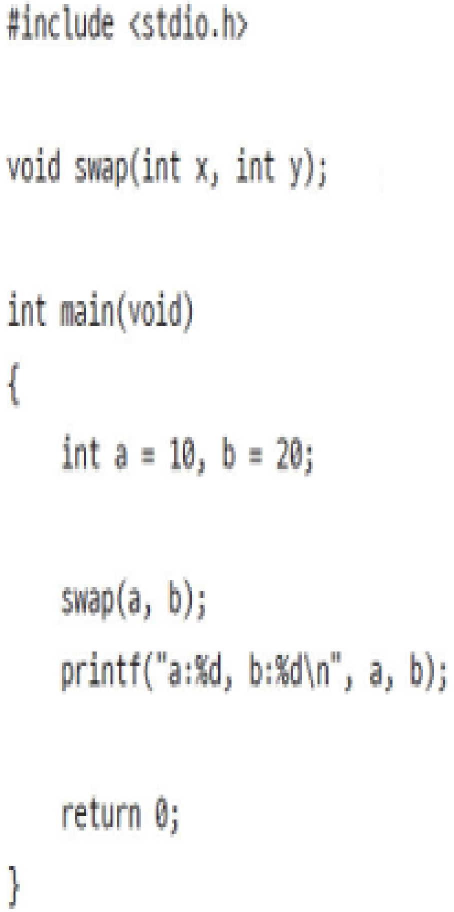




```c
#include <stdio.h>

void swap(void);

int main(void)
{
    int a = 10, b = 20;

    swap();
    printf("a:%d, b:%d\n", a, b);

    return 0;
}
```

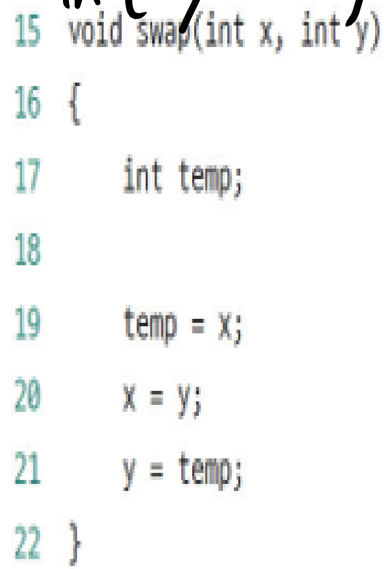
```c
void swap(void)
{
    int temp;

    temp = a;
    a = b;
    b = temp;
}
```

```
error C2065: 'a' : 선언되지 않은 식별자입니다
error C2065: 'a' : 선언되지 않은 식별자입니다
error C2065: 'b' : 선언되지 않은 식별자입니다
error C2065: 'b' : 선언되지 않은 식별자입니다
```

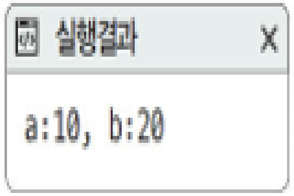


```c
#include <stdio.h>

void swap(int x, int y);

int main(void)
{
    int a = 10, b = 20;

    swap(a, b);
    printf("a:%d, b:%d\n", a, b);

    return 0;
}
```

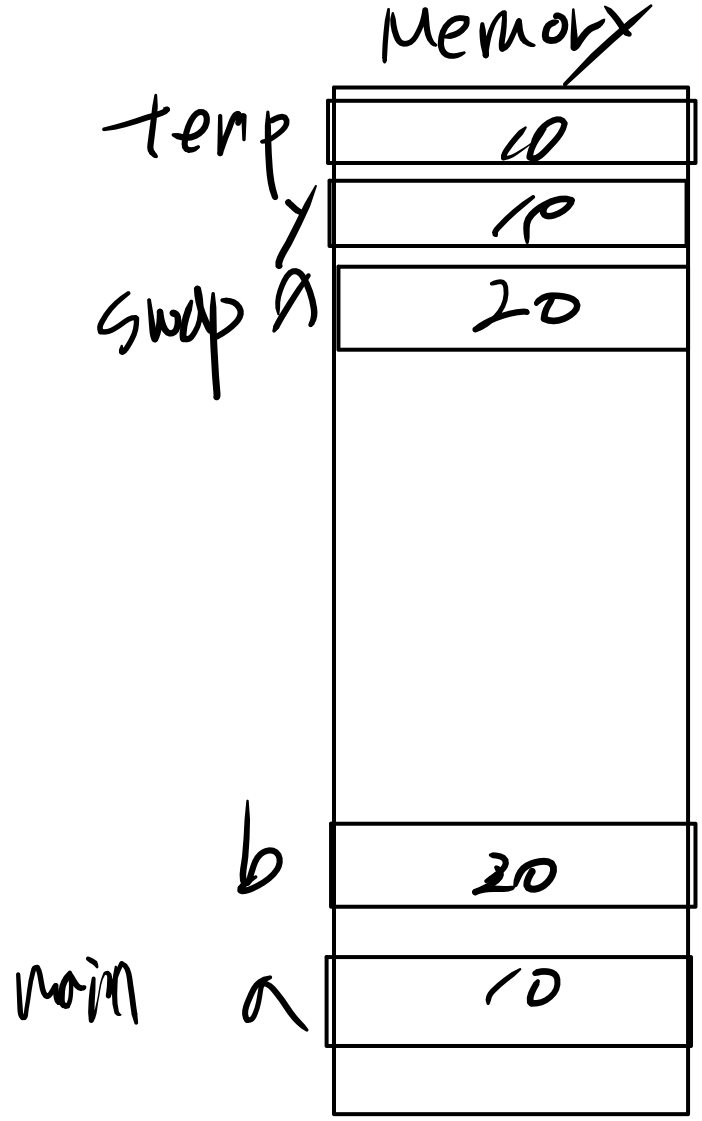
int x = a;
int y = b;



```c
void swap(int x, int y)
{
    int temp;

    temp = x;
    x = y;
    y = temp;
}
```

