C Language

Lecture 7. 구조체

BAE SEKANG(배세강)

Contents

Structure

♣ 구조체

- 서로 다른 자료형의 데이터를 하나의 의미 있는 단위로 묶는 사용자 정의 자료형

```
#include <stdio.h>

struct Student {
    int id;
    char name[20];
    float grade;
};

int main(void) {
    struct Student s1;

    return 0;
}
```

Structure

♣ 구조체 선언

- struct [자료형 이름] {구조체 내부};

```
#include <stdio.h>

struct Student { // type: struct Student이다. 그냥 Student가 아님에 주의!
    int id;
    char name[20];
    float grade;
};

int main(void) {
    struct Student s1;

    return 0;
}
```

Structure

♣ 멤버(member) 변수

- 구조체 내부의 변수: id, name, grade

```
#include <stdio.h>

struct Student {
    int id;
    char name[20];
    float grade;
};

int main(void) {
    struct Student s1;

    return 0;
}
```

Structure

♣ 멤버(member) 접근

- 연산자 .을 사용함

```
#include <stdio.h>

struct Point {
    int x;
    int y;
};

int main(void) {
    struct Point p1 = { .x = 0, .y = 0 };
    p1.x = 10;
    p1.y = -20;
    return 0;
}
```

Structure

[Practice]

두 점 A, B의 정수 좌표 A(x1, y1), B(x2, y2)를 입력 받는다. 입력 형태는 x1 y1 x2 y2이다. 이 때, A와 B를 지나는 직선의 기울기를 출력한다. (단, x1 < x2, y1 < y2인 입력만 존재한다고 가정한다.)

[입력 1]
3 5 6 15

[입력 2]
2 10 4 40

[출력 1]
3.3

[출력 2]
15.0

Structure

[Practice] - Answer

```
#include <stdio.h>

struct Point {
    int x;
    int y;
};

int main(void) {
    struct Point p1, p2;

    scanf("%d %d %d %d", &p1.x, &p1.y, &p2.x, &p2.y);
    printf("%.1f", (float)(p2.y - p1.y) / (p2.x - p1.x));

    return 0;
}
```

Structure

♣ typedef

- 기존 type에 별명을 붙이는 키워드

```
#include <stdio.h>

typedef int Integer;

int main(void) {
    Integer a = 10;
    printf("%d₩n", a);

    return 0;
}
```

Structure

♣ typedef

- struct에 typedef를 사용해 간단히 정의할 수 있다.

```
#include <stdio.h>

typedef struct Student {
    int id;
    char name[20];
} Student;

int main(void) {
    Student s1;
    s1.id = 1;
    printf("%d₩n", s1.id);
    return 0;
}
```

Structure

♣ structure assignment

- 한 구조체의 모든 멤버 값을 다른 구조체로 복사한다. (같은 구조체 type끼리 가능)

```
// …
typedef struct {
    int id;
    char name[20];
    float grade;
} Student;

int main(void) {
    Student s1 = {2024001, "Kim", 4.3f};
    Student s2;
    s2 = s1;
    s2.id = 9999;
    strcpy(s2.name, "Lee");
    s2.grade = 3.0f;
    printf("s1 -> %d, %s, %.1f₩n", s1.id, s1.name, s1.grade);
    printf("s2 -> %d, %s, %.1f₩n", s2.id, s2.name, s2.grade);
    return 0;
}
```

Enum

♣ enum

- 이름 있는 정수 상수들의 집합

```
#include <stdio.h>

enum Day {
   MON, TUE, WED, THU, FRI, SAT, SUN
};

int main(void) {
   enum Day today = WED;

   printf("today = %d₩n", today);

   return 0;
}
```

Enum

♣ internal values

- enum 변수의 값을 직접 줄 수도 있다.

```
#include <stdio.h>

enum Status {
    READY = 1,
    RUNNING = 5,
    FINISHED = 10
};

int main(void) {
    enum Status s = RUNNING;

    printf("%d₩n", s);
    return 0;
}
```

Enum

♣ internal values

- 중간 값을 지정하면 그 뒤는 이어서 증가한다.

```
#include <stdio.h>

enum Example {
    A,      // 0
    B = 5,  // 5
    C,      // 6
    D       // 7
};

int main(void) {
    printf("%d %d %d %d", A, B, C, D);

    return 0;
}
```

