

# 함수 - 추가자료

**박종혁 교수**

**UCS Lab**

Tel: 970-6702

Email: [jhpark1@seoultech.ac.kr](mailto:jhpark1@seoultech.ac.kr)

# 1. Making coffee 프로그램

```
#include <stdio.h>

int main()
{
    int coffee;          ---커피 종류 선택 변수

    printf("어떤 커피 드릴까요? (1:보통, 2:설탕, 3:블랙) ");
    scanf_s("%d", &coffee); ---커피 종류 입력

    printf("\n# 1. 뜨거운 물을 준비한다\n");
    printf("# 2. 종이컵을 준비한다\n");

    switch (coffee)
    {
        case 1 : printf("# 3. 보통커피를 탄다\n"); break;
        case 2 : printf("# 3. 설탕커피를 탄다\n"); break;
        case 3 : printf("# 3. 블랙커피를 탄다\n"); break;
        default : printf("# 3. 아무거나 탄다\n"); break;
    }          --- 안내문 출력

    printf("# 4. 물을 붓는다\n");
    printf("# 5. 스푼으로 저어서 녹인다\n\n");

    printf("손님~ 커피 여기 있습니다.\n\n");
}
```

## 2. Making coffee 프로그램 (모듈화)

```
#include <stdio.h>

int coffee_machine(int button)    --- 커피 자판기 함수 구현
{
    printf("\n# 1.(자동으로) 뜨거운 물을 준비한다\n");
    printf("# 2. (자동으로) 종이컵을 준비한다\n");

    switch (button)              --- 버튼에 따른 안내문 출력
    {
        case 1 : printf("# 3. (자동으로) 보통커피를 탄다\n"); break;
        case 2 : printf("# 3. (자동으로) 설탕커피를 탄다\n"); break;
        case 3 : printf("# 3. (자동으로) 블랙커피를 탄다\n"); break;
        default : printf("# 3. (자동으로) 아무거나 탄다\n"); break;
    }

    printf("# 4. (자동으로) 물을 붓는다\n");
    printf("# 5. (자동으로) 스푼으로 저어서 녹인다\n\n");

    return 0;                    --- 메인함수로 돌아간다
}
```

```
int main()
{
    int coffee;                  --- 변수 선언
    int ret;

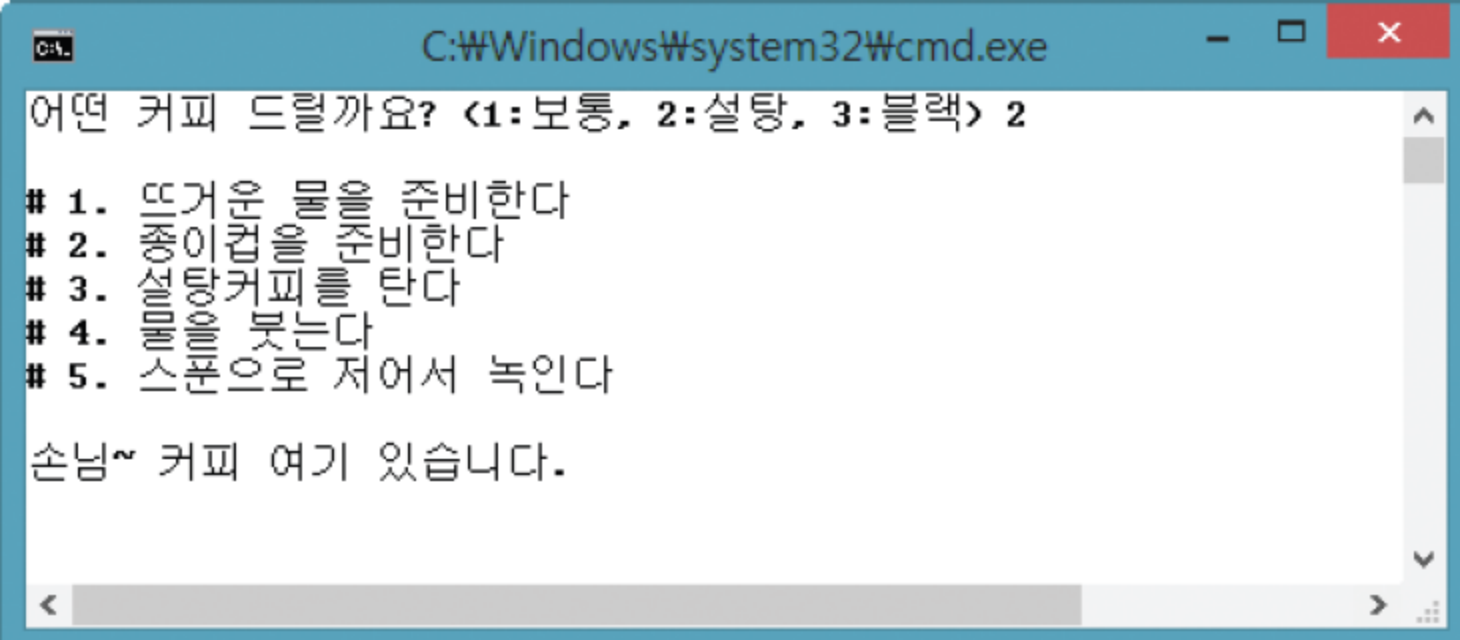
    printf("어떤 커피를 드릴까요? (1:보통, 2:설탕, 3:블랙) ");
    scanf_s("%d", &coffee);    --- 커피 주문

    ret = coffee_machine(coffee); --- 함수 호출

    printf("손님~ 커피 여기 있습니다.\n\n");
}
```

