C 언어 스터디 7주차

복습

```
43  int* GetAddress(int* arr, int value, int length)
44  {
45      while (length)
46      {
47          if (*arr == value)
48              return arr;
49
50          arr++;
51          length--;
52      }
53
54      return NULL;
55  }
```

```
56
57  void PrintArr(int* arr, int length)
58  {
59      while (length)
60      {
61          printf("%d\t", *arr);
62          arr++;
63          length--;
64      }
65      printf("\n");
66  }
```

복습

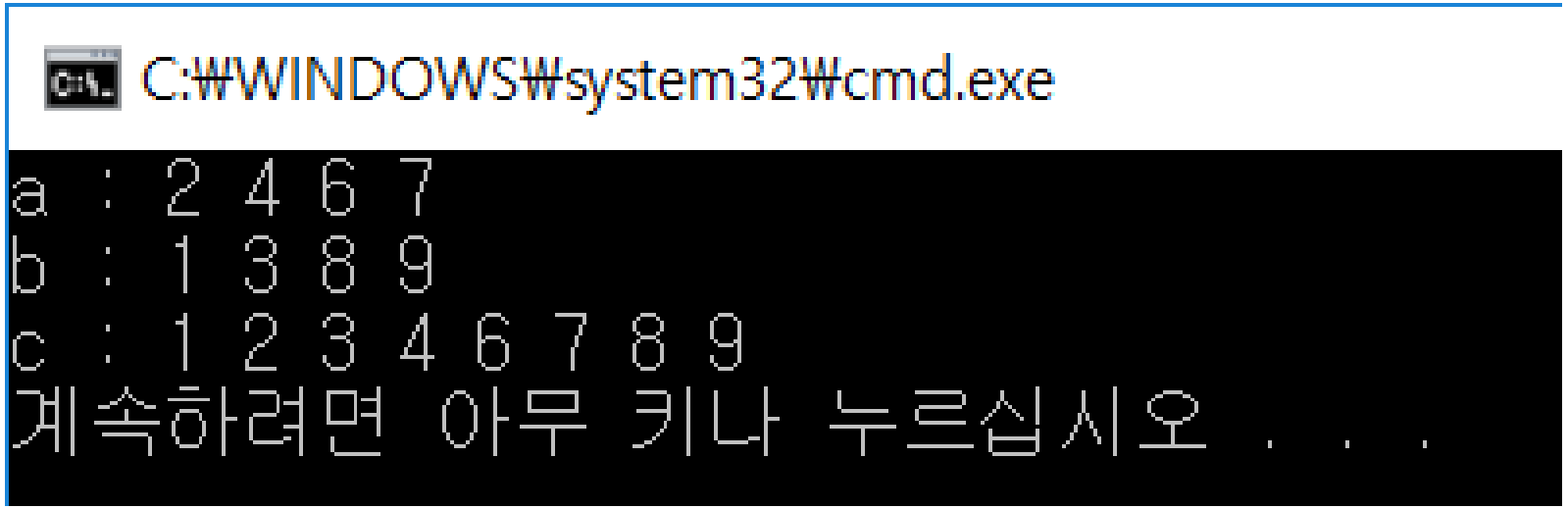
```
19  void Get_gcd_lcm(int x, int y, int* gcdP, int* lcmP)
20  {
21      *lcmP = x * y;
22      int tmp;
23
24      while (y)
25      {
26          tmp = x % y;
27          x = y;
28          y = tmp;
29      }
30
31      *gcdP = x;
32      *lcmP /= *gcdP;
33  }
```

1. 입력으로 두 수 x, y(x > y)가 들어온다.
2. y가 0이라면 x가 최대공약수이다.
3. x가 y로 나누어 떨어지면, y가 최대공약수이다.
4. 그렇지 않으면, x를 y로 나눈 나머지가 새로운 y가 되고, 원래 y가 새로운 x가 된 뒤 3번으로 돌아간다.

복습

- 2개의 정렬된 배열 a, b가 있다. a, b 배열의 크기는 같다.
- 이 두 배열과 크기를 매개변수로 받아서 하나의 배열로 합치는 merge 함수를 작성하여라.

  (새로운 배열 c도 매개변수로 받는다. c의 공간은 넉넉하게 존재한다고 가정한다.)