Enum

♣ Example: enum

- switch문과 함께 주로 사용한다.

```
// …
enum Menu { START, LOAD, EXIT };
int main(void) {
    enum Menu choice = LOAD;
    switch (choice) {
        case START:
            printf("게임 시작₩n");
            break;
        case LOAD:
            printf("불러오기₩n");
            break;
        case EXIT:
            printf("종료₩n");
            break;
    }
    return 0;
}
```

Enum

♣ typedef

- 더욱 간편하게 활용할 수 있다.

```
#include <stdio.h>

typedef enum Status {
    READY,
    RUNNING,
    STOPPED
} Status;

int main(void) {
    Status s1 = READY;
    enum Status s2 = STOPPED;

    return 0;
}
```

Enum

[Quiz] : 다음 프로그램의 출력 결과는?

```
#include <stdio.h>

typedef enum {
   READ  = 1 << 0,
   WRITE = 1 << 1,
   EXEC  = 1 << 2
} Permission;

typedef enum {
   STOPPED,
   WORKING,
   LOCKED = 5
} Status;

int main() {
   Permission user_p = READ | WRITE;
   Status sys_s = WORKING;
   if ((user_p & WRITE) && (sys_s < LOCKED)) sys_s += 2;
   int final_val = (sys_s + EXEC > LOCKED) ? (sys_s * 10) : (user_p + sys_s);

   printf("%d %d", sys_s, final_val);
   return 0;
}
```

Enum

[Quiz] – Answer: 3 30

```
#include <stdio.h>

typedef enum {
   READ  = 1 << 0,
   WRITE = 1 << 1,
   EXEC  = 1 << 2
} Permission;

typedef enum {
   STOPPED,
   WORKING,
   LOCKED = 5
} Status;

int main() {
   Permission user_p = READ | WRITE; // 3
   Status sys_s = WORKING; // 1
   if ((user_p & WRITE) && (sys_s < LOCKED)) sys_s += 2; // (3 & 2) && (1 < 5) ➔ true ➔ sys_s <- 3
   int final_val = (sys_s + EXEC > LOCKED) ? (sys_s * 10) : (user_p + sys_s); // 30

   printf("%d %d", sys_s, final_val);// 3 30
   return 0;
}
```

Union

♣ 공용체

- 서로 다른 자료형의 멤버들이 하나의 메모리 공간을 공유하는 자료형
- 가장 큰 멤버 크기를 기준으로 크기가 결정된다.

```
#include <stdio.h>

union Example {
    int i;
    float f;
    char c;
};

int main(void) {
    union Example x;
    x.i = 10, x.f = 3.14f, x.c = 'A';
    printf("%d %.2f %c₩n", x.i, x.f, x.c); // 1078523201 3.14 A
    return 0;
}
```

