

실습 1.

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

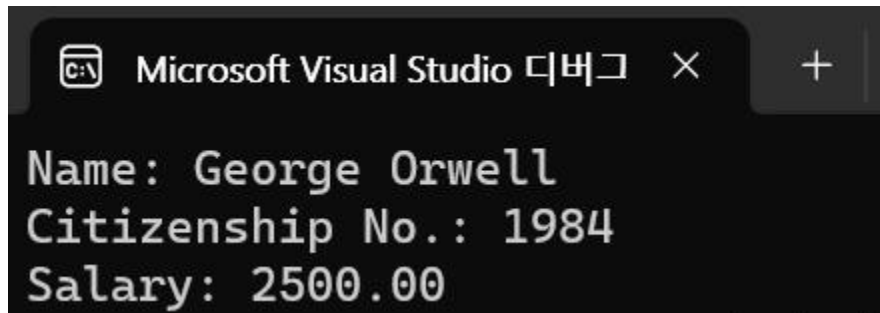
// create struct with person1 variable

struct Pperson {
    char name[50];
    int citNo;
    float salary;
} person1;

int main() {
    // assign value to name of person1
    strcpy_s(person1.name, 50, "George Orwell");
    // assign values to other person1 variables
    person1.citNo = 1984;
    person1.salary = 2500;
    // print struct variables
    printf("Name: %s\n", person1.name);
    printf("Citizenship No.: %d\n", person1.citNo);
    printf("Salary: %.2f", person1.salary);

    return 0;
}
```

실습 1 실행 화면.

A screenshot of a Microsoft Visual Studio debug console window. The window title bar shows the Visual Studio icon, the text "Microsoft Visual Studio 디버그", and standard window controls (close, maximize, and a plus sign for additional windows). The console output displays the results of the program's execution: "Name: George Orwell", "Citizenship No.: 1984", and "Salary: 2500.00".

```
Microsoft Visual Studio 디버그 × +
Name: George Orwell
Citizenship No.: 1984
Salary: 2500.00
```

2019313550_박병현

실습 2.

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

```
#include <string.h>
```

```
// create struct with person1 variable
```

```
typedef struct Pperson {
```

```
    char name[50];
```

```
    int citNo;
```

```
    float salary;
```

```
} person;
```

```
int main() {
```

```
    // create Person variable
```

```
    person p1;
```

```
    // assign value to name of p1
```

```
    strcpy_s(p1.name, 50, "George Orwell");
```

```
    // assign values to other person1 variables
```

```
    p1.citNo = 1984;
```

```
    p1.salary = 2500;
```

```
    // print struct variables
```

```
    printf("Name: %s\n", p1.name);
```

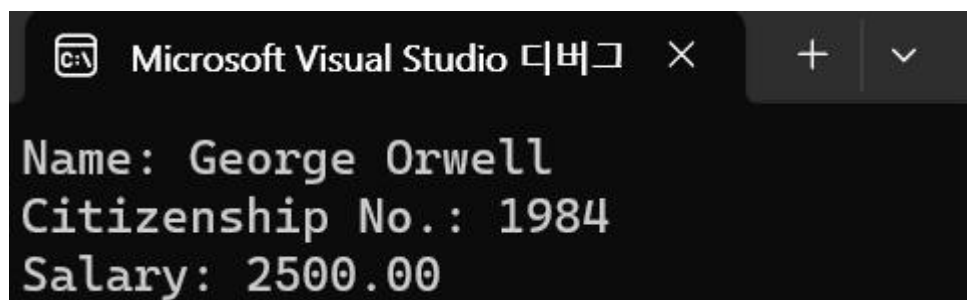
```
    printf("Citizenship No.: %d\n", p1.citNo);
```