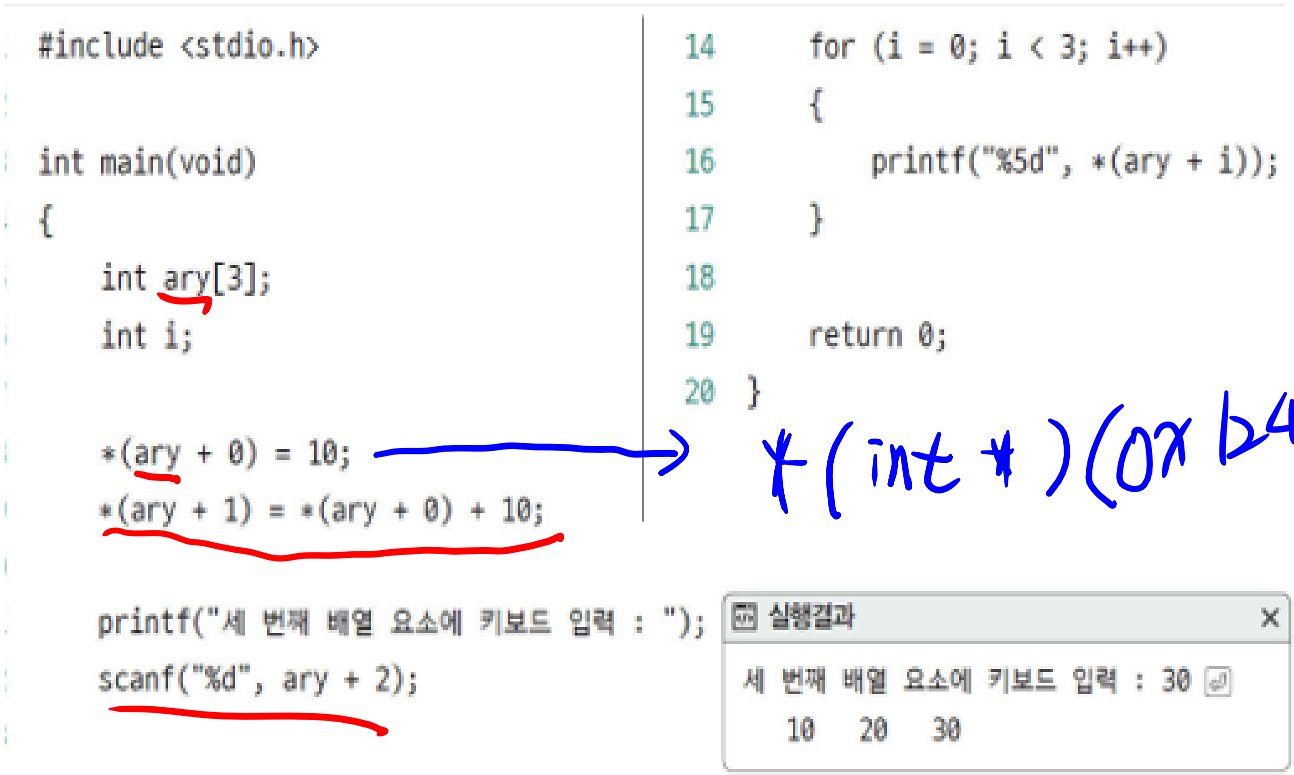


```
a:10, b:20
```



10장 배열과 포인터

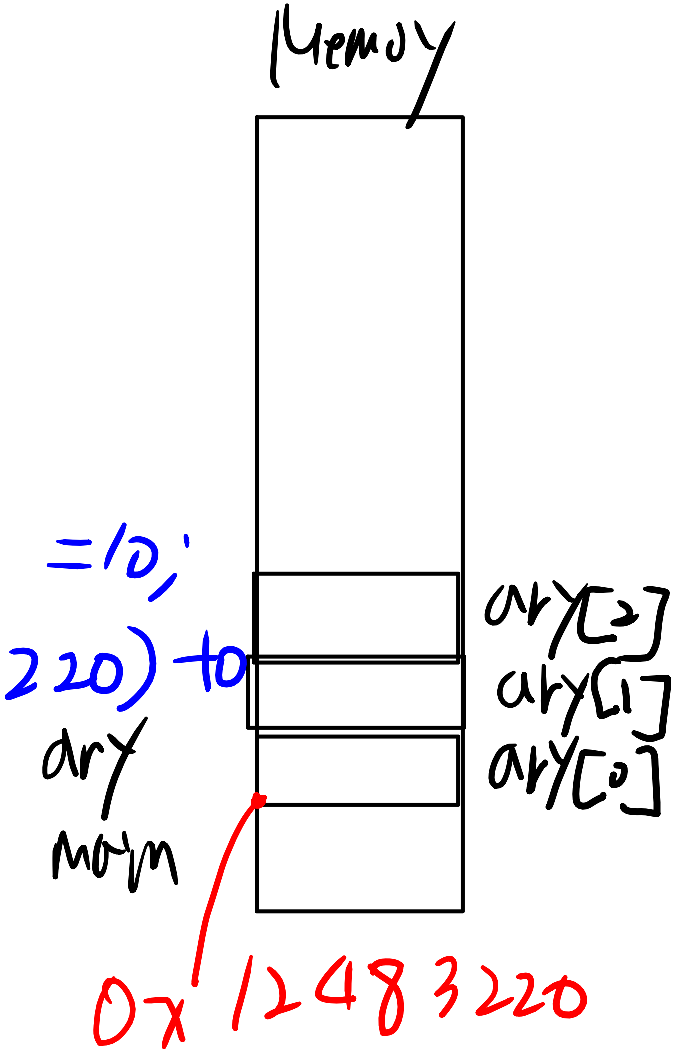
배열의 각요소의 주소값을 포인터를 이용하여 접근 Read/write

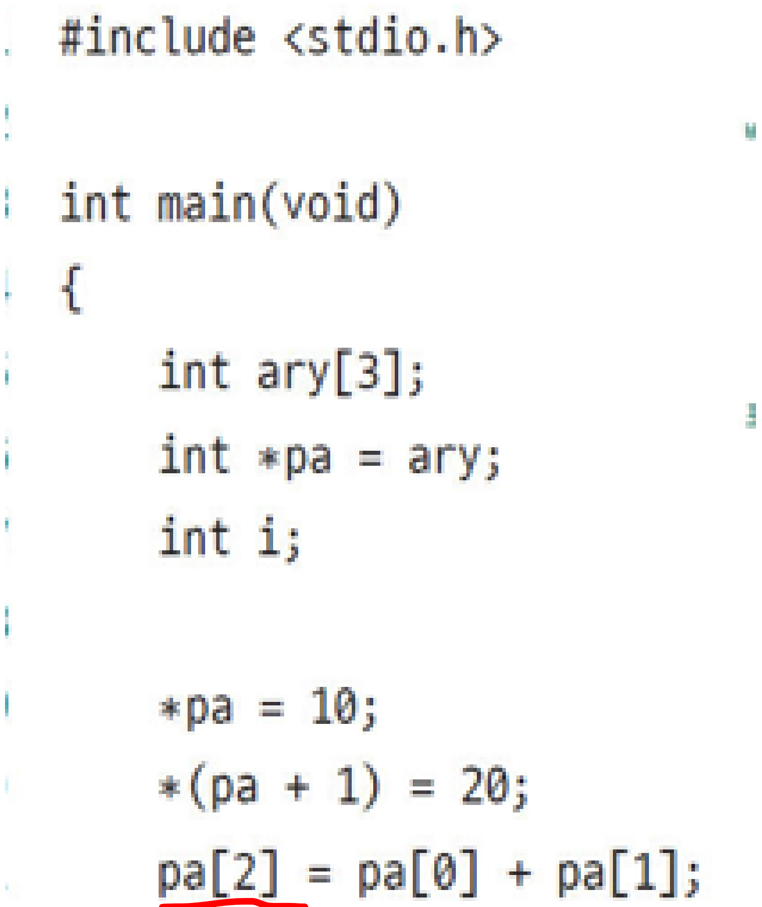


```
#include <stdio.h>

int main(void)
{
    int ary[3];
    int i;

    *(ary + 0) = 10;
