

Week 07 21.12.07. 07:30 PM

+

o

•

C Programming

Second-half period / CMD-C_Programming Study

Mentor : Jh9892

INDEX

- 💧 C Language - Dynamic Memory Handling
- 💧 C Language - Structure Pointer

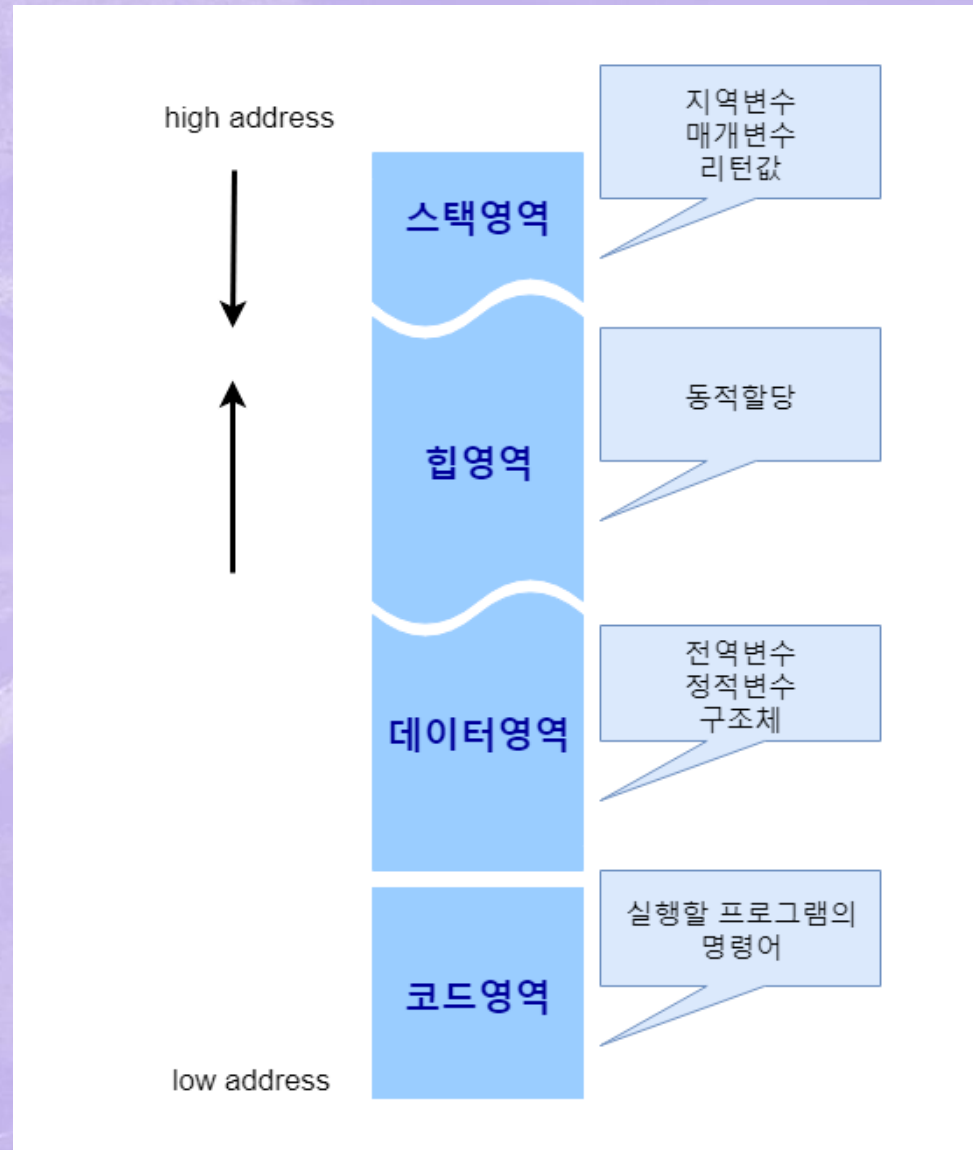
Dynamic Memory Handling

동적 메모리 할당

- *malloc()*
- *calloc()*
- *realloc()*
- *free()*

Dynamic Memory Handling

메모리 구조



Dynamic Memory Handling

include <stdlib.h>

*void *malloc(size_t size);*

*void free(void *ptr);*

```
1  <p>#include <stdio.h>
2  #include <stdlib.h>
3  int main() {
4      int size, i;
5      scanf("%d", &size);
6      int *arr=(int*)malloc(sizeof(int)*size);
7
8      for (i = 0; i < size; i++)
9          scanf("%d", &arr[i]);
10
11     for (i = 0; i < size; i++)
12         printf("arr[%d]=%d\n", i, arr[i]);
13     free(arr);
14 }
15
16
```

```
5
10 20 30 40 50
arr[0]=10
arr[1]=20
arr[2]=30
arr[3]=40
arr[4]=50
계속하려면 아무 키나 누르십시오 . . .
```