```
    printf("Salary: %.2f", p1.salary);
```

```
    return 0;
```

```
}
```

실습 2 실행 화면.

The image shows a screenshot of the Microsoft Visual Studio debug console. The title bar at the top reads "Microsoft Visual Studio 디버그" with standard window controls (minimize, maximize, close). The console output is as follows:

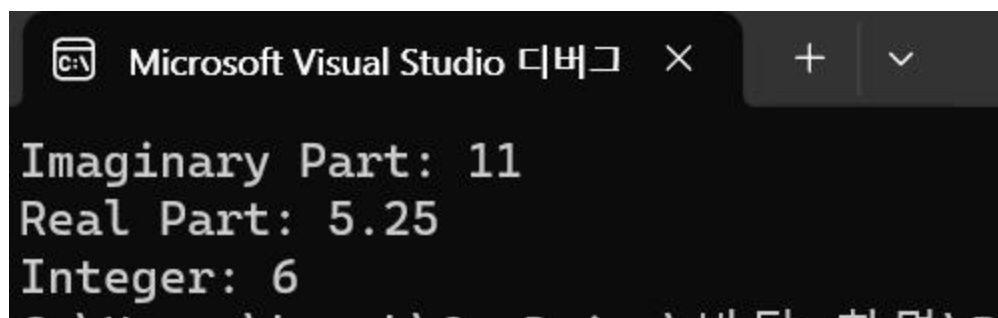
```
Name: George Orwell
Citizenship No.: 1984
Salary: 2500.00
```

실습 3.

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

```
struct complex {  
    int imag;  
    float real;  
};  
  
struct number {  
    struct complex comp;  
    int integer;  
};  
  
int main() {  
  
    struct number num1;  
  
    // initialize complex variables  
    num1.comp.imag = 11;  
    num1.comp.real = 5.25;  
  
    // initialize number variable  
    num1.integer = 6;  
  
    // print struct variables  
    printf("Imaginary Part: %d\n", num1.comp.imag);  
    printf("Real Part: %.2f\n", num1.comp.real);  
    printf("Integer: %d", num1.integer);  
}
```

실습 3 실행 화면.



The screenshot shows the Microsoft Visual Studio debug console window. The title bar reads "Microsoft Visual Studio 디버그" with a close button. The console output displays the results of the program's execution: "Imaginary Part: 11", "Real Part: 5.25", and "Integer: 6".

```
Microsoft Visual Studio 디버그 × + ∨  
  
Imaginary Part: 11  
Real Part: 5.25  
Integer: 6
```

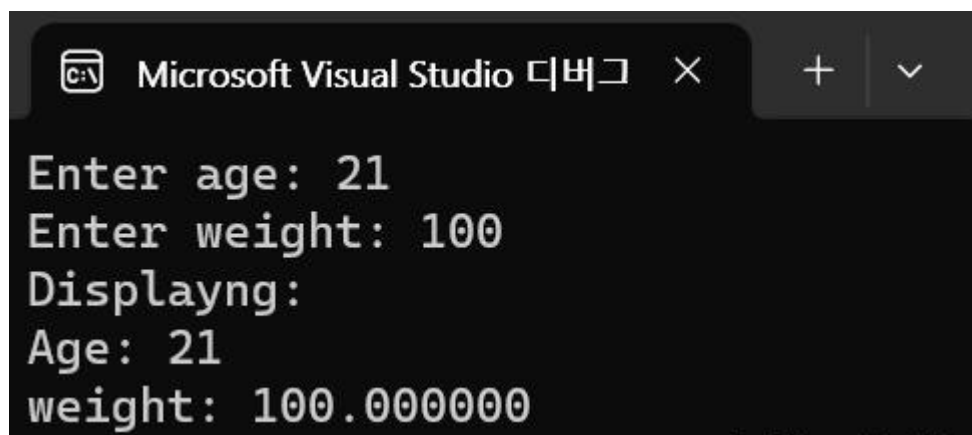
실습 4.

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