    *(ary + 1) = *(ary + 0) + 10;

    printf("세 번째 배열 요소에 키보드 입력 : ");
    scanf("%d", ary + 2);

    for (i = 0; i < 3; i++)
    {
        printf("%5d", *(ary + i));
    }

    return 0;
}
```

```
세 번째 배열 요소에 키보드 입력 : 30
   10   20   30
```

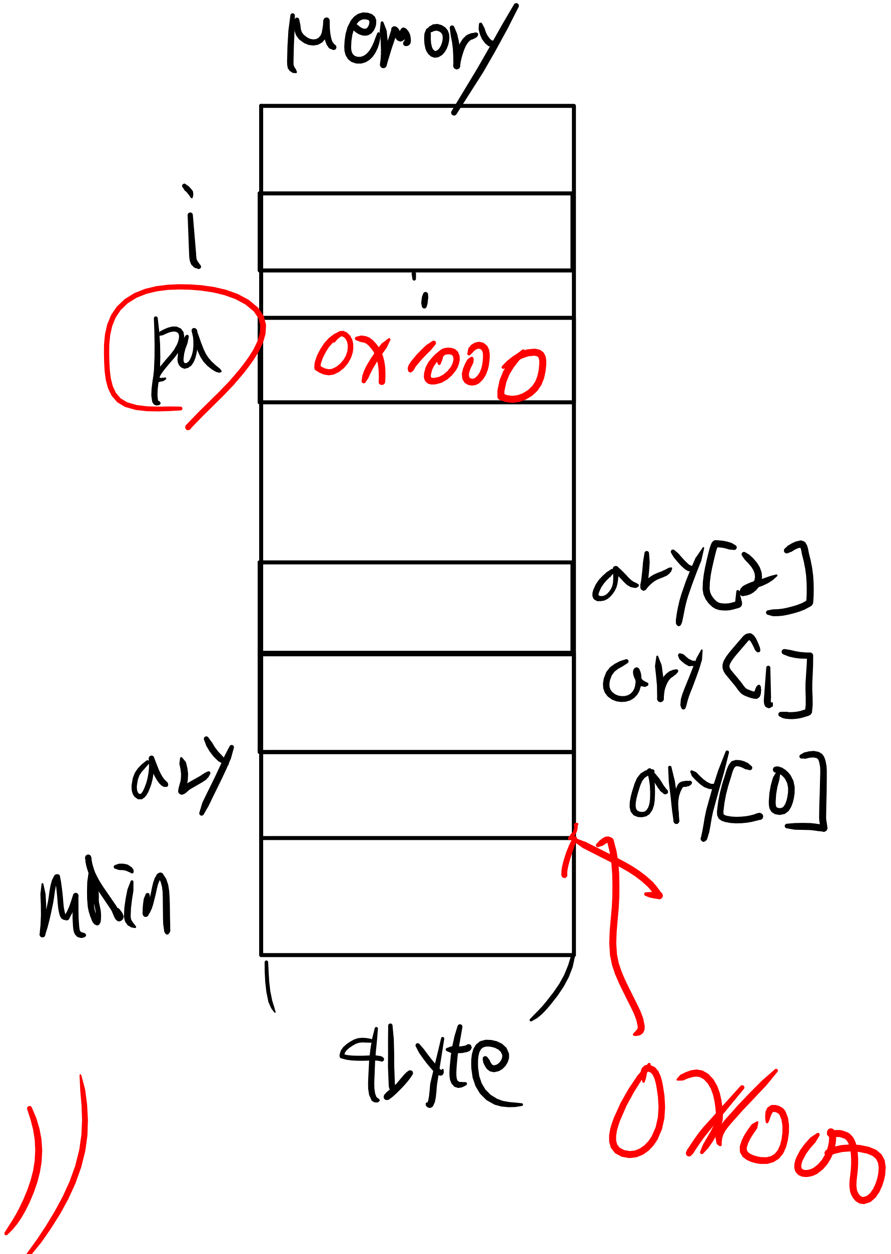
*(int *)(0x b483220)+0 =10;