Dynamic Memory Handling

include <stdlib.h>

*void *calloc(size_t n, size_t size);*

*void free(void *ptr);*

```
5
calloc 0으로 초기화
arr[0]=0 arr[1]=0 arr[2]=0 arr[3]=0 arr[4]=0
10 20 30 40 50
arr[0]=10
arr[1]=20
arr[2]=30
arr[3]=40
arr[4]=50
계속하려면 아무 키나 누르십시오 . . .
```

```
<p>#include <stdio.h>
#include <stdlib.h>
int main() {
    int n, i;
    scanf("%d", &n);
    int *arr=(int*)calloc(n,sizeof(int));
    printf("calloc 0으로 초기화\n");

    for (i = 0; i < n; i++)
        printf("arr[%d]=%d ", i, arr[i]);
    printf("\n");

    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    for (i = 0; i < n; i++)
        printf("arr[%d]=%d\n", i, arr[i]);
    free(arr);
}
```

Dynamic Memory Handling

include <stdlib.h>

*void *realloc(void *memblock, size_t size);*

*void free(void *ptr);*

```
처음 크기 입력
3
1 3 10
arr[0]=1
arr[1]=3
arr[2]=10
다시 할당될 크기 입력
5
20 30
arr[0]=1
arr[1]=3
arr[2]=10
arr[3]=20
arr[4]=30
계속하려면 아무 키나 누르십시오 . . .
```

```
1  <p>#include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      int n,m,i;
6
7      printf("처음 크기 입력\n");
8      scanf("%d", &n);
9      int *arr=(int*)malloc(sizeof(int)*n);
10
11
12      for (i = 0; i < n; i++)
13          scanf("%d", &arr[i]);
14      for (i = 0; i < n; i++)
15          printf("arr[%d]=%d\n", i, arr[i]);
16
17      printf("다시 할당될 크기 입력\n");
18      scanf("%d", &m);
19      //realloc함수도 다시 할당
20      arr = (int*)realloc(arr, sizeof(int)*m);
21
22      for (i = n; i < m;i++)
23          scanf("%d", &arr[i]);
24      for (i = 0; i < m; i++)
25          printf("arr[%d]=%d\n", i, arr[i]);
26      free(arr);
27 }
```

Structure Pointer

실행 결과

이름: 홍길동

나이: 30

주소: 서울시 용산구 한남동

```
#include <stdio.h>
#include <string.h>    // strcpy 함수가 선언된 헤더 파일
#include <stdlib.h>    // malloc, free 함수가 선언된 헤더 파일

struct Person {       // 구조체 정의
    char name[20];     // 구조체 멤버 1
    int age;           // 구조체 멤버 2
    char address[100]; // 구조체 멤버 3
};

int main()
{
    struct Person *p1 = malloc(sizeof(struct Person)); // 구조체 포인터 선언, 메모리 할당

    // 화살표 연산자로 구조체 멤버에 접근하여 값 할당
    strcpy(p1->name, "홍길동");
    p1->age = 30;
    strcpy(p1->address, "서울시 용산구 한남동");

    // 화살표 연산자로 구조체 멤버에 접근하여 값 출력
    printf("이름: %s\n", p1->name);    // 홍길동
    printf("나이: %d\n", p1->age);    // 30
    printf("주소: %s\n", p1->address); // 서울시 용산구 한남동

    free(p1); // 동적 메모리 해제

    return 0;
}
```


