

14강

C_PROGRAMMING



포인터

❖ 정의

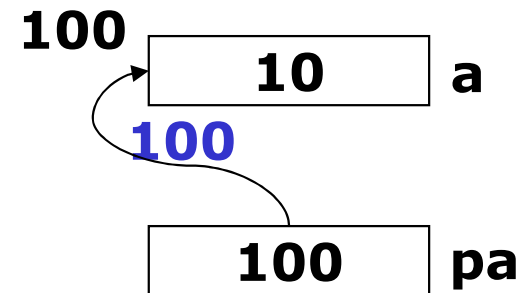
- 주기억장치의 특정 주소를 받는 변수
- 크기는 4byte(0x0012a047)

❖ 형식

- 자료형 *포인터 변수명;
- 자료형은 포인터변수와 연결할 변수와 같은 자료형이어야 한다.

❖ 사용예

```
#include <stdio.h>
main()
{
    int a=10;    //a변수 메모리 할당
    int *pa;     //pa변수 메모리 할당
    pa=&a;       //a의 주소를 pa에 저장
}
```

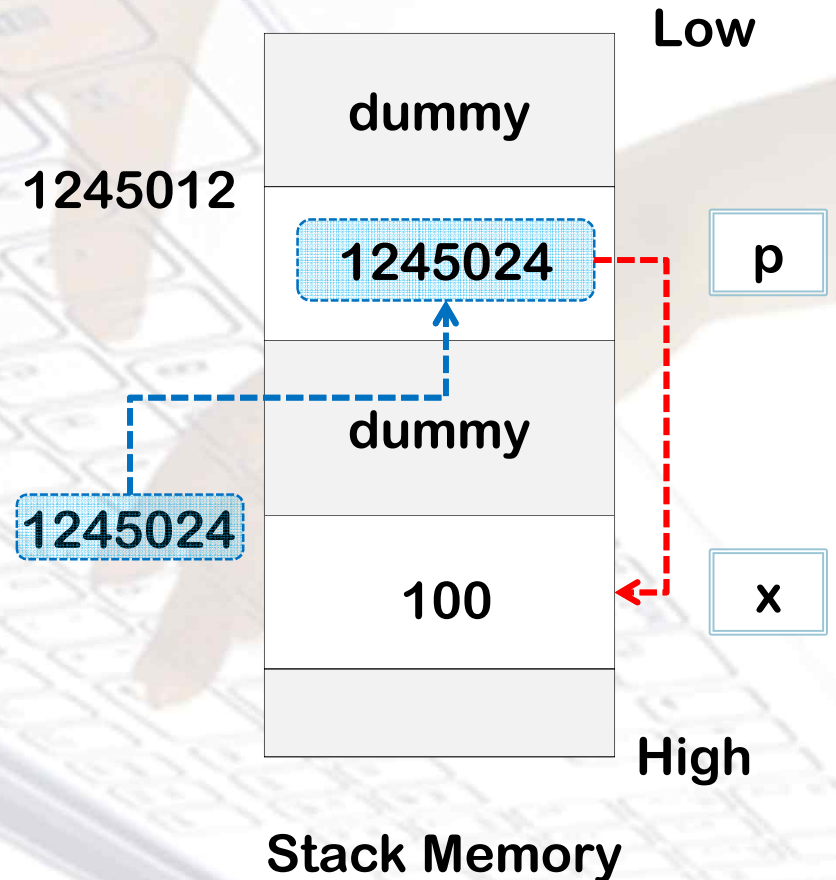


실습예제 1

❖ 포인터 선언 및 활용

```
#include <stdio.h>
int main()
{
    int x = 100;
    int *p;
    //포인터 변수 p 선언

    p = &x;
    //변수 x의 주소를 p에 연결
    printf("p = %d\n", p );
    //p의 값 출력
    printf("*p = %d\n", *p );
    //p의 주소에 접근하여 x의 값 출력
    return 0;
}
```



예제

```
#include <stdio.h>
void main()
{
    char ch1 = 'A', ch2;
    char *ptr;

    ptr = &ch1;
    ch2 = *ptr;
    printf("변수 ch1의 주소는 %p번지이다.", &ch1);
    printf("변수 ptr의 내용은 %u번지이다.", ptr);
    printf("변수 *ptr의 내용은 %c이다.", *ptr);
    printf("변수 ch2의 내용은 '%c'이다. \n", ch2);
}
```