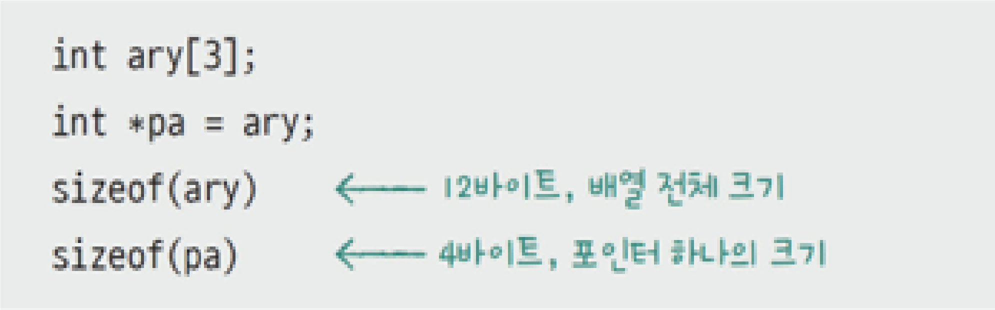


```
#include <stdio.h>

int main(void)
{
    int ary[3];
    int *pa = ary;
    int i;

    *pa = 10;
    *(pa + 1) = 20;
    pa[2] = pa[0] + pa[1];
```

↳ *(int *(3c) +2))

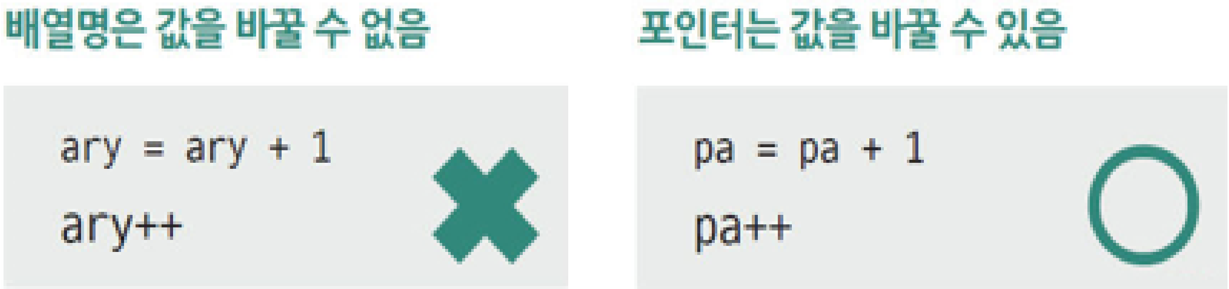


- sizeof 연산의 결과가 다르다.



```
int ary[3];
int *pa = ary;
sizeof(ary)     ← 12바이트, 배열 전체 크기
sizeof(pa)      ← 4바이트, 포인터 하나의 크기
```

- 상수와 변수의 차이가 있다.



배열명은 값을 바꿀 수 없음

```
ary = ary + 1
ary++
```

✖

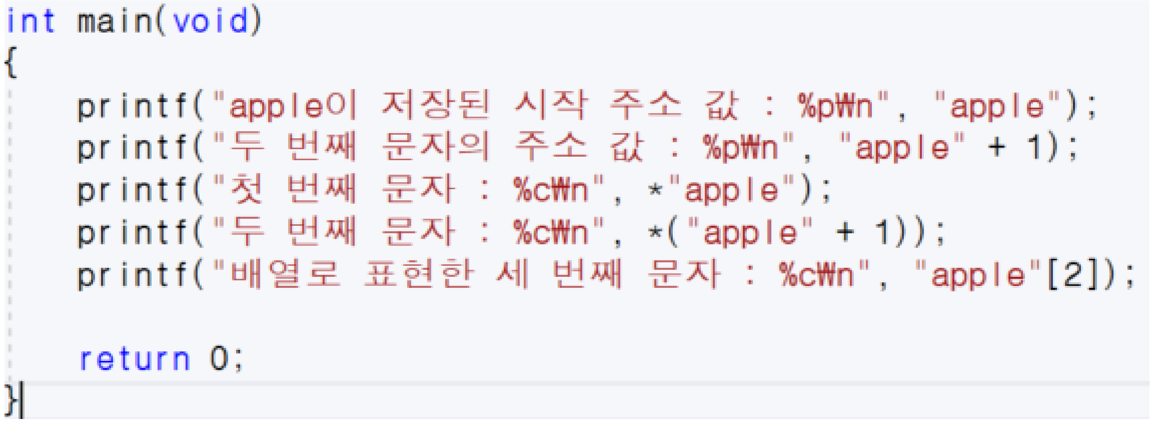
포인터는 값을 바꿀 수 있음

```
pa = pa + 1
pa++
```

○

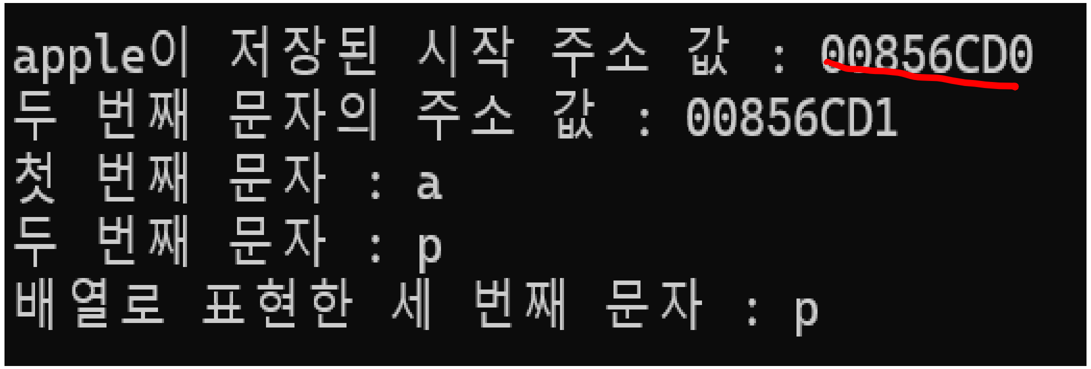
%p : 주소 출력하는 서식문자

“apple” 문자열



```
int main(void)
{
    printf("apple이 저장된 시작 주소 값 : %p₩n", "apple");
    printf("두 번째 문자의 주소 값 : %p₩n", "apple" + 1);
    printf("첫 번째 문자 : %c₩n", *"apple");
    printf("두 번째 문자 : %c₩n", *("apple" + 1));
    printf("배열로 표현한 세 번째 문자 : %c₩n", "apple"[2]);

    return 0;
}
```

