

Do it! C언어 입문

20164389 멀티미디어공학과 박병주

김상엽 지음 [이지스 퍼블리싱]



📖 목차

6장 : 표준 출력 함수
7장 : 연산자



표준 출력 함수



표준 출력 함수

라이브러리

라이브러리란?

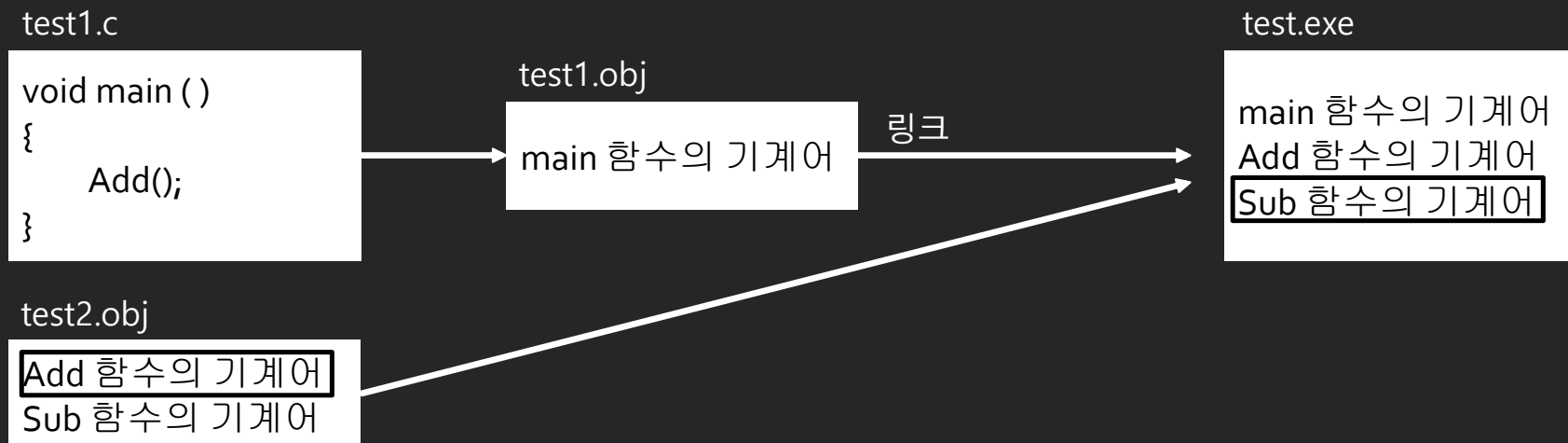
- C언어에서 좀 더 효과적으로 함수를 관리할 수 있도록 하는 기술임
- 지속적으로 업데이트가 필요한 함수들만 소스 파일에 유지하고 나머지 거의 내용이 바뀌지 않는 함수들을 따로 분리하여 라이브러리로 관리