Union

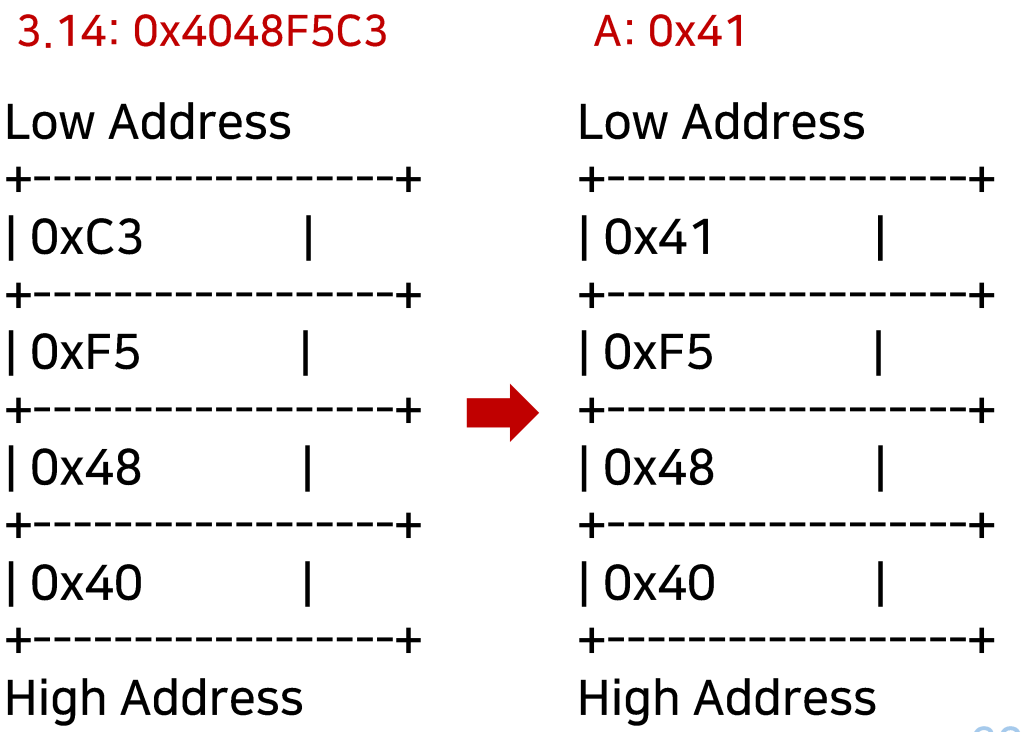
♣ 공용체 메모리 구조

- 모든 멤버가 같은 메모리 공간을 공유하기 때문에 다른 멤버 값을 덮어쓸 수 있다.

```
#include <stdio.h>

union Example {
    int i;
    float f;
    char c;
};

int main(void) {
    union Example x;
    x.i = 10, x.f = 3.14f, x.c = 'A';
    printf("%d %.2f %c₩n", x.i, x.f, x.c); // 1078523201 3.14 A
    return 0;
}
```



Union

♣ Tagged Union

- 공용체만으로는 메모리 내부에 저장된 값이 어떤 자료형인지 알 수 없다.
- Tag를 두어 안전하게 공용체를 사용할 수 있다.

```
#include <stdio.h>

typedef enum {
    TYPE_INT,
    TYPE_FLOAT,
    TYPE_STR
} ValueType;

typedef struct {
    ValueType type;
    union {
        int i;
        float f;
        char s[20];
    } data;
} Variant;
```

Union

♣ Example: Tagged Union

- 공용체만으로는 메모리 내부에 저장된 값이 어떤 자료형인지 알 수 없다.
- Tag를 두어 안전하게 공용체를 사용할 수 있다.

```
// …
Variant v;
switch (v.type) {
      case TYPE_INT:
         printf("정수 출력: %d₩n", v.data.i);
         break;
      case TYPE_FLOAT:
         printf("실수 출력: %.2f₩n", v.data.f);
         break;
      case TYPE_STR:
         printf("문자열 출력: %s₩n", v.data.s);
         break;
   }

// …
```

Exercise

[Problem 1]

정수 N을 입력 받는다. 이후 N명의 학생들의 국어, 영어, 수학 점수를 입력 받을 때, 각 과목 별 평균 점수를 예시와 같이 출력한다. (단, 모든 점수의 출력은 소수점 첫 째 자리까지 출력한다. N의 범위는 $1 \leq N \leq 100$으로 가정한다.)

[입력 1]
3
90 85 100
75 90 80
60 100 95

[출력 1]
국어: 75.0점
영어: 91.7점
수학: 91.7점

Exercise

[Problem 1] - Answer

```
#include <stdio.h>

typedef struct {
    int kor;
    int eng;
    int math;
} Score;

int main() {
    int N;
    scanf("%d", &N);
    Score students[100];
    double kor_sum = 0, eng_sum = 0, math_sum = 0;

    for (int i = 0; i < N; i++) {
        scanf("%d %d %d", &students[i].kor, &students[i].eng, &students[i].math);
        kor_sum += students[i].kor;
        eng_sum += students[i].eng;
        math_sum += students[i].math;
    }
    printf("국어: %.1f점₩n", kor_sum / N);
    printf("영어: %.1f점₩n", eng_sum / N);
    printf("수학: %.1f점₩n", math_sum / N);

    return 0;
}
```

THANK YOU!

Any Question?