## 2. Making coffee 실행결과

실행결과 ▼



```
C:\Windows\system32\cmd.exe
어떤 커피 드릴까요? <1:보통, 2:설탕, 3:블랙> 2
# 1. 뜨거운 물을 준비한다
# 2. 종이컵을 준비한다
# 3. 설탕커피를 탄다
# 4. 물을 붓는다
# 5. 스푼으로 저어서 녹인다

손님~ 커피 여기 있습니다.
```

### 3. Making coffee 프로그램 - 확장 (1)

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

```
int coffee_machine(int button)
```

--- 커피 자판기 함수 구현

```
{
```

```
    printf("\n# 1.(자동으로) 뜨거운 물을 준비한다\n");
```

```
    printf("# 2. (자동으로) 종이컵을 준비한다\n");
```

```
    switch (button)
```

--- 버튼에 따른 안내문 출력

```
    {
```

```
        case 1 : printf("# 3. (자동으로) 보통커피를 탄다\n"); break;
```

```
        case 2 : printf("# 3. (자동으로) 설탕커피를 탄다\n"); break;
```

```
        case 3 : printf("# 3. (자동으로) 블랙커피를 탄다\n"); break;
```

```
        default : printf("# 3. (자동으로) 아무거나 탄다\n"); break;
```

```
    }
```

```
    printf("# 4. (자동으로) 물을 붓는다\n");
```

```
    printf("# 5. (자동으로) 스푼으로 저어서 녹인다\n\n");
```

```
    return 0;
```

--- 호출한 곳으로 복귀

```
}
```

## 3. Making coffee 프로그램 - 확장 (2)

```
int main()
{
    int coffee;
    int ret;

    printf("A님, 어떤 커피 드릴까요? (1:보통, 2:설탕, 3:블랙) ");
    scanf_s("%d", &coffee);
    ret = coffee_machine(coffee);
    printf("A님 커피 여기 있습니다.\n\n");

    printf("B님, 어떤 커피 드릴까요? (1:보통, 2:설탕, 3:블랙) ");
    scanf_s("%d", &coffee);
    ret = coffee_machine(coffee);
    printf("B님 커피 여기 있습니다.\n\n");

    printf("C님, 어떤 커피 드릴까요? (1:보통, 2:설탕, 3:블랙) ");
    scanf_s("%d", &coffee);
    ret = coffee_machine(coffee);
    printf("C님 커피 여기 있습니다.\n\n");
}
```