Structure Pointer

```
struct Person *p1 = malloc(sizeof(struct Person));    // 구조체 포인터 선언, 메모리 할당
```

```
// 화살표 연산자로 구조체 멤버에 접근하여 값 할당
```

```
strcpy(p1->name, "홍길동");
```

```
p1->age = 30;
```

```
strcpy(p1->address, "서울시 용산구 한남동");
```

```
// 화살표 연산자로 구조체 멤버에 접근하여 값 출력
```

```
printf("이름: %s\n", p1->name);    // 홍길동
```

```
printf("나이: %d\n", p1->age);    // 30
```

```
printf("주소: %s\n", p1->address);    // 서울시 용산구 한남동
```

Structure Pointer

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

struct Data {
    char c1;
    int *numPtr;    // 포인터
};

int main()
{
    int num1 = 10;
    struct Data d1;    // 구조체 변수
    struct Data *d2 = malloc(sizeof(struct Data));    // 구조체 포인터에 메모리 할당

    d1.numPtr = &num1;
    d2->numPtr = &num1;

    printf("%d\n", *d1.numPtr);    // 10: 구조체의 멤버를 역참조
    printf("%d\n", *d2->numPtr);    // 10: 구조체 포인터의 멤버를 역참조

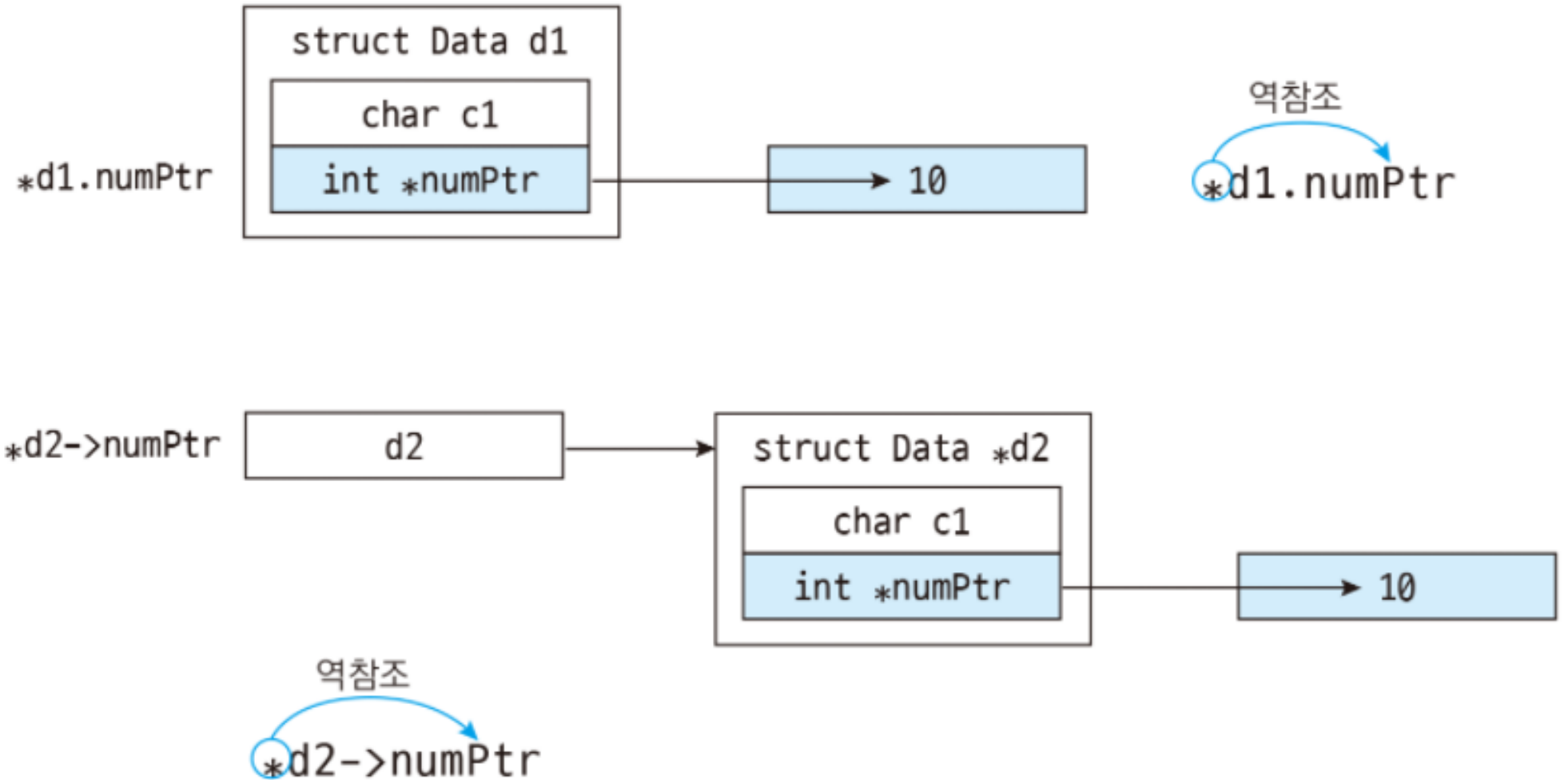
    d2->c1 = 'a';
    printf("%c\n", (*d2).c1);    // a: 구조체 포인터를 역참조하여 c1에 접근
    // d2->c1과 같음
    printf("%d\n", *(*d2).numPtr); // 10: 구조체 포인터를 역참조하여 numPtr에 접근한 뒤 다시 역참조
    // *d2->numPtr과 같음

