

# EC C언어 스터디

-6강-



포인터

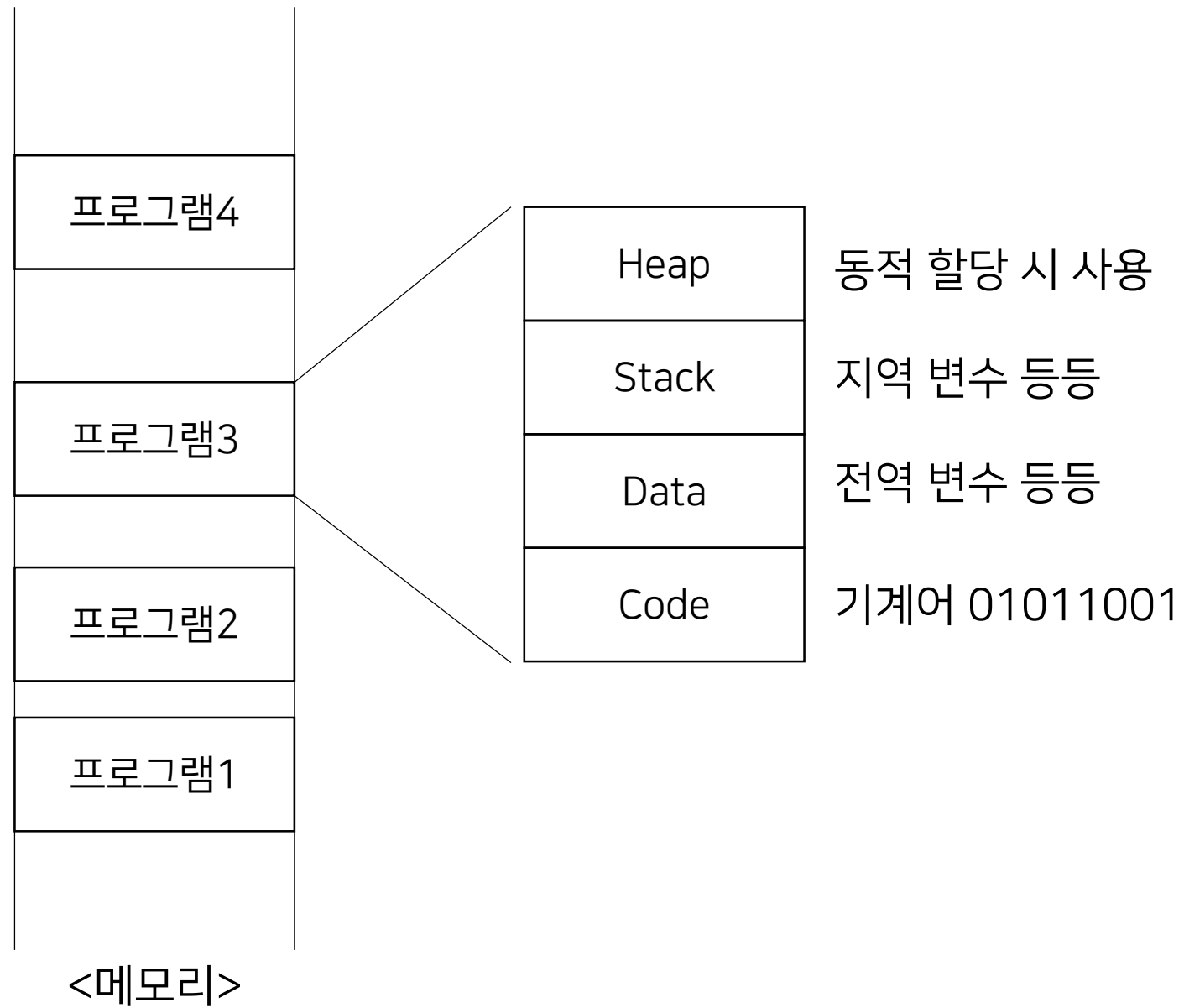


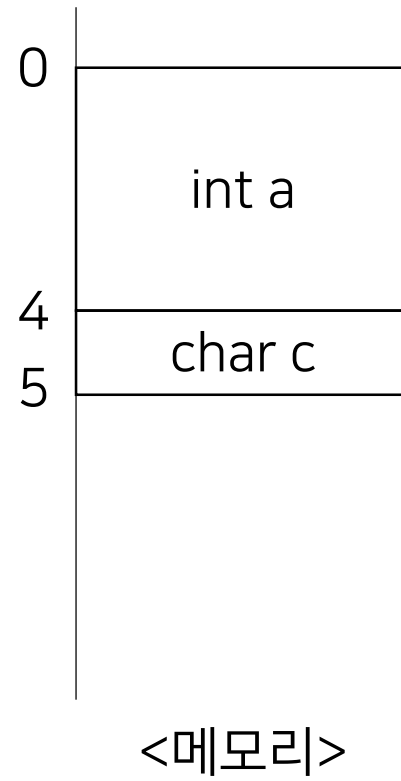
동적 할당





포인터





```
자료형* 변수명;  
int* p = NULL;  
자료형 *변수명1, *변수명2;
```

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

```
int main()
```

```
{
```

```
    int a = 1;
```

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

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

```
    *p = 2;
```

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

```
}
```

a=1    \*p=1

a=2    \*p=2

```
#include <stdio.h>

void f(int a)
{
    a = 5;
}

int main()
{
    int a = 1;
    printf("a = %d\n", a); // 1
    f(a);
    printf("a = %d\n", a); // 1
}
```

```
#include <stdio.h>

void f(int *a)
{
    *a = 5;
}

int main()
{
    int a = 1;
    printf("a = %d\n", a); // 1
    f(&a);
    printf("a = %d\n", a); // 5
}
```



```
#include <stdio.h>

void swap(int* a, int* b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main()
{
    int a = 1, b = 4;
    printf("a = %d, b = %d\n", a, b);
    swap(&a, &b);
    printf("a = %d, b = %d\n", a, b);
}
```

```
#include <stdio.h>

int main()
{
    int a;
    scanf("%d", &a);
}
```

```
#include <stdio.h>

int main()
{
    int arr[5] = { 1,2,3,4,5 };
    for (int i = 0; i < 5; i++)
    {
        printf("%p ", &arr[i]); // 00AFF754 00AFF758 00AFF75C 00AFF760 00AFF764
    }
    printf("\n&arr[0] = %p\narr = %p", &arr[0], arr);
    // &arr[0] = 00AFF754
    // arr = 00AFF754
}
```

```
#include <stdio.h>

int main()
{
    int arr[5] = { 1,2,3,4,5 };

    int* p = arr;