복습

```c
#include<stdio.h>
void merge(int* a, int* b, int* c, int length);

int main(void)
{
    int i;
    int a[] = { 2, 4, 6, 7 };
    int b[] = { 1, 3, 8, 9 };
    int c[8];

    merge(a, b, c, 4);
   printf("a : ");
   for (i = 0;i < 4;i++)
       printf("%d ", a[i]);
   printf("\n");

   printf("b : ");
   for (i = 0;i < 4;i++)
       printf("%d ", b[i]);
   printf("\n");

   printf("c : ");
   for (i = 0;i < 8;i++)
       printf("%d ", c[i]);
   printf("\n");
}
```

```c
void merge(int* a, int* b, int* c, int length)
{
    int i = 0, j = 0;
    int k = 0;

    while (i < length && j < length)
    {
        if (a[i] > b[j])
            c[k++] = b[j++];
        else
            c[k++] = a[i++];
    }

    while (i < length)
        c[k++] = a[i++];

    while (j < length)
        c[k++] = b[j++];
}
```

메모리 할당

프로그램의 메모리 할당

- 정적(static) 할당

  - 프로그램이 시작되기 전(컴파일 타임)에 미리 정해진 크기의 메모리 할당

  - ex) int number; int score[100];

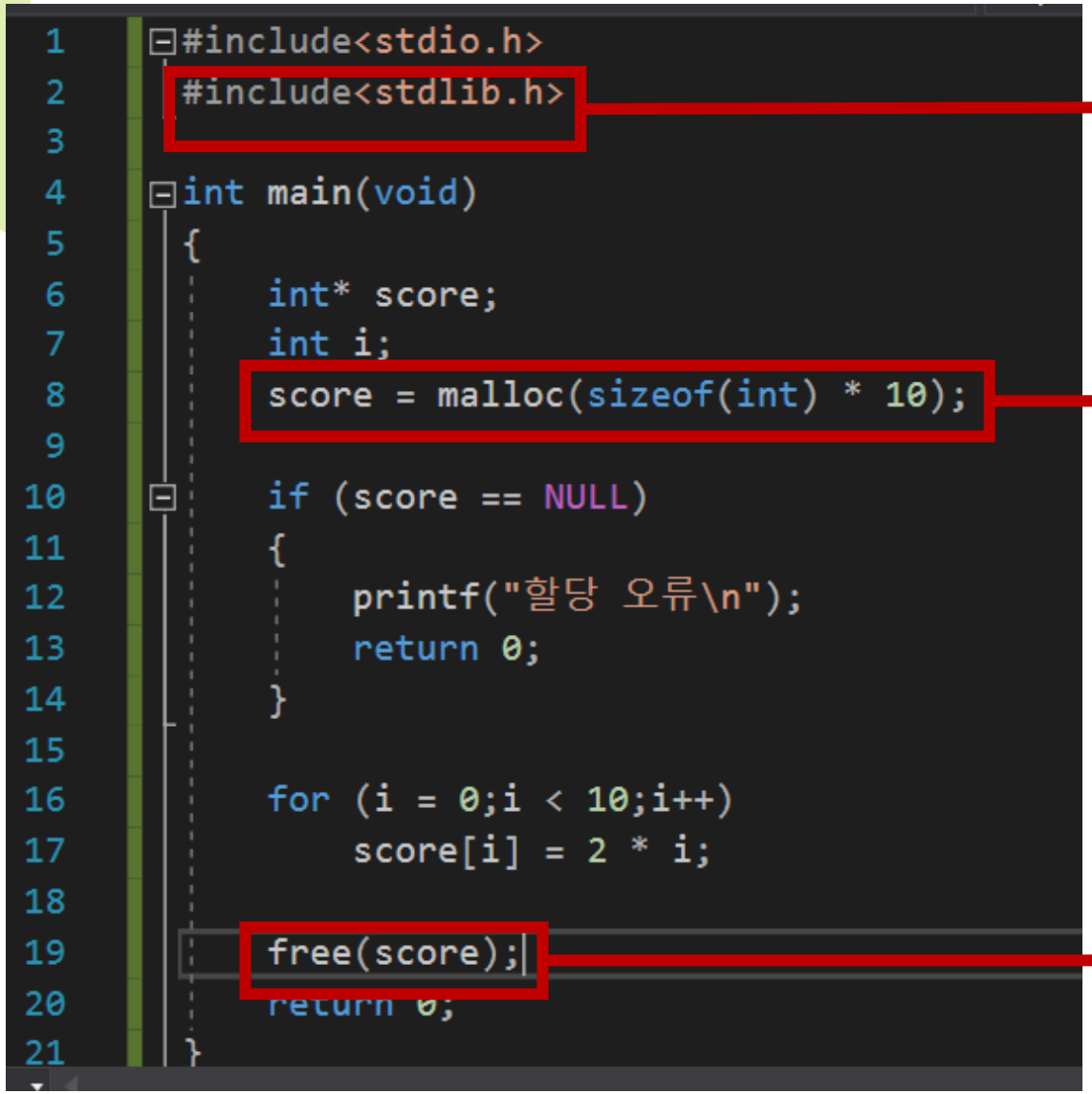
  - 크기 변경 불가

메모리 할당

- 동적(dynamic) 할당
  - 실행 도중(런타임)에 동적으로 메모리를 할당받는 것
  - 사용이 끝나면 시스템에 메모리 반납

  - ex) int* score = (int*) malloc(sizeof(int) * 100);
  - 필요한 만큼만 할당받아서 사용

메모리 할당



헤더파일 포함

동적 메모리 할당

메모리 할당 해제

동적 할당

- malloc 함수
  - 매개변수로 할당받을 바이트의 수를 넘겨줌

  - free() 함수는 동적 할당된 메모리 블록을 반납

```
int *score;
score = (int *)malloc(sizeof(int)*10);
…
free(score);
```