    free(d2);

    return 0;
}
```

Structure Pointer

▼ 그림 49-1 구조체 멤버가 포인터일 때 역참조하기



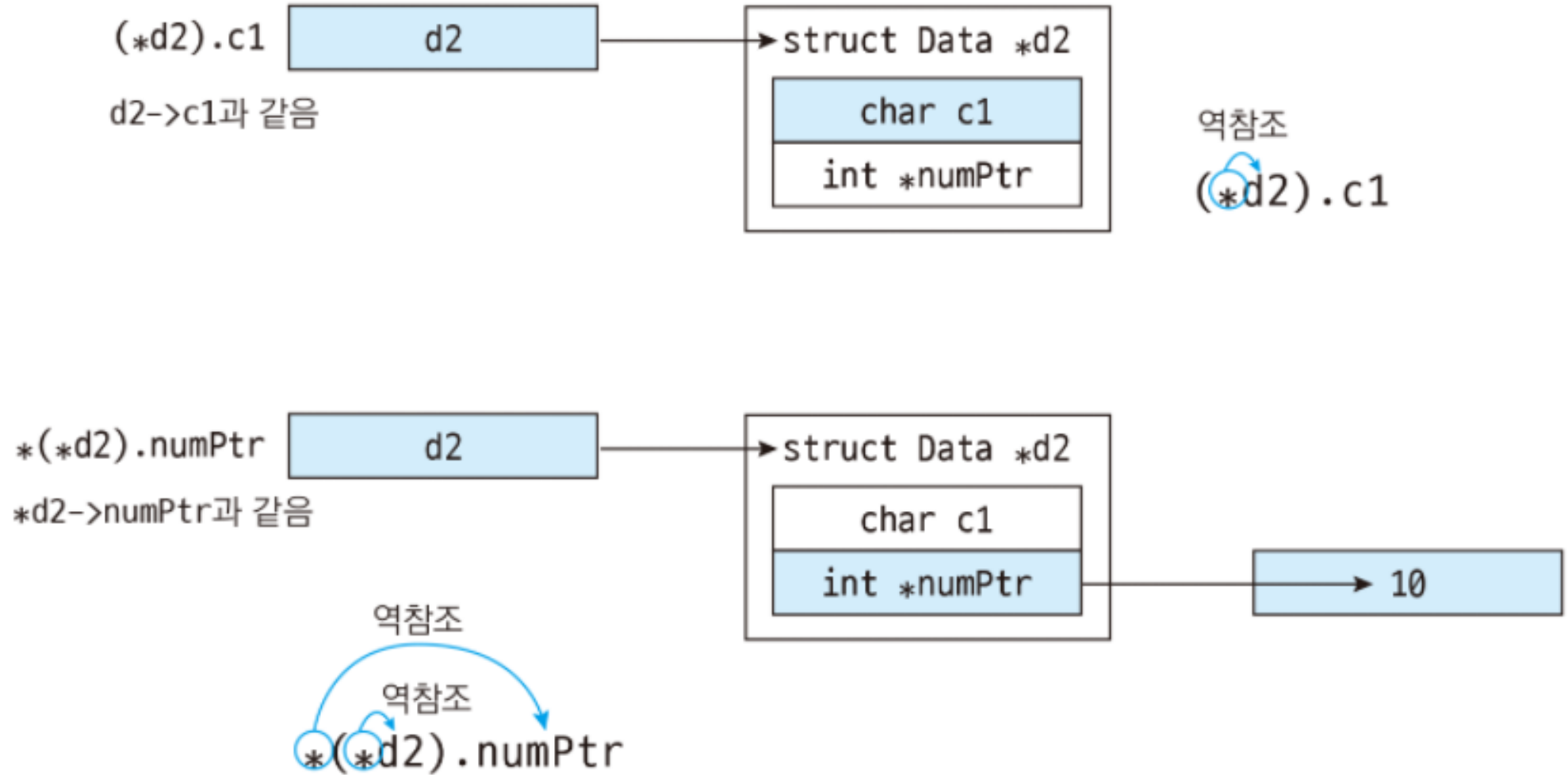
Structure Pointer

- (*구조체포인터).멤버
- *(*구조체포인터).멤버