    for (int i = 0; i < 5; i++)
    {
        printf("%d ", *(p + i)); // 1 2 3 4 5
    }
}
```

$\text{arr}[i] = \text{*(arr+i)}$

```
#include <stdio.h>

void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

//void reverse(int arr[], int len)
void reverse(int* arr, int len)
{
    for (int i = 0; i < len / 2; i++)
    {
        swap(&arr[i], &arr[len - i - 1]);
    }
}
```

```
int main()
{
    int arr[5] = { 1,2,3,4,5 };
    for (int i = 0; i < 5; i++)
    {
        printf("%d ", *(arr + i));
    }
    printf("\n");
    reverse(arr, 5); // arr를 집어넣음, 즉 주소를 넘겨줌
    for (int i = 0; i < 5; i++)
    {
        printf("%d ", *(arr + i));
    }
}
```

1	2	3	4	5
5	4	3	2	1

```
#include <stdio.h>

int main()
{
    char s[100] = "";
    scanf("%s", s);
}
```

```
#include <stdio.h>

int main()
{
    int a = 10, b = 20, c = 30;
    int* arr[3] = { &a, &b, &c }; // int형 포인터 배열 선언

    for (int i = 0; i < 3; i++)
    {
        printf("%d\n", *arr[i]);
    }
}
```

```
#include <stdio.h>

int main()
{
    int* numPtr1;    // 단일 포인터 선언
    int** numPtr2;   // 이중 포인터 선언
    int num1 = 10;

    numPtr1 = &num1;    // num1의 메모리 주소 저장
    numPtr2 = &numPtr1; // numPtr1의 메모리 주소 저장

    printf("%d\n", **numPtr2);    // 포인터를 두 번 역참조하여 num1의 메모리 주소에 접근
}
```



```
자료형 (*포인터이름)[가로크기];  
int (*p)[4];
```

```
#include <stdio.h>

int main()
{
    int arr[2][3] =
    {
        {1,2,3},
        {4,5,6}
    };

    int (*p)[3] = arr;

    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            printf("%d ", (*(p + i) + j));
        }
        printf("\n");
    }
}
```