동적 할당

- free 함수
  - 매개변수로 동적 할당된 메모리를 가리키는 포인터 넘겨줌
  - malloc() 함수는 메모리 블록의 첫 번째 바이트에 대한 주소 반환
  - 만약 요청한 메모리 공간을 할당할 수 없는 경우 NULL 반환

```
int *score;
score = (int *)malloc(sizeof(int)*10);
if( score == NULL ){
        ... // 오류 처리
}
```

동적 할당

- 할당받은 공간의 이용
  - 배열과 같이 취급하여 사용

```
score[0] = 100;
score[1] = 200;
score[3] = 300;
…
```

  - 포인터를 통하여 사용

```
*score = 100;
*(score + 1) = 200;
*(score + 2) = 300;
…
```

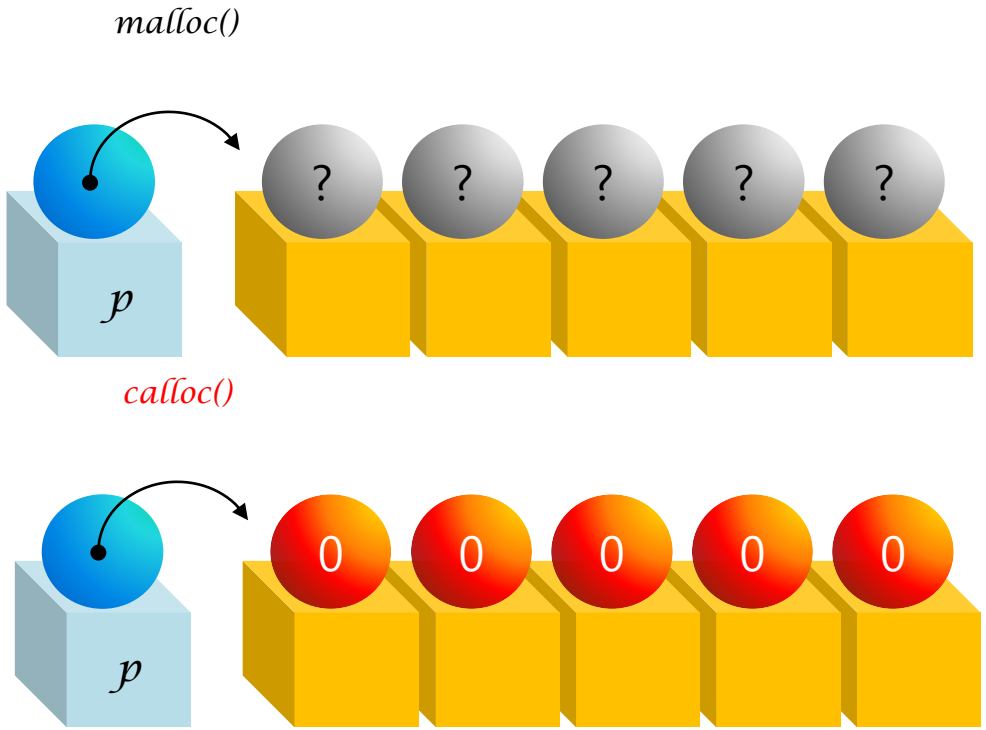
동적 할당

```
4   int main(void)
5   {
6       int* list;
7       int i, students;
8       double mean = 0;
9
10      printf("학생의 수 입력 : ");
11      scanf("%d", &students);
12
13      list = malloc(students * sizeof(int));
14
15      for (i = 0;i < students;i++)
16      {
17          printf("학생 #%d 성적 입력 : ", i + 1);
18          scanf("%d", &list[i]);
19          mean += list[i];
20      }
21
22      printf("성적 평균 : %lf\n", mean / students);
23      return 0;
24  }
```