---- 주문을 받고 버튼을 누른다

---- 주문을 받고 버튼을 누른다

---- 주문을 받고 버튼을 누른다

### 3. Making coffee (확장) 실행결과

실행결과 ▼

```
C:\Windows\system32\cmd.exe
A님, 어떤 커피 드릴까요? <1:보통, 2:설탕, 3:블랙> 1
# 1.<자동으로> 뜨거운 물을 준비한다
# 2.<자동으로> 종이컵을 준비한다
# 3.<자동으로> 보통커피를 탄다
# 4.<자동으로> 물을 붓는다
# 5.<자동으로> 스푼으로 저어서 녹인다
A님 커피 여기 있습니다.
B님, 어떤 커피 드릴까요? <1:보통, 2:설탕, 3:블랙> 2
# 1.<자동으로> 뜨거운 물을 준비한다
# 2.<자동으로> 종이컵을 준비한다
# 3.<자동으로> 설탕커피를 탄다
# 4.<자동으로> 물을 붓는다
# 5.<자동으로> 스푼으로 저어서 녹인다
B님 커피 여기 있습니다.
C님, 어떤 커피 드릴까요? <1:보통, 2:설탕, 3:블랙>
```

## 4. 더하기 함수

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

```
int plus(int v1, int v2)
```

```
{
```

```
    int result;
```

```
    result = v1 + v2;
```

```
    return result;
```

```
}
```

--- plus() 함수 정의

--- 매개변수의 합을 구하고,  
result 값을 반환

```
int main()
```

```
{
```

```
    int hap;
```

```
    hap = plus(100, 200);
```

--- 매개변수 두 개를 지정해서 plus() 함수를 호출하고  
반환값은 hap에 저장

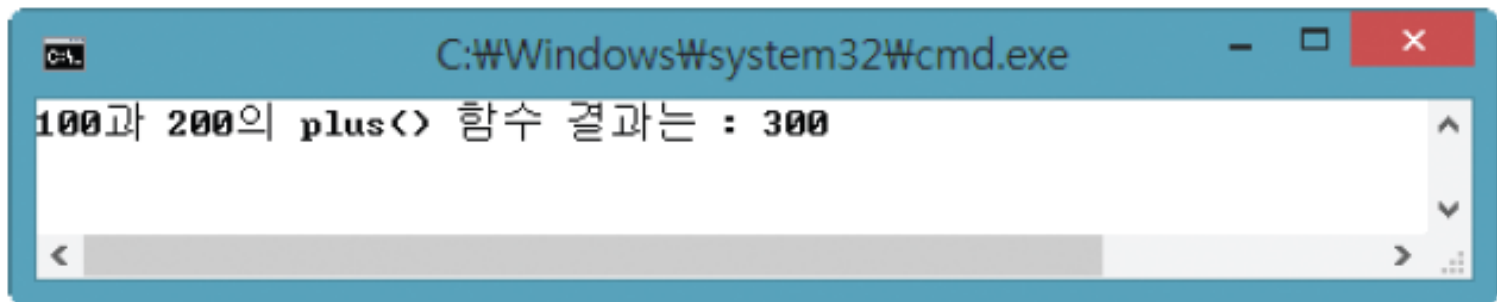
```
    printf("100과 200의 plus() 함수 결과는 : %d\n", hap);
```

```
}
```



## 4. 더하기 함수 실행결과

실행결과 ▼



A screenshot of a Windows Command Prompt window. The title bar shows the path `C:\Windows\system32\cmd.exe`. The command prompt displays the text `100과 200의 plus() 함수 결과는 : 300`. The window has a blue title bar and standard Windows window controls (minimize, maximize, close).

```
C:\Windows\system32\cmd.exe
100과 200의 plus() 함수 결과는 : 300
```