예제

```
#include <stdio.h>
void main()
{
    int num1 = 10, num2;
    int *ptr;

    num1++;
    num2 = 5;
    ptr = &num2;
    *ptr = *ptr + num1 + num2;
    printf("num1=%d, num2=%d, *ptr=%d\n", num1, num2, *ptr);
}
```

예제(skip)

```
#include <stdio.h>
void main()
{
    int * ptr;
    *ptr = 10;
    printf(" %d\n",*ptr);
}
```

예제

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

```
void main()
```

```
{
```

```
    char  vc = 'A', *pc;
```

```
    int   vi = 123, *pi;
```

```
    pc = &vc; /* 문자형 포인터 */
```

```
    pi = &vi; /* 정수형 포인터 */
```

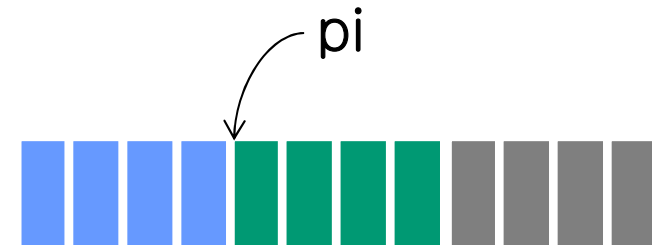
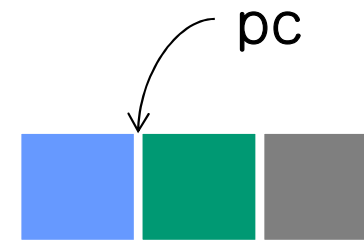
```
    printf("문자형 포인터: ");
```

```
    printf("pc-1=%u번지 pc=%u번지 pc+1=%u번지\n", pc-1, pc, pc+1);
```

```
    printf("정수형 포인터: ");
```

```
    printf("pi-1=%u번지 pi=%u번지 pi+1=%u번지\n", pi-1, pi, pi+1);
```

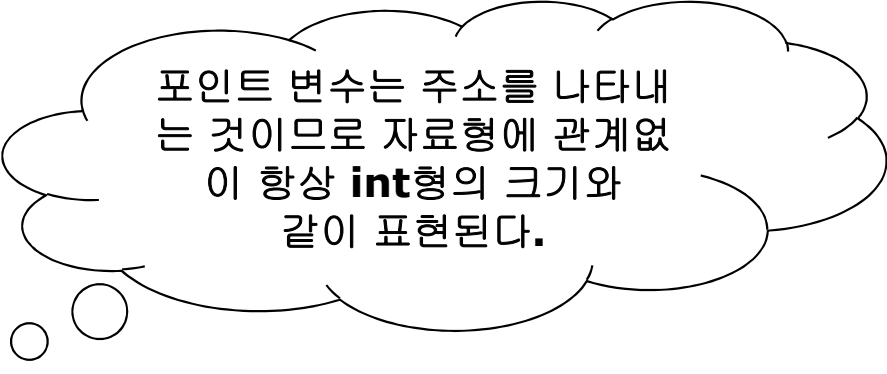
```
}
```



포인터변수의 크기

❖ 예제

```
#include <stdio.h>
main()
{
    char a='A';
    float b=12.345f;
    char *pa;
    float *pb;
    pa=&a;
    pb=&b;
    printf("문자형 변수의 크기 : %d \Wn", sizeof(a));
    printf("실수형 변수의 크기 : %d \Wn", sizeof(b));
    printf("문자형 포인터 변수의 크기 : %d \Wn", sizeof(pa));
    printf("실수형 포인터 변수의 크기 : %d \Wn", sizeof(pb));
}
```



포인터 변수는 주소를 나타내는 것이므로 자료형에 관계없이 항상 **int**형의 크기와 같이 표현된다.