동적 할당

- calloc 함수
  - 0으로 초기화 된 메모리 할당

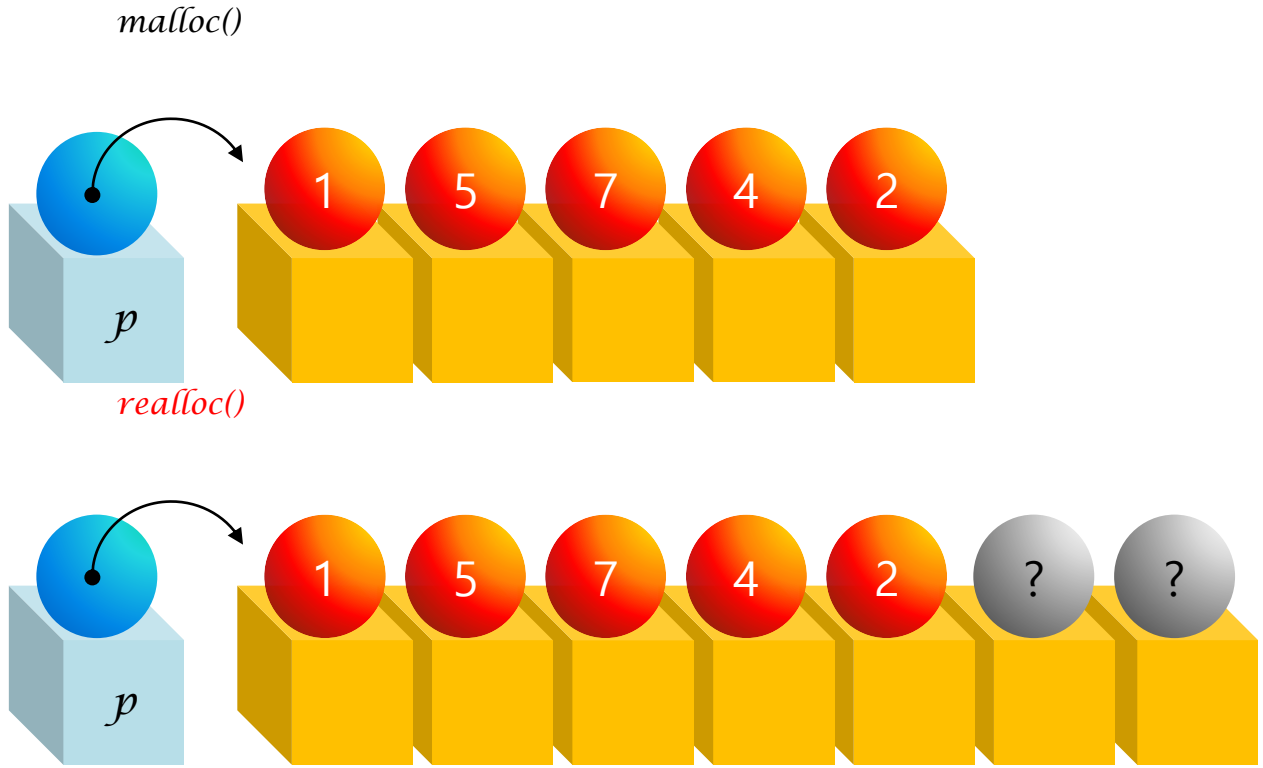
  - ex) score = (int*) calloc(5, sizeof(int));