```
struct person {  
    int age;  
    float weight;  
};
```

```
int main() {  
    struct person* personPtr, person1;  
    personPtr = &person1;  
  
    printf("Enter age: ");  
    scanf("%d", &personPtr->age);  
  
    printf("Enter weight: ");  
    scanf("%f", &personPtr->weight);  
  
    printf("Displayng:\n");  
    printf("Age: %d\n", personPtr->age);  
    printf("weight: %f", personPtr->weight);  
  
    return 0;  
}
```

실습 4 실행 화면.



The screenshot shows the Microsoft Visual Studio Debug Console window. The title bar reads 'Microsoft Visual Studio 디버그'. The console output is as follows:

```
Enter age: 21  
Enter weight: 100  
Displayng:  
Age: 21  
weight: 100.000000
```

실습 5.

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

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

```
typedef struct {
```

```
    int x;
```

```
    int y;
```

```
} point;
```

```
int main() {
```

```
    point* mypoint = NULL;
```

```
    /* Dynamically allocate a new point struct which mypoint points to here*/
```

```
    mypoint = (point*)malloc(sizeof(point));
```

```
    mypoint->x = 10;
```

```
    mypoint->y = 5;
```

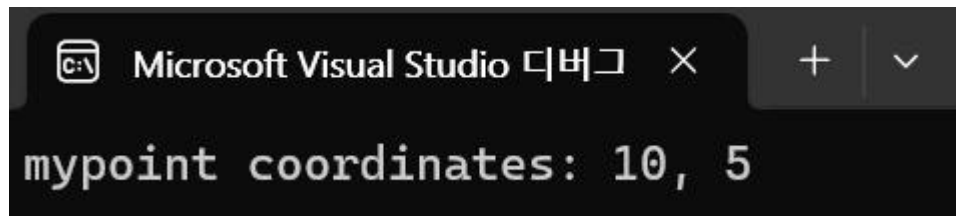
```
    printf("mypoint coordinates: %d, %d\n", mypoint->x, mypoint->y);
```

```
    free(mypoint);
```

```
    return 0;
```

```
}
```

실습 5 실행 화면.



실습 6.

```
#include <stdio.h>
struct student {
    char name[50];
    int age;
};

// function prototype
void display(struct student s);

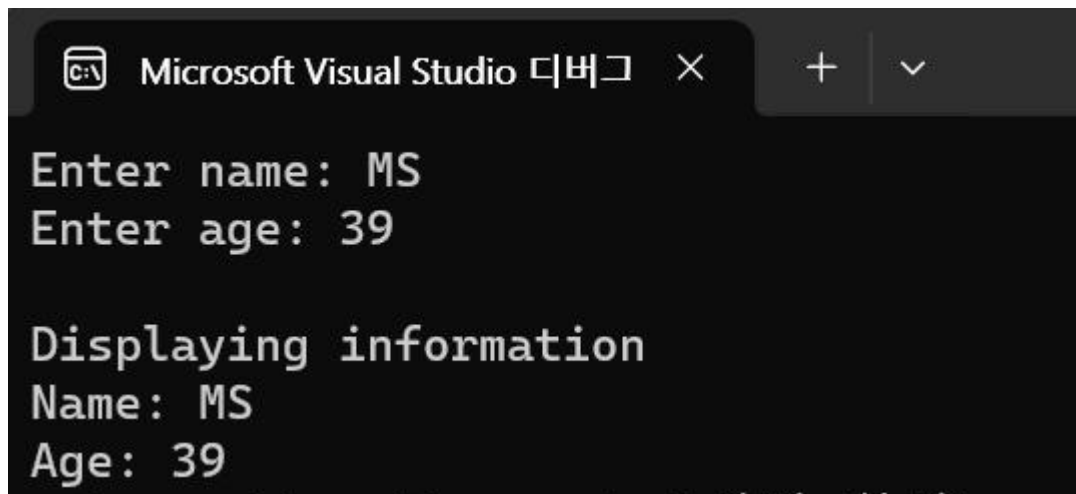
int main() {
    struct student s1;

    printf("Enter name: ");
    scanf("%s", s1.name, sizeof(s1.name));
    printf("Enter age: ");
    scanf("%d", &s1.age);
    display(s1); // passing struct as an argument

    return 0;
}

void display(struct student s) {
    printf("\nDisplaying information\n");
    printf("Name: %s", s.name);
    printf("\nAge: %d", s.age);
}
```

실습 6 실행 화면.