표준 출력 함수

라이브러리

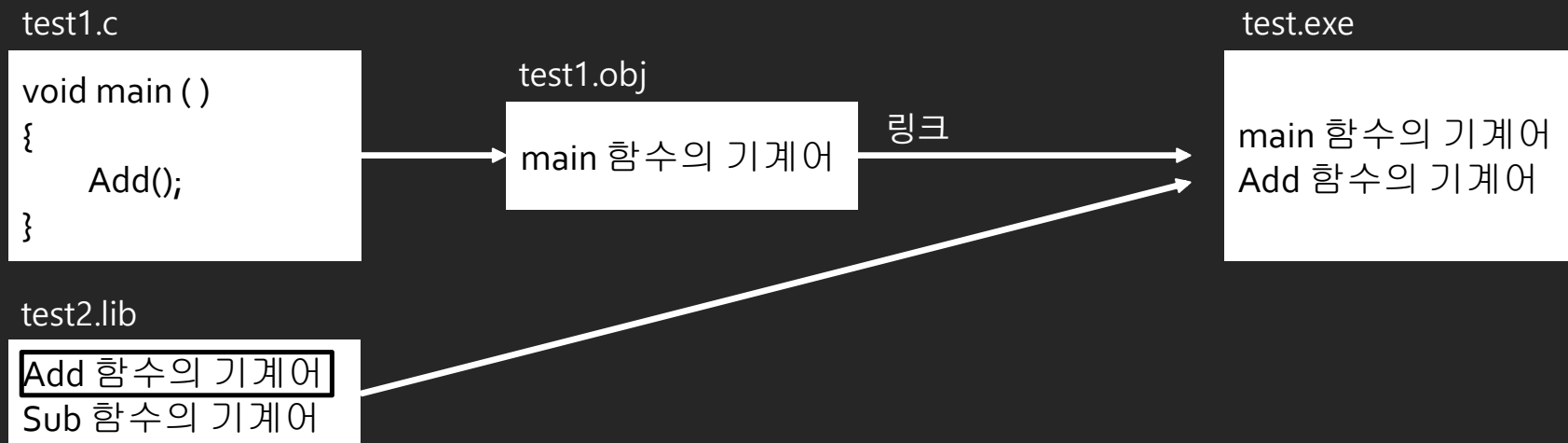
object파일 사용



표준 출력 함수

라이브러리

library파일 사용



표준 출력 함수

헤더 파일

main.c 소스 파일

```
void main ( )  
{  
    int result1, result2, result3, result4;  
    result1 = Add(2, 3); /*오류발생 */  
    result2 = Sub(2, 3); /*오류발생 */  
    result3 = Mul(2, 3); /*오류발생 */  
    result4 = Div(2, 3); /*오류발생 */  
}
```

MyMath.lib 라이브러리 파일

Add, Sub, Mul, Div
함수가 있지만 기계어로
번역되어 내용을 확인할
수 없음



표준 출력 함수

헤더 파일

main.c 소스 파일

(MyMath.lib 파일에 있는 함수를 사용하기
위해 각 함수 원형을 선언함)
`int Add(int value1, int value2);`
`int Sub(int value1, int value2);`
`int Mul(int value1, int value2);`
`int Div(int value1, int value2);`

MyMath.lib 라이브러리 파일

Add, Sub, Mul, Div
함수가 있지만 기계어로
번역되어 내용을 확인할
수 없음

※ 사용할 함수가 많아지면 하나하나 선언하기 번거로워짐



표준 출력 함수

헤더 파일

※ 사용할 함수가 많아지면 하나하나 선언하기 번거로워짐

-> 헤더파일 : 함수 원형들을 특정 파일에 미리 선언해둔 파일임

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



표준 출력 함수

전처리기

전처리기란?

컴파일러가 원하는 사항을 컴파일러에 직접 지시하는 문법



표준 출력 함수

#include 전처리기

컴파일러에 프로그래머가 명시한 파일을 읽도록 지시함

#include "읽을 파일 이름" 으로 표기함

```
#include "MyMath.h"  
#include "C:\download\MyMath.h"
```



표준 출력 함수

#include 전처리기

main.c 소스 파일

```
#include "MyMath.h"
void main ( )
{
    int result1, result2, result3, result4;
    result1 = Add(2, 3); /*오류발생 */
    result2 = Sub(2, 3); /*오류발생 */
    result3 = Mul(2, 3); /*오류발생 */
    result4 = Div(2, 3); /*오류발생 */
}
```

Mymath.h 헤더 파일

(MyMath.lib 파일에 있는 함수를 사용하기 위해 각 함수 원형을 선언함)

```
int Add(int value1, int value2);
int Sub(int value1, int value2);
int Mul(int value1, int value2);
int Div(int value1, int value2);
```

MyMath.lib 라이브러리 파일

Add, Sub, Mul, Div 함수가 있지만 기계어로 번역되어 내용을 확인할 수 없음



표준 출력 함수

#define 전처리기

상수나 명령문을 치환하는 문법

#define 치환할 이름 치환될 내용

```
#define MAX_COUNT 3
```

```
#define POW_VALUE(a) (a * a)  
int data = POW_VALUE(3);
```