이.5



```
apple이 저장된 시작 주소 값 : 00856CD0
두 번째 문자의 주소 값 : 00856CD1
첫 번째 문자 : a
두 번째 문자 : p
배열로 표현한 세 번째 문자 : p
```

(char *) 0x00856CD0

대입 str1, str2 배열가정

str1 = str2 (X) strcpy / strncpy

(,) (, ,)

비교

str1 == str2 (X) strcmp

연산(덧셈)

str2= str1+str2 (X) strcat / strncat

(,) (, ,)

13장 변수의 영역과 데이터 공유

C언어에서 사용영역(유효영역)에 따른 변수의 종류

① 지역변수 (로컬 변수)

{ } 내부에서만 유효한것, 함수의 매개변수에서만

초기화하지 X 쓰레기값.

② 전역변수 (글로벌 변수)

main 함수 중괄호 밖에 선언, 어디서든 접근 가능

초기화를 하지 X 0으로 자동으로 초기화

전역변수는 상수 초기화

③ 정적변수 (Static 변수)

초기화는 1번만 함.

함수가 종료되어도 메모리상에 남아있음.

{ } 사용이 가능

17장 사용자 정의 자료형

구조체 서로다른 자료형을 모아? 하나의 단일 자료형처럼 사용

```
struct 구조체이름
{
    구조체 멤버변수;
          ⋮
};
```

```
typedef struct student
        {
            ⋮
        }student_t;
```

```
struct student s1;

    student_t s1;
```