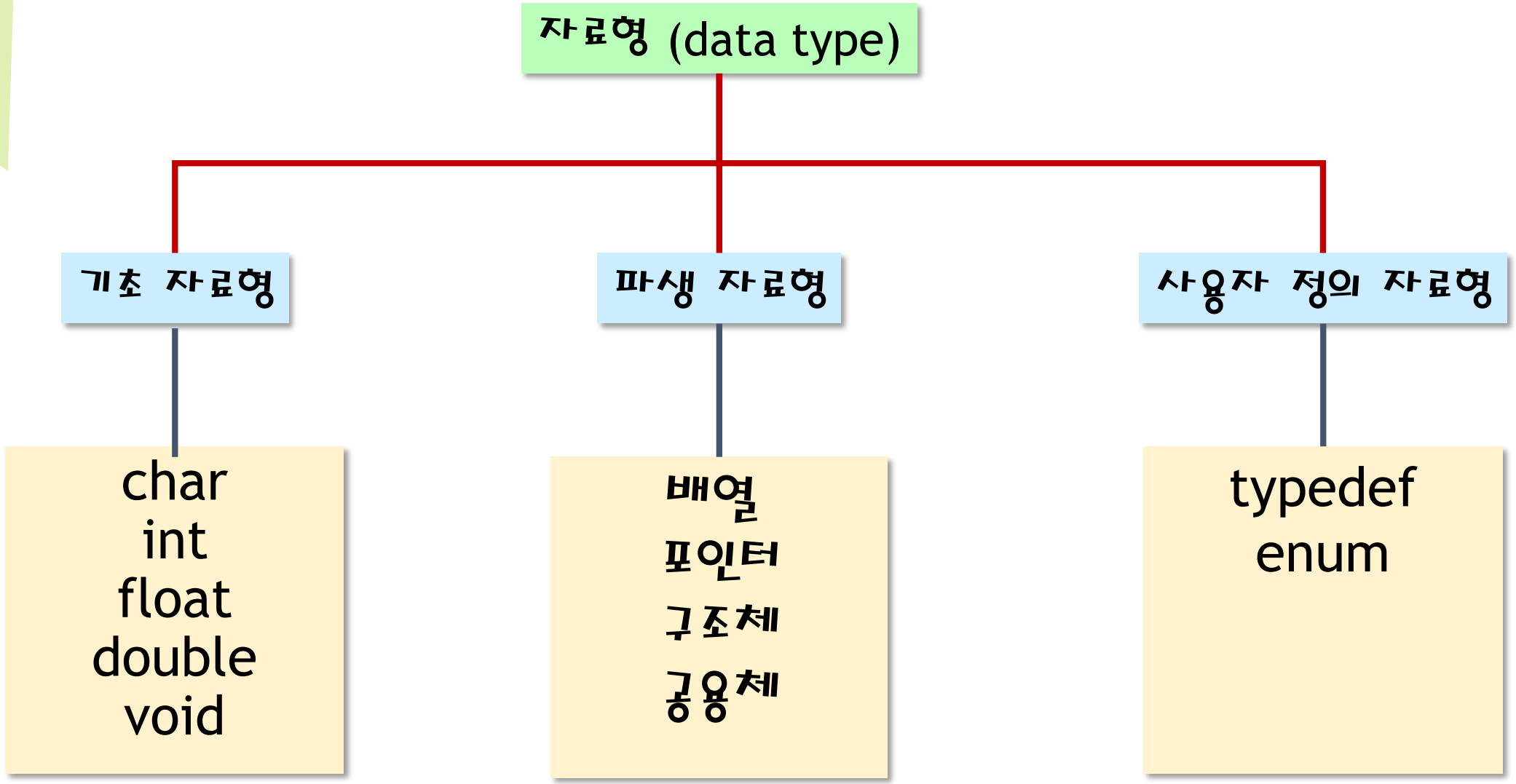
동적 할당

- realloc 함수
  - 이미 할당되어 있는 메모리 블록의 크기 변경

  - ex) score = realloc(score, 7 * sizeof(int));



구조체


자료형 (data type)
기초 자료형
파생 자료형
사용자 정의 자료형
char
int
float
double
void
배열
포인터
구조체
공용체
typedef
enum

구조체

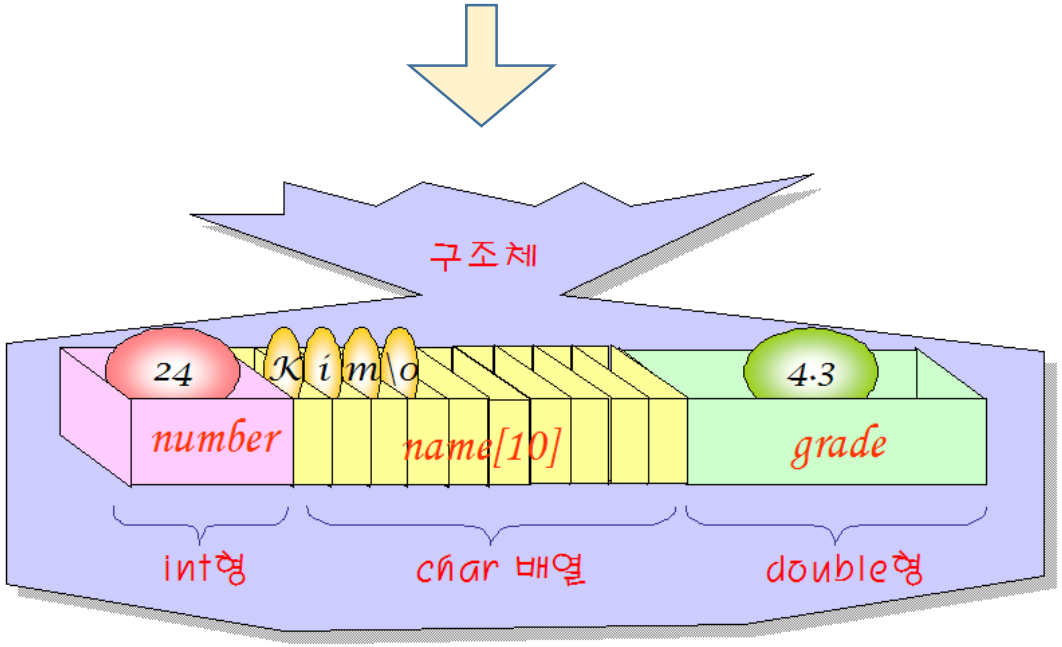
- 학생에 대한 데이터를 표현하려면?
  - 학번, 이름, 학점 등의 여러 개의 데이터 필요

  - int number;
  - char name[10];
  - double grade;
  - ...

  - 이러한 정보를 묶어서 표시할 수 없을까 -> 구조체

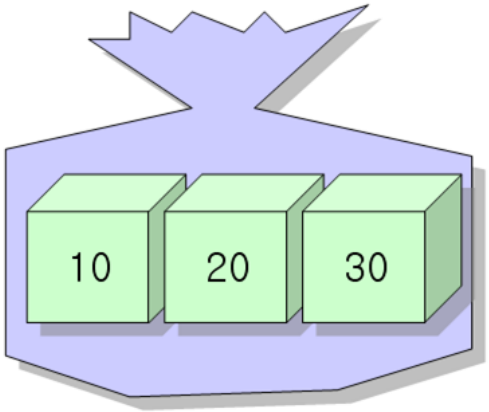
구조체

```
int number;
char name[10];
double grade;
```