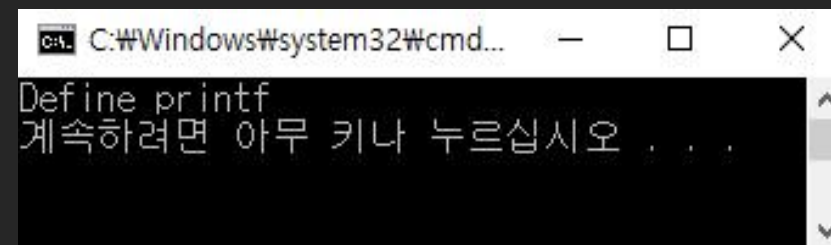


표준 출력 함수

#define 전처리기

```
#include <stdio.h>
#define PRINT printf("Define printf #n")

int main()
{
    PRINT;
    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd...'. The window contains two lines of text: 'Define printf' on the first line and '계속하려면 아무 키나 누르십시오 . . .' (Press any key to continue) on the second line.



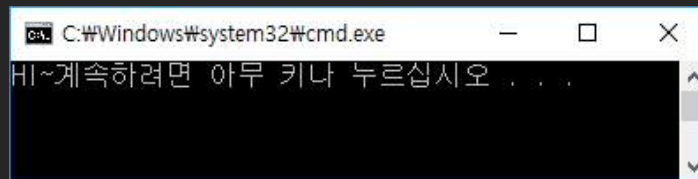
표준 출력 함수

표준 출력 함수

putchar 함수

```
#include <stdio.h>

void main()
{
    putchar('H');
    putchar('I');
    putchar('~');
}
```



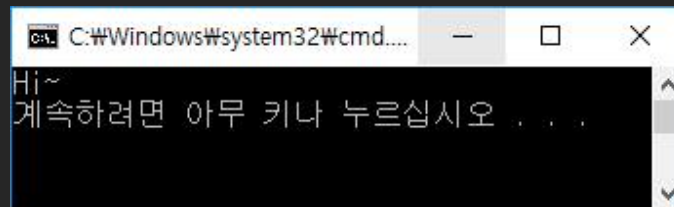
C:\Windows\system32\cmd.exe

Hi~계속하려면 아무 키나 누르십시오 . . .

puts 함수

```
#include <stdio.h>

void main()
{
    puts("Hi~");
}
```



C:\Windows\system32\cmd.exe

Hi~
계속하려면 아무 키나 누르십시오 . . .



표준 출력 함수

문자열 출력 함수 printf

변수 값을 일정한 형식으로 출력하는 함수임

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

```
void main()
```

```
{
```

```
    int data1 = 3;
```

```
    int data2 = 5;
```

```
    printf("data1 = %d, data2 = %d \n", data1, data2);
```

```
}
```

C:\Windows\system32\cmd...

data1 = 3, data2 = 5

계속하려면 아무 키나 누르십시오 . . .

키워드	%d	%f	%c	%s
출력 형식	정수(10진수)	실수	문자	문자열

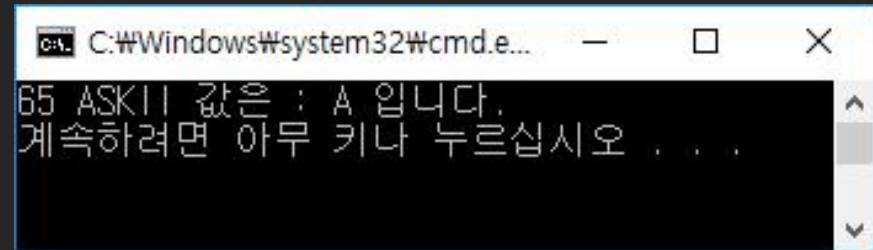


표준 출력 함수

문자열 출력 함수 printf