마이크로 프로세서 실습

마이크로 컨트롤러 실습

AVR
ARM

MPU	MCU
CPU를 단일 IC 칩에 만든 반도체	CPU + α (메모리 등)
범용목적	특별한 목적에 사용

AUR - ATMEGA128A

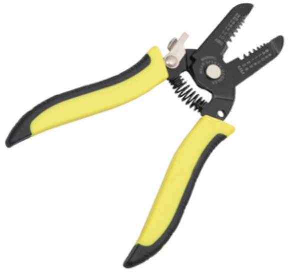
8-bit MCU

1명당

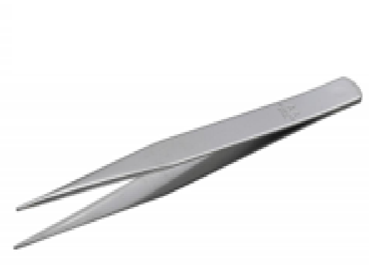
빵판 1EA

AUR 1EA

케이블 1EA


1EA

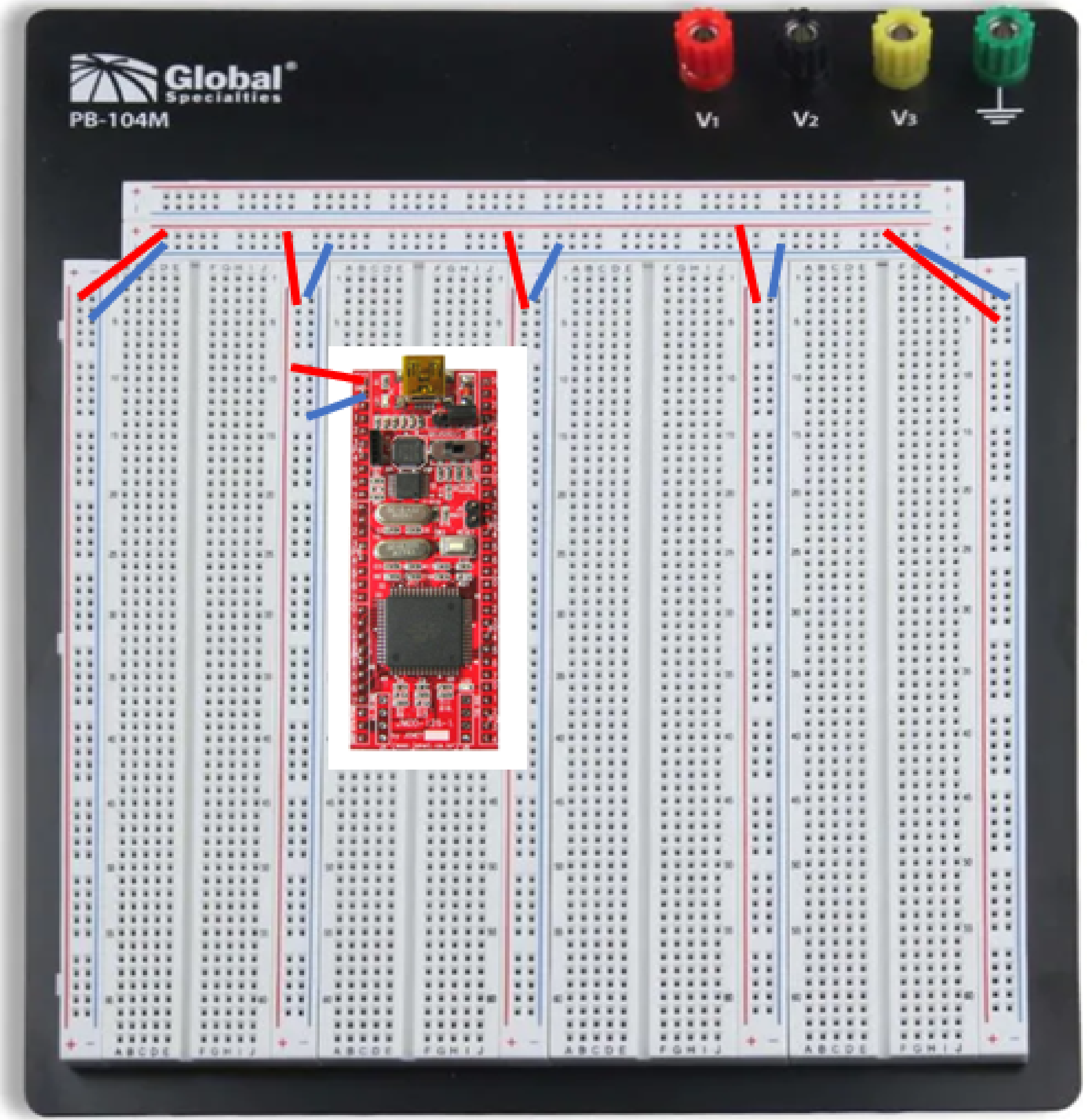
2명당

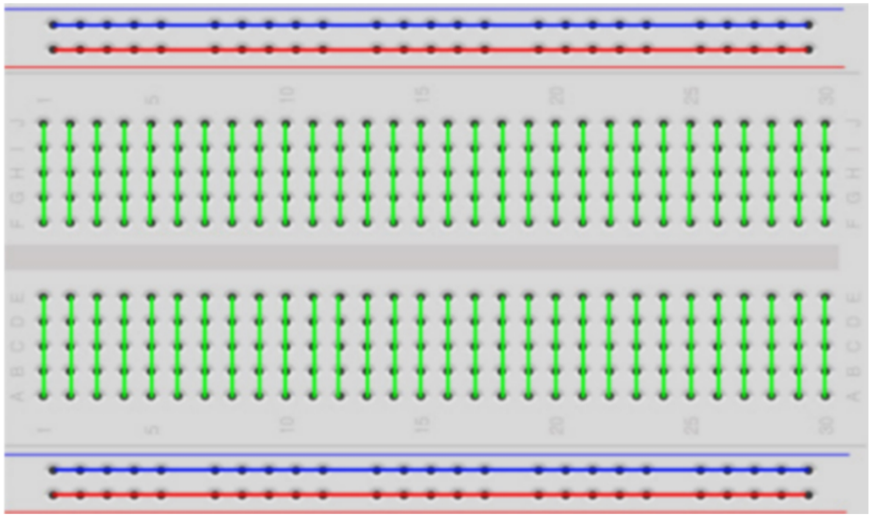