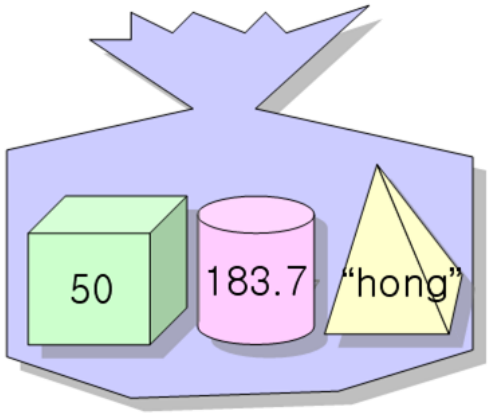
구조체

- 배열 vs 구조체
  - 배열 : 같은 자료형의 집합



배열

  - 구조체 : 다른 자료형의 집합

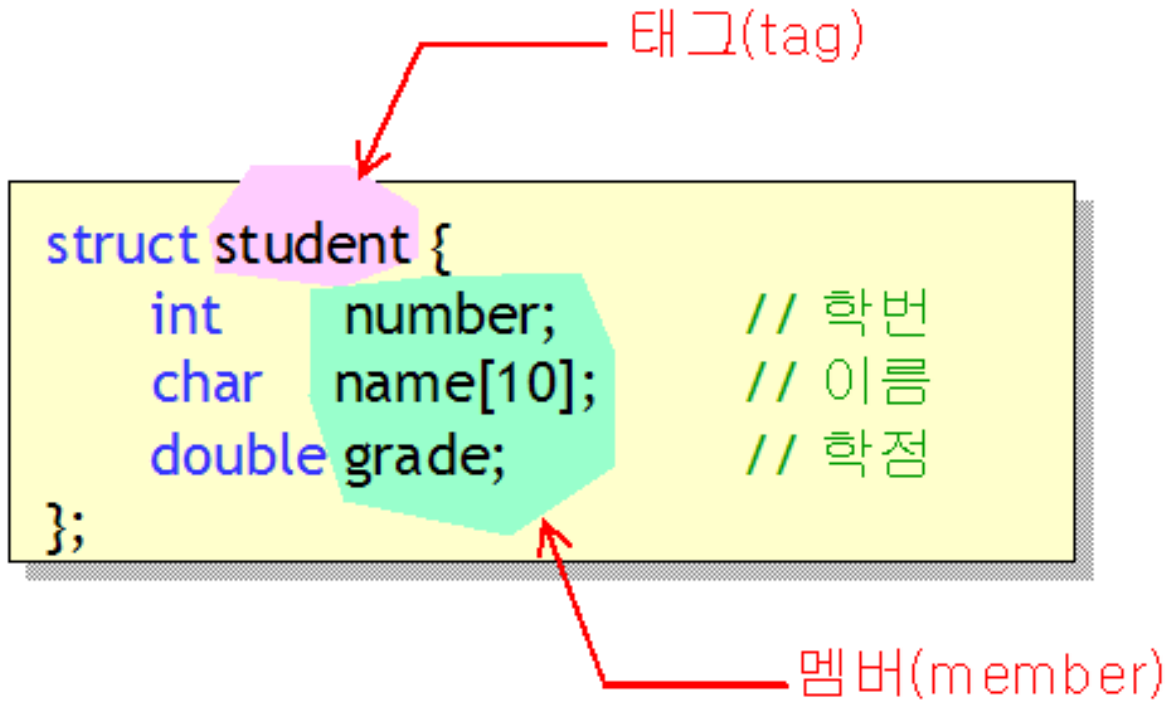


구조체

구조체

- 구조체 선언 형식

```
struct 태그
{
    자료형 멤버1;
    자료형 멤버2;
    ...
};  //세미콜론 주의
```



구조체 변수 선언

- 구조체 선언과 구조체 변수 선언은 다름
  - 구조체 선언은 틀 선언, 구조체 변수 선언은 실제 사용

```
struct student {
        int number;
        char name[10];
        double grade;
};  //구조체 선언은 메인함수 밖에서
…
…
```

```
int main(void)
{
        struct student s1;

        struct student s2 = {24, “Kim”, 4.3};
}
```

구조체

```c
struct student
{
    int number;
    char name[10];
    double grade;
};

int main(void)
{
    struct student s;

    printf("학번을 입력하시오 : ");
    scanf("%d", &s.number);
    printf("이름을 입력하시오 : ");
    scanf("%s", s.name);
    printf("성적을 입력하시오 : ");
    scanf("%lf", &s.grade);

    printf("%d 번 학생 %s의 성적은 %lf입니다.\n", s.number, s.name, s.grade);

    return 0;
```