```
#include <stdio.h>

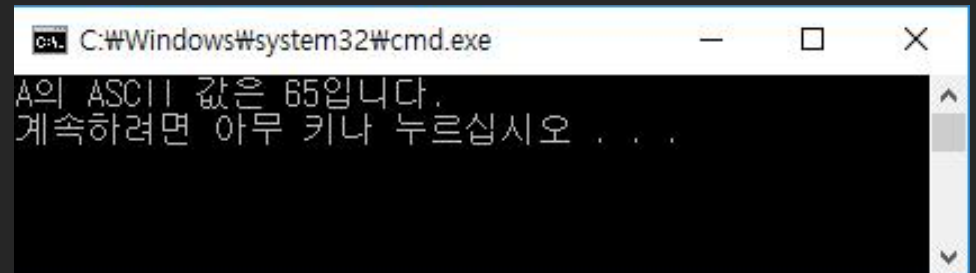
void main()
{
    printf("65 ASCII 값은 : %c 입니다. \n", 65);
}
```



C:\Windows\system32\cmd.e...
65 ASCII 값은 : A 입니다.
계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>

void main()
{
    char data = 65;
    printf("%c의 ASCII 값은 %d입니다.\n", data, data);
}
```



C:\Windows\system32\cmd.exe
A의 ASCII 값은 65입니다.
계속하려면 아무 키나 누르십시오 . . .

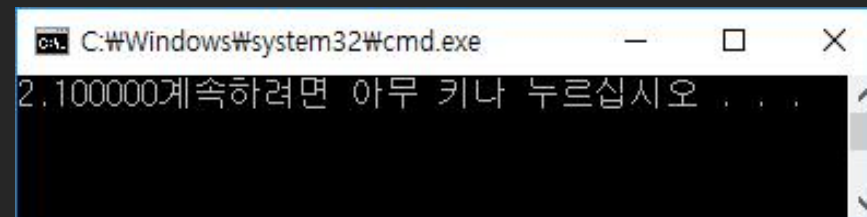


표준 출력 함수

문자열 출력 함수 printf

```
#include <stdio.h>

void main()
{
    float value = 2.1f;
    printf("%f", value);
}
```

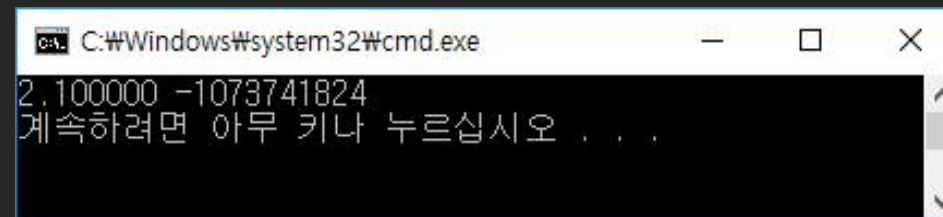


C:\Windows\system32\cmd.exe

2.100000계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>

void main()
{
    float value = 2.1f;
    printf("%f %d %n", value, value);
}
```



C:\Windows\system32\cmd.exe

2.100000 -1073741824
계속하려면 아무 키나 누르십시오 . . .



연산자

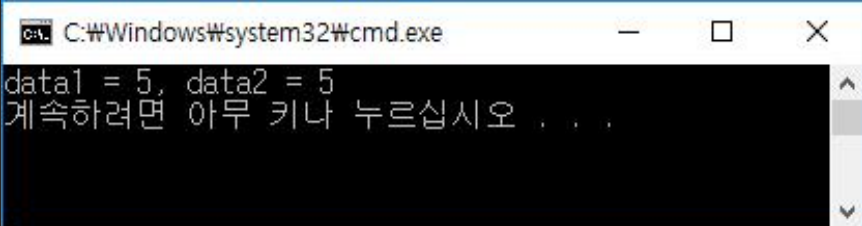


연산자

대입 연산자

```
#include <stdio.h>

void main()
{
    int data1, data2;
    data1 = 5;
    data2 = data1;
    printf("data1 = %d, data2 = %d \n", data1, data2);
}
```



C:\Windows\system32\cmd.exe

```
data1 = 5, data2 = 5
계속하려면 아무 키나 누르십시오 . . .
```

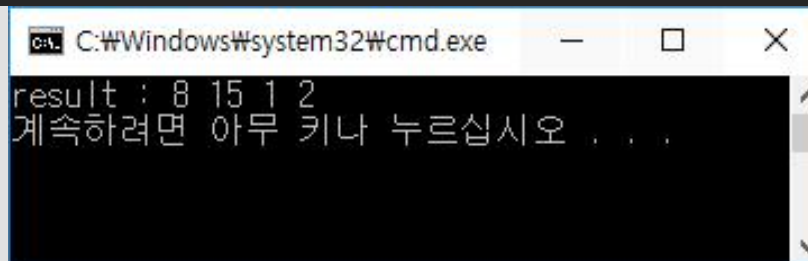
x = 3 [x에 3을 대입하는 것임]



연산자

산술 연산자