1EA

니퍼 1EA

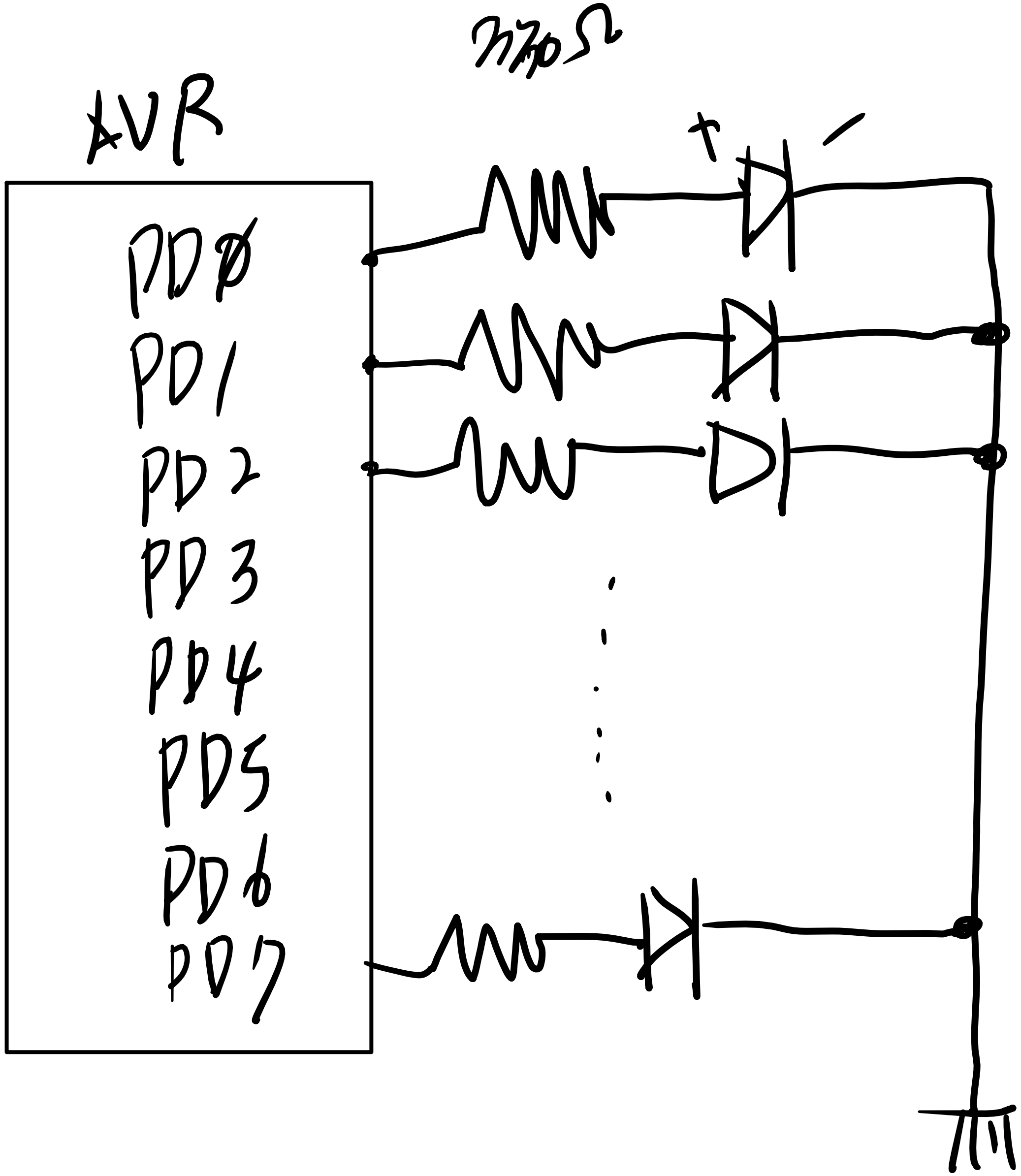
4명당

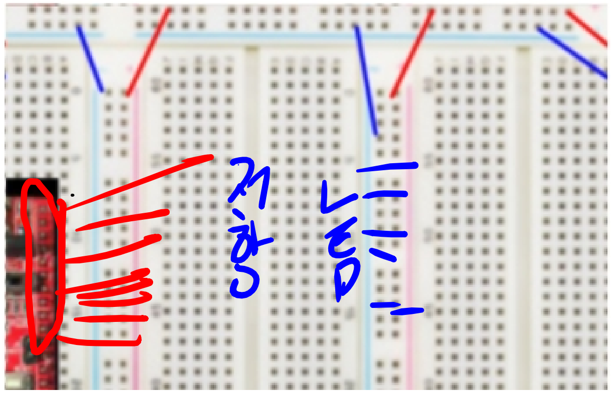
드라이버 1EA

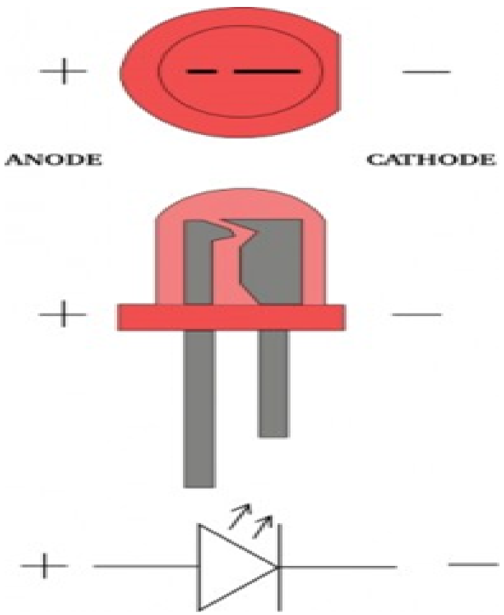

Global Specialties
PB-104M
V1
V2
V3

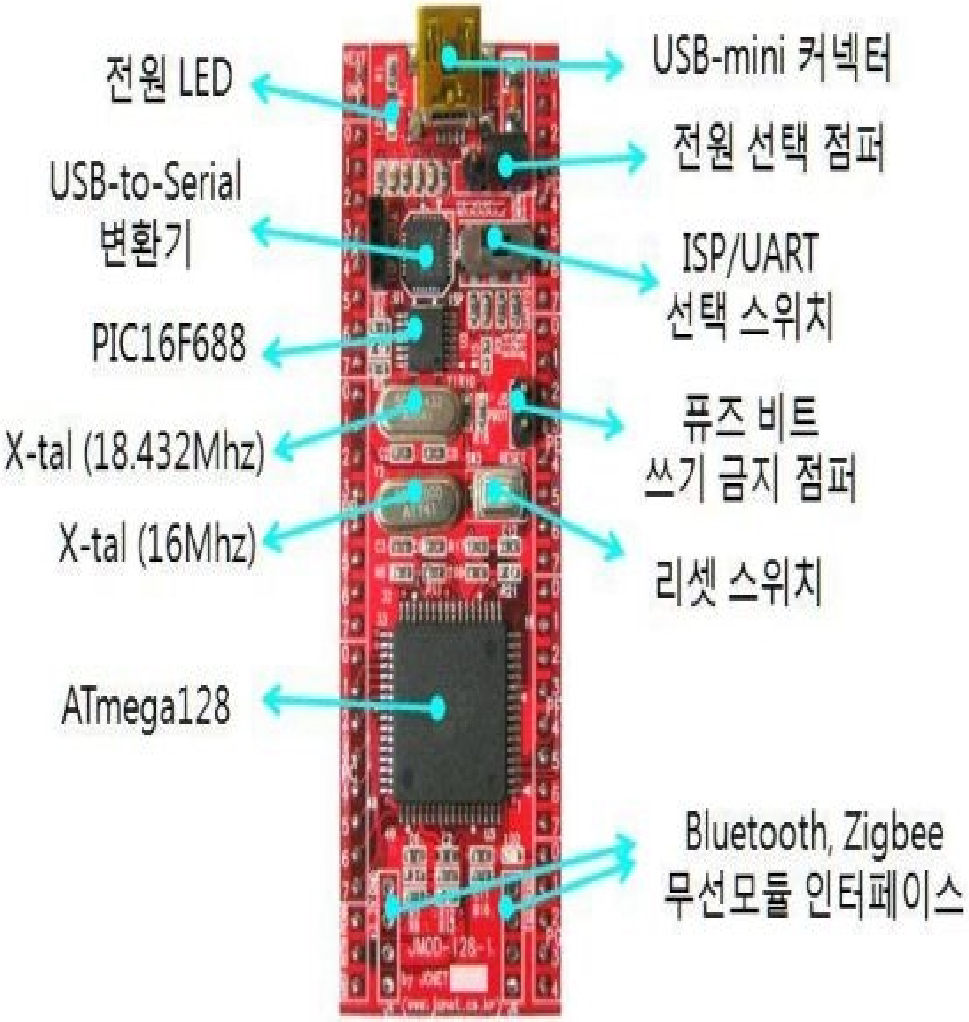

1
5
10
15
20
25
30
F G H I J
A B C D E

LED 8EA , 330Ω 저항 8EA
(사람당)