```
Microsoft Visual Studio 디버그
Enter name: MS
Enter age: 39

Displaying information
Name: MS
Age: 39
```

실습 7.

```
#include <stdio.h>
struct student {
    char name[50];
    int age;
};

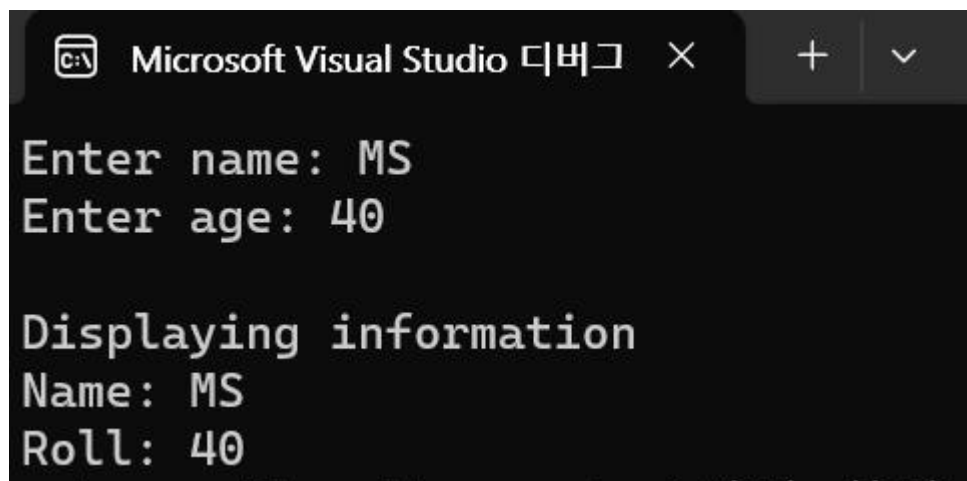
// function prototype
struct student getinformation();

int main() {
    struct student s;
    s = getinformation();
    printf("\nDisplaying information\n");
    printf("Name: %s", s.name);
    printf("\nRoll: %d", s.age);
    return 0;
}

struct student getinformation() {
    struct student s1;

    printf("Enter name: ");
    scanf("%s", s1.name, sizeof(s1.name));
    printf("Enter age: ");
    scanf("%d", &s1.age);
    return s1;
}
```

실습 7 실행 화면.

A screenshot of the Microsoft Visual Studio debugger's output window. The window has a dark background and a title bar that reads "Microsoft Visual Studio 디버깅" with standard window controls. The output text is as follows:
Enter name: MS
Enter age: 40

Displaying information
Name: MS
Roll: 40

실습 8.

```
#include <stdio.h>
typedef struct Complex{
    float real;
    float imag;
} complex;

void addNumbers(complex c1, complex c2, complex* result)
{
    result->real = c1.real + c2.real;
    result->imag = c1.imag + c2.imag;
}

int main(){
    complex c1, c2, result;


    printf("For first number,\n");
    printf("Enter real part: ");
    scanf("%f", &c1.real);
    printf("Enter imaginary part: ");
    scanf("%f", &c1.imag);

    printf("For second number, \n");
    printf("Enter real part: ");
    scanf("%f", &c2.real);
    printf("Enter imaginary part: ");
    scanf("%f", &c2.imag);

    addNumbers(c1, c2, &result);
    printf("\nresult.real = %.1f\n", result.real);
    printf("result.imag = %.1f", result.imag);

    return 0;
}
```

실습 8 실행 화면.



```
Microsoft Visual Studio 디버그
For first number,
Enter real part: 3
Enter imaginary part: 5
For second number,
Enter real part: 2
Enter imaginary part: 4

result.real = 5.0
result.imag = 9.0
```