# 5. 계산기 함수

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

```
int calc(int v1, int v2, int op) --- 3개의 매개변수를 받는 함수
```

```
{
    int result;

    switch (op) --- 매개변수에 따라 연산자 결정
    {
        case 1 : result = v1 + v2; break;
        case 2 : result = v1 - v2; break;
        case 3 : result = v1 * v2; break;
        case 4 : result = v1 / v2; break;
    }

    return result; --- 계산 결과 반환
}
```

```
int main()
```

```
{
    int res; --- 변수 선언
    int oper,a,b;

    printf("계산 입력 (1:+, 2:-, 3:*, 4:/) : ");
    scanf_s("%d", &oper); --- 연산자 입력

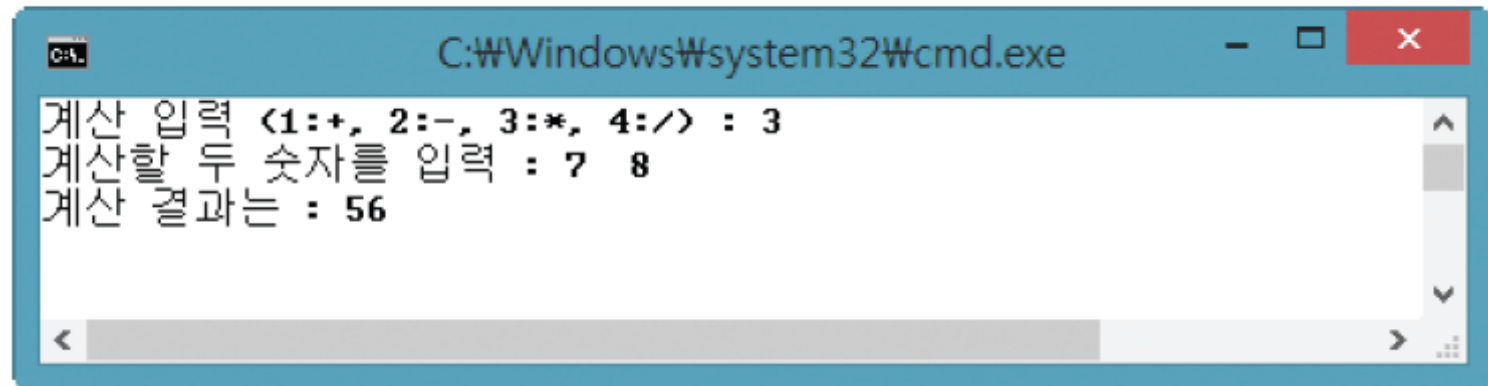
    printf("계산할 두 숫자를 입력 : ");
    scanf_s("%d %d", &a, &b); --- 피연산자 입력

    res = calc(a, b, oper); --- 계산기 함수 호출

    printf("계산 결과는 : %d\n", res);
}
```

## 5. 계산기 함수 실행결과

실행결과 ▼



```
C:\Windows\system32\cmd.exe
계산 입력 <1:+, 2:-, 3:*, 4:/> : 3
계산할 두 숫자를 입력 : 7 8
계산 결과는 : 56
```

## 6. 지역변수와 전역변수

```
#include <stdio.h>

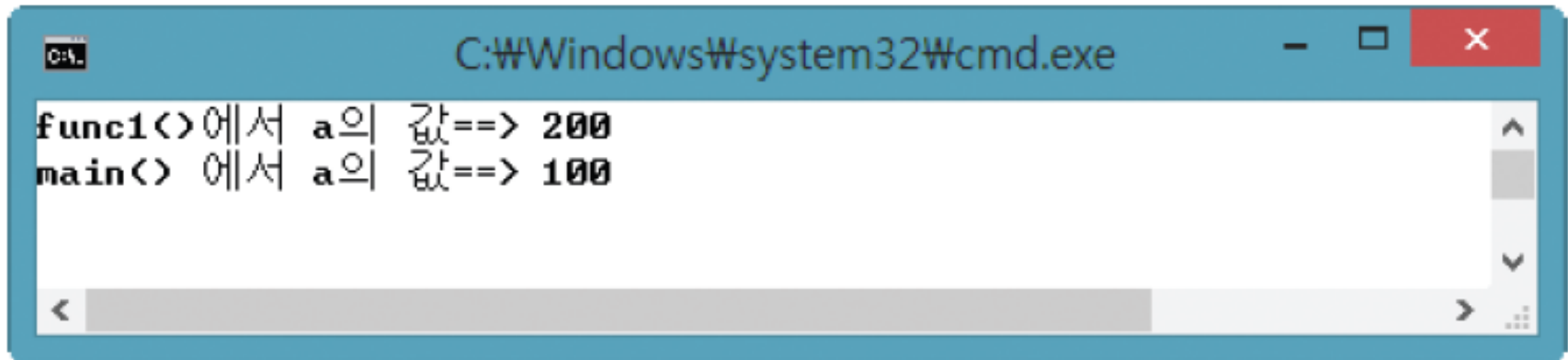
int a = 100;          --- 전역변수 a선언

void func1()
{
    int a = 200;     --- 지역변수 a 선언
    printf ("func1()에서 a의 값==> %d\n", a); --- 지역변수 출력
}

int main()
{
    func1();         --- 함수 호출
    printf ("main() 에서 a의 값==> %d\n", a); --- 전역변수 출력
}
```