$$\begin{aligned} \text{arr}[i][j] &= *((\text{arr} + i) + j) \\ &= *(\text{arr}[i] + j) = (*(\text{arr} + i))[j] \end{aligned}$$

1	2	3
4	5	6

```
int (*p)[3];  
int *p[3];
```

반환값자료형 (\*함수포인터이름)(인자의 자료형, 인자2의 자료형...);  
`void (*fp)();` // 반환값과 매개변수가 없는 함수 포인터 `fp` 선언

```
#include <stdio.h>

void hello()
{
    printf("Hello, world!\n");
}

void bonjour()
{
    printf("bonjour le monde!\n");
}

int main()
{
    void (*fp)(); // 반환값과 매개변수가 없는 함수 포인터 fp 선언

    fp = hello;   // hello 함수의 메모리 주소를 함수 포인터 fp에 저장
    fp();         // Hello, world!: 함수 포인터로 hello 함수 호출

    fp = bonjour; // bonjour 함수의 메모리 주소를 함수 포인터 fp에 저장
    fp();         // bonjour le monde!: 함수 포인터로 bonjour 함수 호출

    return 0;
}
```

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

```
int add(int a, int b)
{
    return a + b;
}
```

```
int sub(int a, int b)
{
    return a - b;
}
```

```
int mul(int a, int b)
{
    return a * b;
}
```

```
int div(int a, int b)
{
    // TODO : hand
    return a / b;
}
```

```
int main()
```

```
{
    int (*cal)(int, int) = NULL;
```

```
    int c;
```

```
    while (1)
```

```
    {
```

```
        printf("**사칙연산*\n숫자를 입력하세요\n");
```

```
        printf("1:더하기 2:빼기 3:곱하기 4:나누기\n");
```

```
        printf("입력 : ");
```

```
        scanf("%d", &c);
```

```
        if (c >= 1 && c <= 4) break;
```

```
        else printf("다시 입력하세요\n\n");
```

```
    }
```

```
    switch (c)
```

```
    {
```

```
        case 1:
```

```
            cal = add;
```

```
            break;
```

```
        case 2:
```

```
            cal = sub;
```

```
            break;
```

```
        case 3:
```

```
            cal = mul;
```

```
            break;
```

```
        case 4:
```

```
            cal = div;
```

```
            break;
```

```
    }
```

```
    int a, b;
```

```
    printf("정수 두개를 입력하세요 : ");
```

```
    scanf("%d%d", &a, &b);
```

```
    int res = cal(a, b);
```

```
    printf("결과는 %d", res);
```

```
}
```

```
int main()
{
    int (*cal[4])(int, int) = {add, sub, mul, div}; // 함수 포인터 배열

    int c;
    while (1)
    {
        printf("*사칙연산*\n숫자를 입력하세요\n");
        printf("1:더하기 2:빼기 3:곱하기 4:나누기\n");
        printf("입력 : ");
        scanf("%d", &c);
        if (c >= 1 && c <= 4) break;
        else printf("다시 입력하세요\n\n");
    }

    int a, b;
    printf("정수 두개를 입력하세요 : ");
    scanf("%d%d", &a, &b);

    int res = cal[c](a, b); // 사용
    printf("결과는 %d", res);
}
```



**동적 할당**



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

int main()
{
    int n;
    scanf("%d", &n);

    int* p;
    p = (int*)malloc(n * sizeof(int));

    for (int i = 0; i < n; i++)
    {
        scanf("%d", p + i);
    }
    for (int i = 0; i < n; i++)
    {
        printf("%d ", *(p + i));
    }
}
```

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

int main()
{
    int n;
    scanf("%d", &n);

    int* p;
    p = (int*)malloc(n * sizeof(int));

    free(p);
}
```



수고하셨습니다!