구조체

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

struct point
{
    int x, y;
};

int main(void)
{
    struct point p1, p2;
    double distance;

    printf("점 1의 좌표를 입력하시오 : ");
    scanf("%d %d", &p1.x, &p1.y);

    printf("점 2의 좌표를 입력하시오 : ");
    scanf("%d %d", &p2.x, &p2.y);

    distance = sqrt((p1.x - p2.x)*(p1.x - p2.x) + (p1.y - p2.y)*(p1.y - p2.y));

    printf("거리는 %lf입니다.\n", distance);

    return 0;
}
```

구조체를 멤버로 갖는 구조체

```c
struct point
{
    int x, y;
};

struct circle
{
    struct point po;
    double radius;
};

int main(void)
{
    struct circle cir;

    cir.po.x = 1;
    cir.po.y = 2;
    cir.radius = 5.0;

    printf("중심의 위치는 (%d, %d), 넓이는 %lf입니다.\n", cir.po.x, cir.po.y, cir.radius * cir.radius * 3.14);

    return 0;
}
```

구조체 배열

```
4   struct student
5   {
6       int number;
7       char name[10];
8       double grade;
9   };
10
11  int main(void)
12  {
13      struct student list[3] = {
14          {1, "Kim", 4.3},
15          {2, "Lee", 4},
16          {3, "Park", 4.2}
17      };
18
19      list[1].number = 5;
20  }
```

구조체 배열

```
4  struct student
5  {
6      int number;
7      char name[10];
8      double grade;
9  };
10
11 int main(void)
12 {
13     struct student list[3];
14     int i;
15
16     for (i = 0;i < 3;i++)
17     {
18         printf("%d번째 학생 정보 입력 : ");
19         scanf("%d %s %lf", &list[i].number, list[i].name, &list[i].grade);
20     }
21
22     for (i = 0;i < 3;i++)
23         printf("%d %s %lf\n", list[i].number, list[i].name, list[i].grade);
24
25     return 0;
26 }
27
```

구조체와 함수

```c
#include<stdio.h>

struct Point
{
    int x, y;
};
void PrintPoint(struct Point p);
void PrintAll(struct Point p[], int length);

int main(void)
{
    struct Point p1 = { 1,2 };
    struct Point list[2] = { {3, 4}, {5, 6} };

    PrintPoint(p1);
    PrintAll(list, 2);

    return 0;
}
```

```c
void PrintPoint(struct Point p)
{
    printf("(%d, %d)\n", p.x, p.y);
}

void PrintAll(struct Point p[], int length)
{
    int i;
    for (i = 0;i < length;i++)
        PrintPoint(p[i]);
}
```

구조체 포인터

```
4   struct student
5   {
6       int number;
7       char name[10];
8       double grade;
9   };
10
11  int main(void)
12  {
13      struct student s = { 1, "kim", 4.3 };
14      struct student* p;
15
16      p = &s;
17
18      printf("%d %s %lf\n", (*p).number, (*p).name, (*p).grade);
19
20      return 0;
21  }
```

구조체 포인터

```c
struct student
{
    int number;
    char name[10];
    double grade;
};

int main(void)
{
    struct student s = { 1, "kim", 4.3 };
    struct student* p;

    p = &s;

    printf("%d %s %lf\n", p->number, p->name, p->grade);

    return 0;
}
```

(*p).number p->number

둘은 같은 표현

구조체 동적할당

```c
struct student
{
    int number;
    char name[10];
    double grade;
};

int main(void)
{
    struct student* list = malloc(sizeof(struct student) * 3);
    int i;

    for (i = 0;i < 3;i++)
    {
        printf("정보 입력 : ");
        scanf("%d %s %lf", &list[i].number, list[i].name, &list[i].grade);
    }

    for (i = 0;i < 3;i++)
        printf("%d %s %lf\n", list[i].number, list[i].name, list[i].grade);
}
```

typedef

```c
typedef struct student
{
    int number;
    char name[10];
    double grade;
}Student;

void Print(Student s);