Quiz

```
#include <stdio.h>
void main(void)
{
    int i=5, j=10;
    int *p=&i, *q=&j;

    printf("(1) i = %d, j = %d\n", i, j);
    printf("(2) &i = %p, &j = %p\n", &i, &j);
    printf("(3) p = %p, q = %p\n", p, q);
    printf("(4) *p = %d, *q = %d\n", *p, *q);
    printf("(5) p == &i ? %d\n", p == &i);
}
```

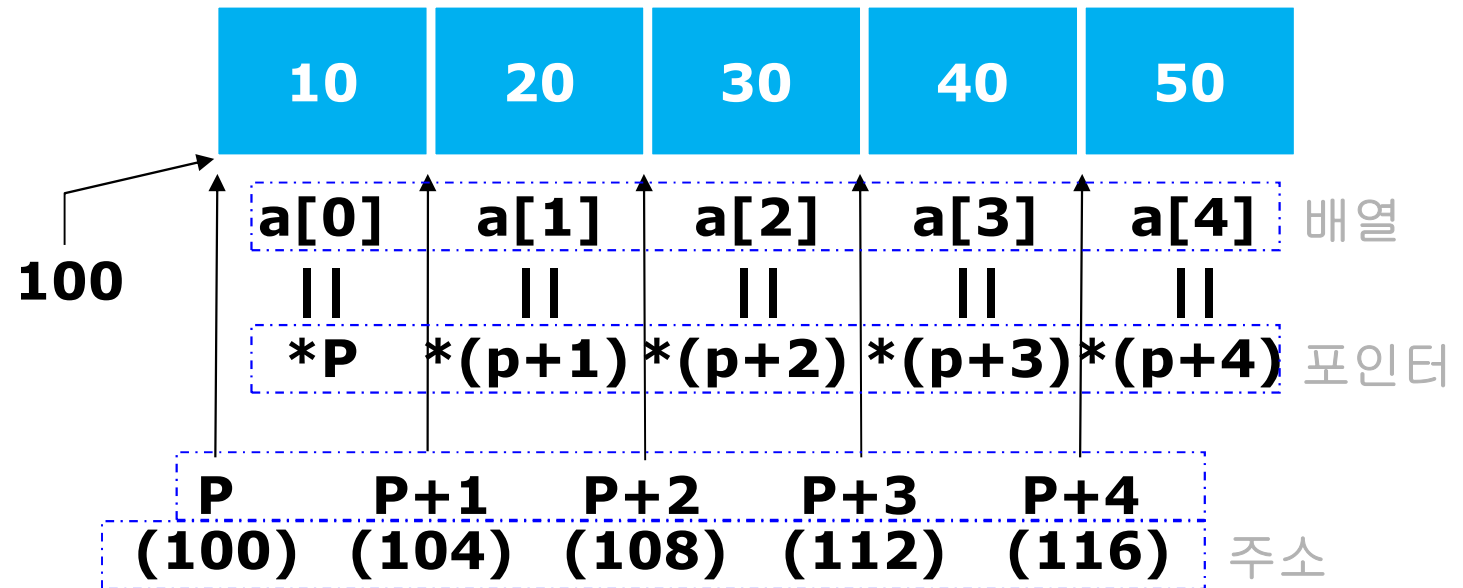
포인터와 배열

배열 명은 그 배열의 시작 주소를 의미한다(포인터 상수).

포인터의 경우 배열의 주소를 받아들이므로 포인터에 배열을 연결하여 사용할 수 있다.

```
int a[5]={10,20,30,40,50};
```

```
int *p = a;
```



예제

```
#include <stdio.h>
main()
{
    int i;
    int a[5]={10, 20, 30, 40, 50};
    for(i=0;i<5;i++)
    {
        printf("배열의 주소 %d, 배열의 값 %d\n", a+i, *(a+i));
    }
}
```

예제

```
#include <stdio.h>
main()
{
    int a[5]={10, 20, 30, 40, 50};
    int i, *pa;
    pa=a;

    for(i=0;i<5;i++)
        printf("배열의 주소 %d, 배열의 값 %d\n", pa+i, *(pa+i));
    printf("\n");

    for(i=0;i<5;i++)
        printf("배열의 주소 %d, 배열의 값 %d\n", pa++, *pa);
}
```

포인터를 이용하여
배열의 주소와 값을
구할수 있다.

예제

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

```
void main(void)
```

```
{
```

```
    int a=10, b[5]={1,2,3,4,5}, *p;
```

```
    p = &a;
```

```
    printf("(1) a=%d, *p=%d\n", a, *p);
```

```
    p = b;
```

```
    printf("(2) a=%d, *p=%d, *(p+1)=%d\n", a, *p, *(p+1));
```

```
    (*p)++;
```

```
    printf("(3) a=%d, *p=%d, *(p+1)=%d\n", a, *p, *(p+1));
```

```
}
```

Quiz

❖ 문자열에 저장되어 있는 단어들의 첫 영문자만 대문자로 변환하시오.