## 6. 실행결과

실행결과 ▼



```
C:\Windows\system32\cmd.exe  
func1(<)에서 a의 값==> 200  
main(<)에서 a의 값==> 100
```

# 7. 함수의 반환값에 따른 차이

```
#include <stdio.h>

void func1()          --- void형 함수 선언
{
    printf("void 형 함수는 돌려줄게 없음.\n");
}

int func2()           --- int형 함수 선언
{
    return 100;
}

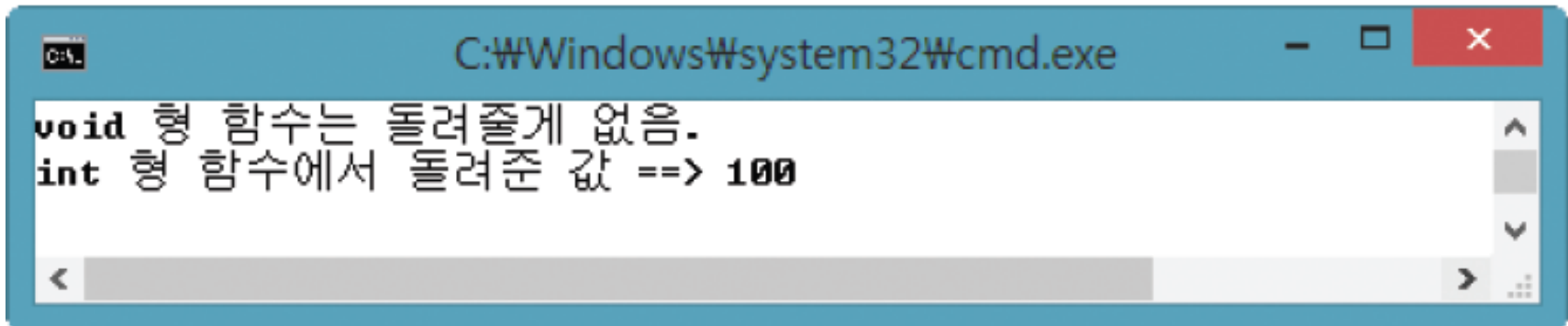
int main()            --- void형 함수 호출
{
    int a;

    func1();

    a = func2();      --- int형 함수 호출
    printf("int 형 함수에서 돌려준 값 ==> %d\n", a);
}
```

# 7. 실행 결과

실행결과 ▼



```
C:\Windows\system32\cmd.exe  
void 형 함수는 돌려줄게 없음.  
int 형 함수에서 돌려준 값 ==> 100
```

## 8. 매개변수 전달 방법 - call by value

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

```
void func1(int a)
```

```
{
```

```
    a = a + 1;
```

```
    printf("전달받은 a ==> %d\n", a);
```

```
}
```