int main(void)
{
    Student s = { 1, "Kim", 4.3 };
    Print(s);
}

void Print(Student s)
{
    printf("%d %s %lf\n", s.number, s.name, s.grade);
}
```

실습문제

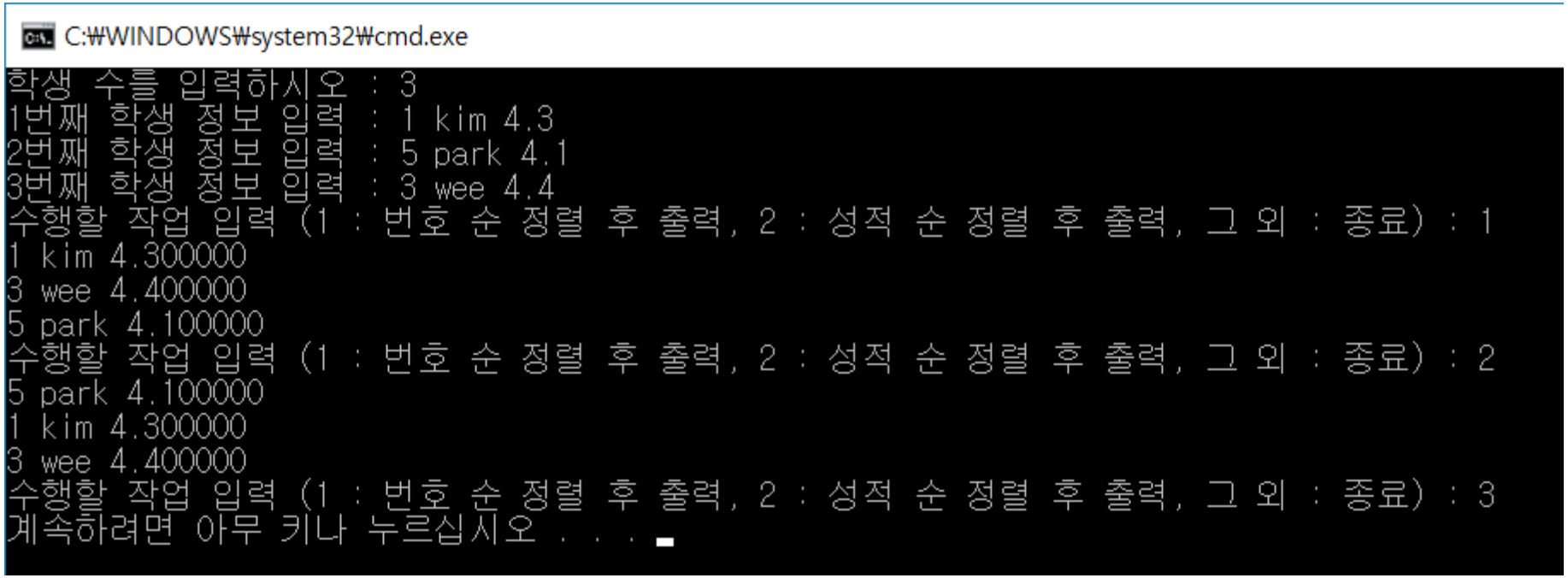
- 구조체 student를 이용한다. 프로그램 시작 시, 학생 수를 입력받아 동적할당을 받는다. 그 후 모든 학생의 정보를 입력받는다.

  번호 순으로 정렬하는 함수 SortByNumber(struct student list[], int length)와

  성적 순 정렬함수 SortByGrade(struct student list[], int length)를 만든다.

  그 후 아래 출력과 같이 동작하도록 하라.

- 힌트 : 두 함수는 차이점이 거의 없다.



실습문제

```
5   typedef struct student
6   {
7       int number;
8       char name[10];
9       double grade;
10  }Student;
11
12  void SortByNumber(Student list[], int length);
13  void SortByGrade(Student list[], int length);
14  void Print(Student list[], int length);
```

```
16  int main(void)
17  {
18      int length;
19      int menu;
20      int i;
21      Student* list;
22
23      printf("학생 수를 입력하시오 : ");
24      scanf("%d", &length);
25
26      list = malloc(sizeof(Student) * length);
27
28      for (i = 0;i < length;i++)
29      {
30          printf("%d번째 학생 정보 입력 : ", i + 1);
31          scanf("%d %s %lf", &list[i].number, list[i].name, &list[i].grade);
32      }
```

실습문제

```
34      while (1)
35      {
36          printf("수행할 작업 입력 (1 : 번호 순 정렬 후 출력, 2 : 성적 순 정렬 후 출력, 그 외 : 종료) : ");
37          scanf("%d", &menu);
38
39          switch (menu)
40          {
41          case 1:
42              SortByNumber(list, length);
43              Print(list, length);
44              break;
45          case 2:
46              SortByGrade(list, length);
47              Print(list, length);
48              break;
49          default:
50              free(list);
51              return 0;
52          }
53      }
54  }
```

```
100 void Print(Student list[], int length)
101 {
102     int i;
103     for (i = 0;i < length;i++)
104     {
105         printf("%d %s %lf\n", list[i].number, list[i].name, list[i].grade);
106     }
107 }
```

실습문제

```
56  void SortByNumber(Student list[], int length)
57  {
58      int min_idx;
59      int i, j;
60      Student tmp;
61
62      for (i = 0;i < length - 1;i++)
63      {
64          min_idx = i;
65
66          for (j = i + 1;j < length;j++)
67          {
68              if (list[min_idx].number > list[j].number)
69                  min_idx = j;
70          }
71
72          tmp = list[i];
73          list[i] = list[min_idx];
74          list[min_idx] = tmp;
75      }
76  }
```

```
78  void SortByGrade(Student list[], int length)
79  {
80      int min_idx;
81      int i, j;
82      Student tmp;
83
84      for (i = 0;i < length - 1;i++)
85      {
86          min_idx = i;
87
88          for (j = i + 1;j < length;j++)
89          {
90              if (list[min_idx].grade > list[j].grade)
91                  min_idx = j;
92          }
93
94          tmp = list[i];
95          list[i] = list[min_idx];
96          list[min_idx] = tmp;
97      }
98  }
```

감사합니다