```
d2->c1 = 'a';  
printf("%c\n", (*d2).c1);    // a: 구조체 포인터를 역참조하여 c1에 접근  
                             // d2->c1과 같음  
printf("%d\n", *(*d2).numPtr); // 10: 구조체 포인터를 역참조하여 numPtr에 접근한 뒤 다시 역참조  
                             // *d2->numPtr과 같음
```

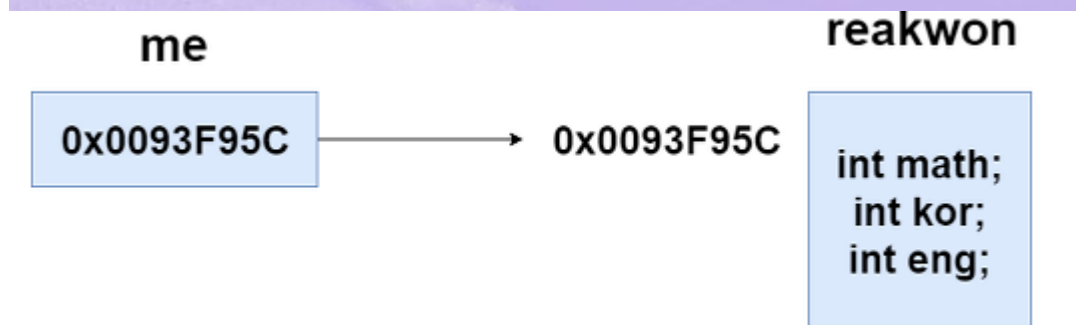
Structure Pointer

▼ 그림 49-2 구조체 포인터를 역참조한 뒤 괄호로 묶기



Structure Pointer

```
1  #include <stdio.h>
2
3
4  typedef struct student{
5      char *name;
6      int math;
7      int kor;
8      int eng;
9  } student;
10
11 float avg(student person);
12 student getHonorStudent(student me, student you);
13
14 int main() {
15     student reakwon = { "REAKWON",40,50,40 };
16     student seonmi = { "선미",90,95,100 };
17
18     student honorStudent = getHonorStudent(reakwon, seonmi);
19     printf("우등생은 %s입니다.\n", honorStudent.name);
20     return 0;
21 }
22
23 float avg(student person) {
24     return (person.math + person.kor + person.eng) / 3.0;
25 }
26 student getHonorStudent(student me, student you) {
27     if (avg(me) > avg(you))
28         return me;
29     if (avg(me) < avg(you))
30         return you;
31 }
```



Structure Pointer

```
1  #include <stdio.h>
2
3
4  typedef struct student{
5      char *name;
6      int math;
7      int kor;
8      int eng;
9  } student;
10
11
12  int main() {
13      student reakwon = { "REAKWON",40,50,40 };
14      student *me = &reakwon;
15
16      printf("me의 크기:%d\n", sizeof(me));
17      printf("reakwon의 주소:%p, me가 가리키는 주소:%p\n", &reakwon, me);
18      printf("\n");
19      printf("포인터를 통해서 값을 읽어오는 방법 1");
20      printf("수학:%d, 국어:%d, 영어:%d\n", (*me).math,(*me).kor,(*me).eng);
21
22      printf("\n");
23      printf("포인터를 통해서 값을 읽어오는 방법 2");
24      printf("수학:%d, 국어:%d, 영어:%d\n", me->math, me->kor, me->eng);
25
26      return 0;
27  }
```



- Finish -