--- 전달값 a를 1증가시키고 출력

```
void main()
```

```
{
```

```
    int a=10;
```

--- 변수 a 선언

```
    func1(a);
```

--- a를 매개변수로 함수 호출

```
    printf("func1() 실행 후의 a ==> %d\n", a);
```

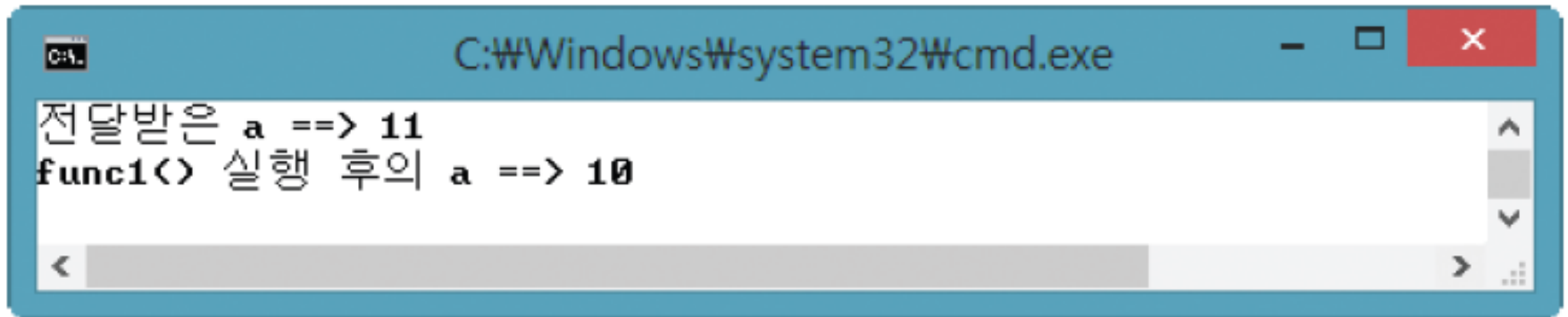
--- 함수 호출 후 a 값 출력

```
}
```



## 8. 실행 결과

실행결과 ▼



```
C:\Windows\system32\cmd.exe  
전달받은 a ==> 11  
func1(<) 실행 후의 a ==> 10
```

The image shows a Windows command prompt window with a blue title bar. The title bar contains the text "C:\Windows\system32\cmd.exe" and standard window control buttons (minimize, maximize, close). The main area of the window is white and contains two lines of black text: "전달받은 a ==> 11" and "func1(<) 실행 후의 a ==> 10". The window has a scroll bar on the right side and a horizontal scrollbar at the bottom.

## 9. 매개변수 전달 방법 - call by reference

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

```
void func1(int *a)
```

--- 매개변수로 주소를 받음

```
{
```

```
  *a = *a + 1;
```

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

--- a가 가리키는 곳의 실제값+1을 수행

--- a가 가리키는 곳의 실제값을 출력

```
}
```

```
void main()
```

```
{
```

```
  int a=10;
```

--- a를 10으로 초기화

```
  func1(&a);
```

--- 함수 호출 시 a의 주소를 전달

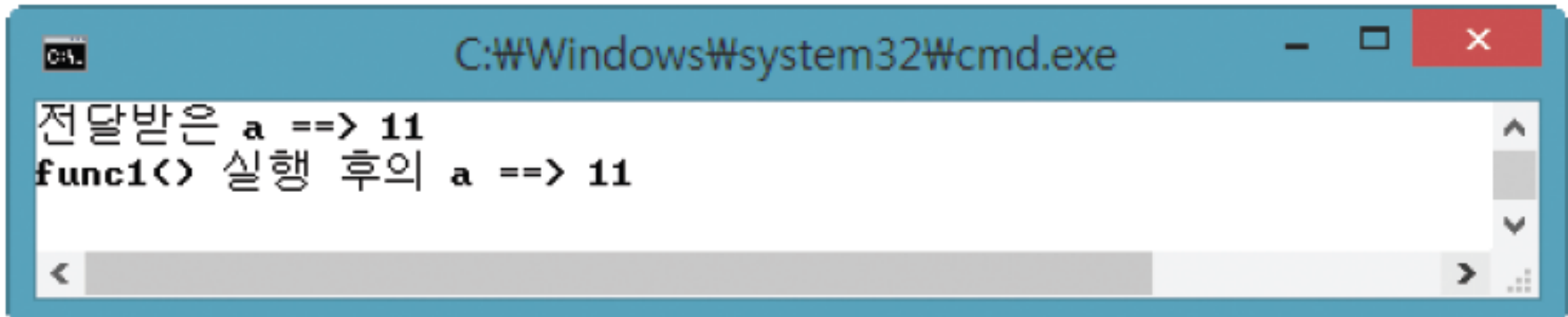
```
  printf("func1() 실행 후의 a ==> %d\n", a);
```