```
void main()  
{  
    int data1 = 5, data2 = 3;  
  
    int result1 = data1 + data2;  
    int result2 = data1 * data2;  
    int result3 = data1 / data2;  
    int result4 = data1 % data2;  
  
    printf("result : %d %d %d %d \n", result1, result2, result3, result4);  
}
```



C:\Windows\system32\cmd.exe
result : 8 15 1 2
계속하려면 아무 키나 누르십시오 . . .

+, -, *, /, %는 산술 연산자로
각각 더하기, 빼기, 곱하기,
나누기, 나머지 연산을 뜻함



연산자

증감 연산자

전위형

```
int i = 5, sum;  
sum = ++i;
```

후위형

```
int i = 5, sum;  
sum = i++;
```



연산자

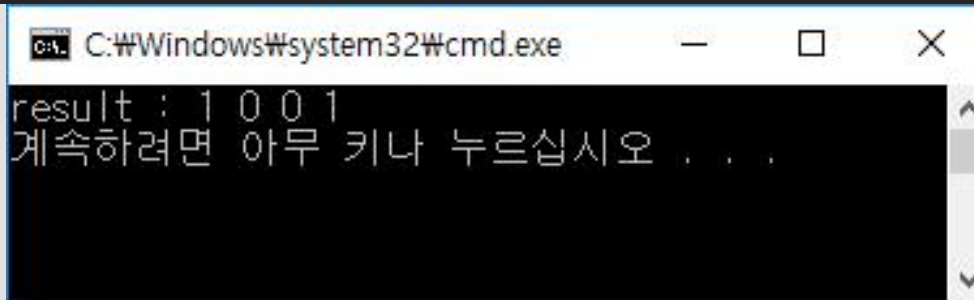
관계 연산자

```
#include <stdio.h>

void main()
{
    int data1 = 5, data2 = 3;

    int result1 = data1 > data2;
    int result2 = data1 <= data2;
    int result3 = data1 == data2;
    int result4 = data1 != data2;

    printf("result : %d %d %d %d \n", result1, result2, result3, result4);
}
```



```
C:\Windows\system32\cmd.exe
result : 1 0 0 1
계속하려면 아무 키나 누르십시오...
```

관계 연산자	연산 특성	관계 연산자	연산 특성
$A < B$	A가 B보다 작으면 참	$A > B$	A가 B보다 크면 참
$A \leq B$	A가 B보다 작거나 같으면 참	$A \geq B$	A가 B보다 크거나 같으면 참
$A == B$	A와 B가 같으면 참	$A != B$	A와 B가 같지 않으면 참



연산자

논리 연산자

A	B	A && B	A B	!A
0	0	0	0	1
0	1	0	1	
1	0	0	1	0
1	1	1	1	

논리 연산자	연산 특성
&&	A와 B가 모두 참이어야 결과 값이 참, 하나라도 거짓이면 거짓
	A 또는 B가 참이면 참, 둘 다 거짓일 때만 거짓
!	A가 거짓이면 참, 참이면 거짓



연산자

논리 연산자

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

```
void main()
```

```
{
```

```
    int data1 = 5, data2 = 3;
```

```
    int result1 = 0 || 1;
```

```
    int result2 = 3 && -1;
```

```
    int result3 = data1 == 3 || data2 == 3;
```

```
    int result4 = data1 == 3 && data2 == 3;
```

```
    int result5 = ! data1;
```

```
    printf("result : %d %d %d %d %d\n", result1, result2, result3, result4, result5);
```

```
}
```

C:\Windows\system32\cmd.exe

result : 1 1 1 0 0

계속하려면 아무 키나 누르십시오 . . .





감사합니다.

The End.