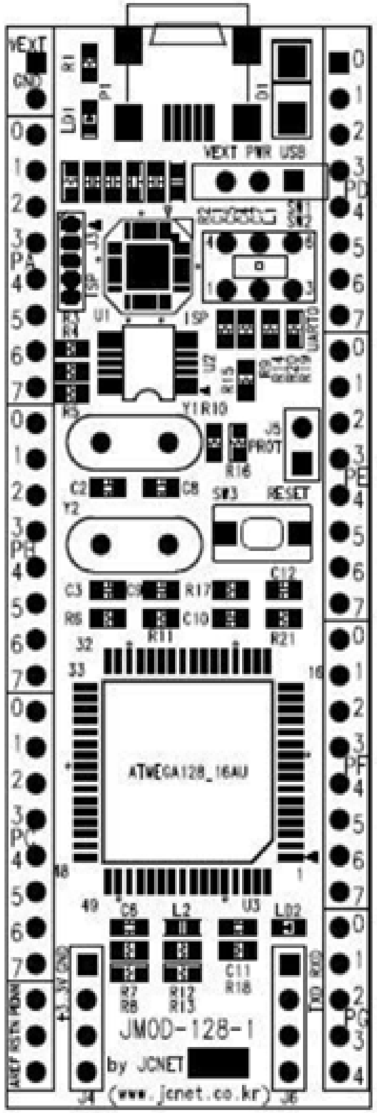








1	VEXT
2	GND
3	PA0, AD0
4	PA1, AD1
5	PA2, AD2
6	PA3, AD3
7	PA4, AD4
8	PA5, AD5
9	PA6, AD6
10	PA7, AD7
11	PB0, SS*
12	PB1, SCK
13	PB2, MOSI
14	PB3, MISO
15	PB4, OC0
16	PB5, OC1A
17	PB6, OC1B
18	PB7, OC2, OC1C
19	PC0, A8
20	PC1, A9
21	PC2, A10
22	PC3, A11
23	PC4, A12
24	PC5, A13
25	PC6, A14
26	PC7, A15
27	PEN*
28	RESET*
29	AREF



PD0, SCL, INT0	58
PD1, SDA, INT1	57
PD2, RXD1, INT2	56
PD3, TXD1, INT3	55
PD4, ICP1	54
PD5, XCK1	53
PD6, T1	52
PD7, T2	51
PE0, RXD0, PDI	50
PE1, TXD0, PDO	49
PE2, XCK0, AIN0	48
PE3, OC3A, AIN1	47
PE4, OC3B, INT4	46
PE5, OC3C, INT5	45
PE6, T3, INT6	44
PE7, ICP3, INT7	43
PF0, ADC0	42
PF1, ADC1	41
PF2, ADC2	40
PF3, ADC3	39
PF4, ADC4, TCK	38
PF5, ADC5, TMS	37
PF6, ADC6, TDO	36
PF7, ADC7, TDI	35
PG0, WR*	34
PG1, RD*	33
PG2, ALE	32
PG3, TOSC2	31
PG4, TOSC1	30

GPIO

(General purpose Input Output)

Port A ~ Port G 53개

(A~F, 8EA), 5EA

범용 입출력 PIN 53개

유저가 마음대로 입/출력을 정해서

사용하는것

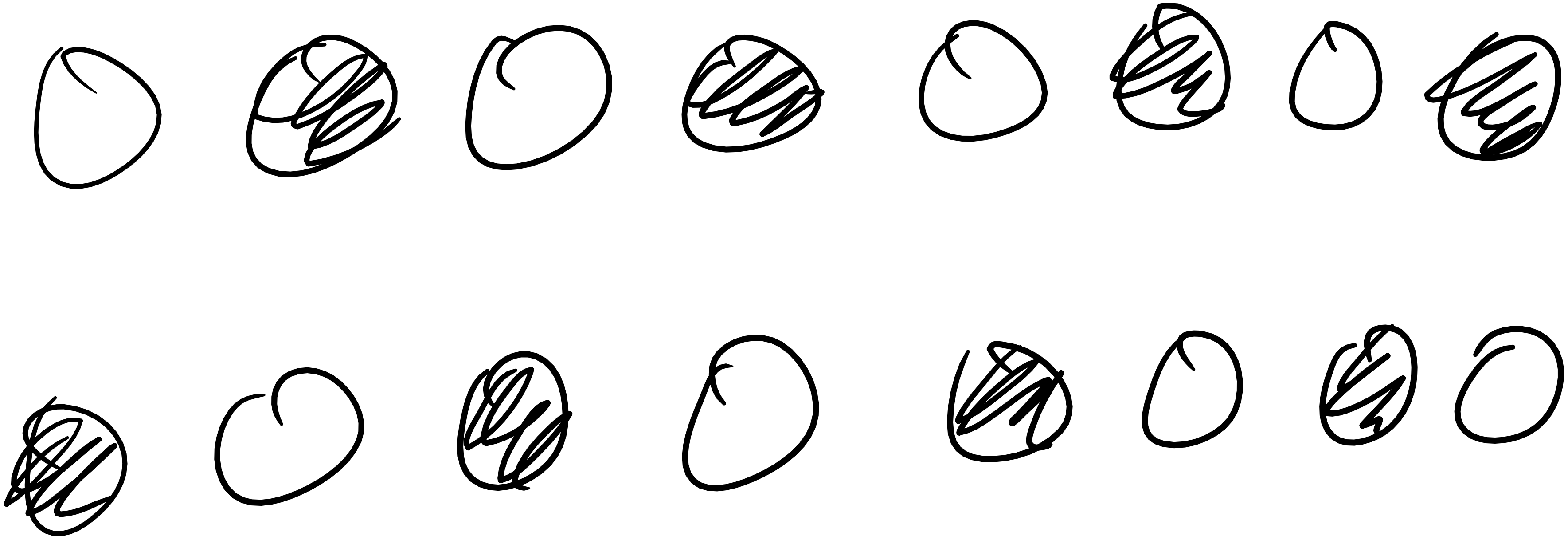
GPIO 레지스터

PortD LED 8EA 8bit

DDRx Register(Data Direction Register)

1bit 0: Input
1: Output

DDRD = 0xff;
= 0b 11111111;



폰 노이만 vs 하버드

데이터 메모리
프로그램 메모리

1개 2개