--- 함수 호출 후 a값 출력

```
}
```

## 9. 매개변수 전달 방법 - call by reference

실행결과 ▼



```
C:\Windows\system32\cmd.exe
전달받은 a ==> 11
func1() 실행 후의 a ==> 11
```

# 10. 매개변수 전달 방법 비교

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

```
void func1 (char a, char b)
{
    int imsi;
    imsi = a;
    a = b;
    b = imsi;
}
```

---- 매개변수가 값인 함수

---- 두 문자를 교환

```
void func2 (char *a, char *b)
{
    int imsi;
    imsi = *a;
    *a = *b;
    *b = imsi;
}
```

---- 매개변수가 주소인 함수

---- 두 문자를 교환

```
void main()
{
    char x = 'A', y = 'Z';
    printf("원래 값 : x=%c, y=%c\n", x, y);
    func1(x, y);
    printf("값을 전달한 후 : x=%c, y=%c\n", x, y);
    func2(&x, &y);
    printf("주소를 전달한 후: x=%c, y=%c\n", x, y);
}
```

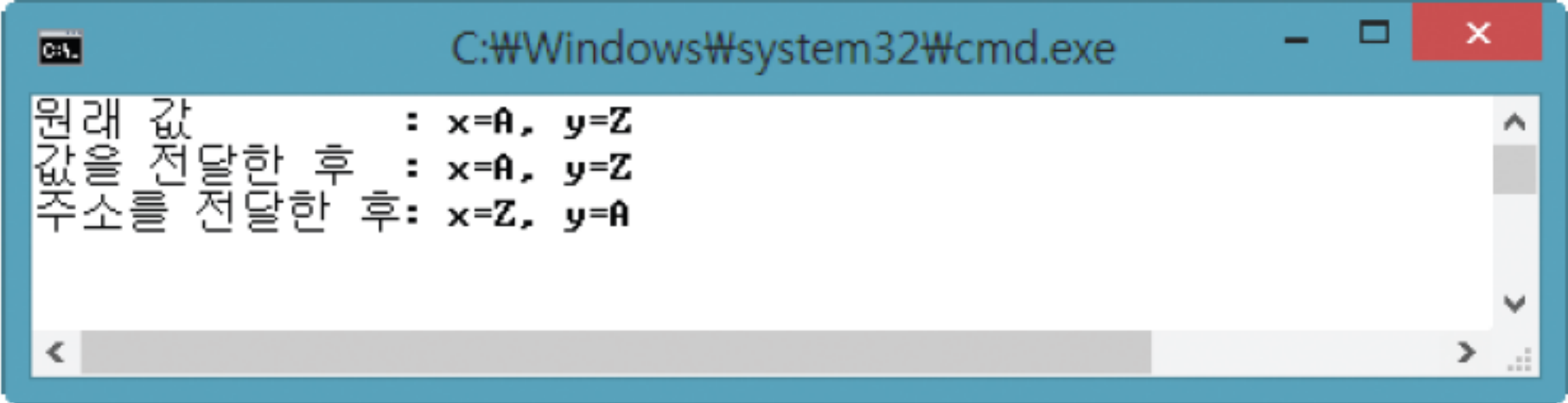
---- 원래 문자를 출력

---- 값을 전달하여 func1() 함수 호출

---- 주소를 전달하여 func2() 함수 호출

# 10. 매개변수 전달 방법 비교

실행결과 ▼



```
C:\Windows\system32\cmd.exe
원래 값      : x=A, y=Z
값을 전달한 후 : x=A, y=Z
주소를 전달한 후: x=Z, y=A
```

# 참고문헌

- 열혈 C 프로그래밍, 윤성우, 오렌지미디어
- 쉽게 풀어쓴 C언어 Express, 천인국, 생능출판사
- 뇌를 자극하는 C 프로그래밍, 서현우, 한빛미디어
- C언어 for Beginner, 우재남, 한빛미디어