(단. 숫자, 특수문자 입력 시 무시하고 포인터 변수를 이용할 것.)

- Ex>test string _Quiz
- Test String _Quiz

포인터와 함수

❖ Call by value

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

```
void ex_func(int sa, int sb){
```

```
    sa += 5;
```

```
    sb += 5;
```

```
    printf("값1 : %d %d\n", sa, sb);
```

```
}
```

```
main()
```

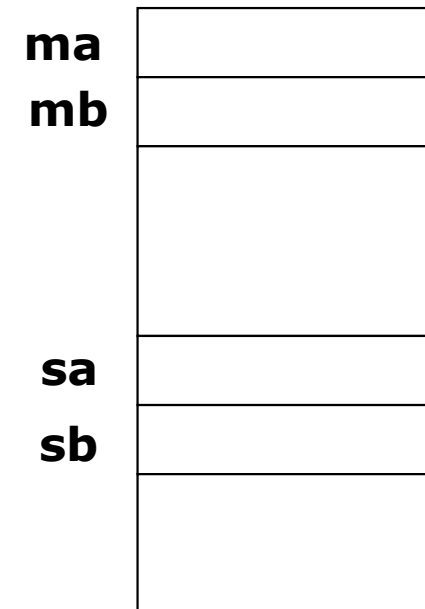
```
{
```

```
    int ma=100, mb=200;
```

```
    ex_func(ma, mb);
```

```
    printf("값2 : %d %d\n", ma, mb);
```

```
}
```



포인터와 함수

❖ Call by reference

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

```
void int_swap(int *x, int *y)
```

```
{
```

```
    int tmp;
```

```
    tmp=*x;
```

```
    *x=*y;
```

```
    *y=tmp;
```

```
}
```

```
main()
```

```
{
```

```
    int a=100, b=200;
```

```
    printf("교환전 : a=%d, b=%d\n", a, b);
```

```
    int_swap(&a, &b);
```

```
    printf("교환후 : a=%d, b=%d\n", a, b);
```

```
}
```

a		1245052
b		1245048
x	1245052	
y	1245048	
tmp		

예제

```
#include <stdio.h>

void convertToUppercase(char *s){
    while (*s != '\0') {
        if (*s >= 'a' && *s <= 'z')
            *s = *s - 32; /* ASCII 대문자 변환 */
        ++s; /* 다음 문자로 포인터 증가 */
    }
}

void main(void) {
    char string[ ]="characters";
    printf("변환 전의 스트링 : %s\n", string);
    convertToUppercase(string);
    printf("변환 후의 스트링 : %s\n", string);
}
```

Quiz

❖ 문자열에 저장되어 있는 단어들의 첫 영문자만 대문자로 변환하시오.

(단. 숫자, 특수문자 입력 시 무시하고 Call by reference 방식의 함수를 이용할 것.)

- Ex>test string _Quiz
- Test String _Quiz

예제

- ❖ 하나의 숫자를 입력 받아 1~n까지 합을 구하는 함수를 만드시오

```
#include <stdio.h>
int sumf(int n);
main()
{
    int num, ret;
    scanf("%d", &num);
    ret=sumf(num);
    printf("합은 %d입니다.", ret);
}
int sumf(int n)
{
    int i, sum=0;
    for(i=1;i<=n;i++)
        sum+=i;
    return sum;
}
```

문자열 예제

❖ 문자 하나를 입력 받아 소문자를 대문자로 변환

```
#include <stdio.h>
#define UPPERMASK    0x5f
char uppercase(char ch);
main()
{
    char    ch;
    scanf("%c", &ch);
    ch=uppercase(ch);
    printf("변환된 내용은 %c", ch);
}
char uppercase(char ch)
{
    if(ch>='a'&&ch<='z')
        ch&=UPPERMASK;
    return ch;
}
```

예제

❖ 두수를 입력 받아 큰 수를 출력하시오

```
#include <stdio.h>
int maxf(int n1, int n2);
main()
{
    int num1, num2, max;
    scanf("%d%d", &num1, &num2);
    max=maxf(num1, num2);
    printf("max = %d", max);
}
int maxf(int n1, int n2)
{
    int max;
    (n1>n2)?(max=n1):(max=n2);
    return max;
}
```