

함수와 RANDOM값

2025 프로그래밍 기초

목표

- 함수를 생성하고 사용
- RANDOM 값을 생성하고 사용

(연습) 반복문 : for

- 언제사용?) 동일한 코드(명령)를 100번 **반복**하고 싶다.

```
for( int i=0 ; i<100 ; i++ )  
{  
    //반복할 코드(명령)  
    ....  
}
```

(연습) 반복문 : for

- 언제사용?) 동일한 코드(명령)를 100번 **반복**하고 싶다.

```
          초기화      탈출조건      증감
for( int i=0 ; i<100 ; i++ )
{
    //반복할 코드(명령)
    ....
}
```

(연습) 반복문 : for

- 1~10까지 홀수를 곱(*)한 결과



```
PS C:\Users\Juh  
result = 3840  
PS C:\Users\Juh
```

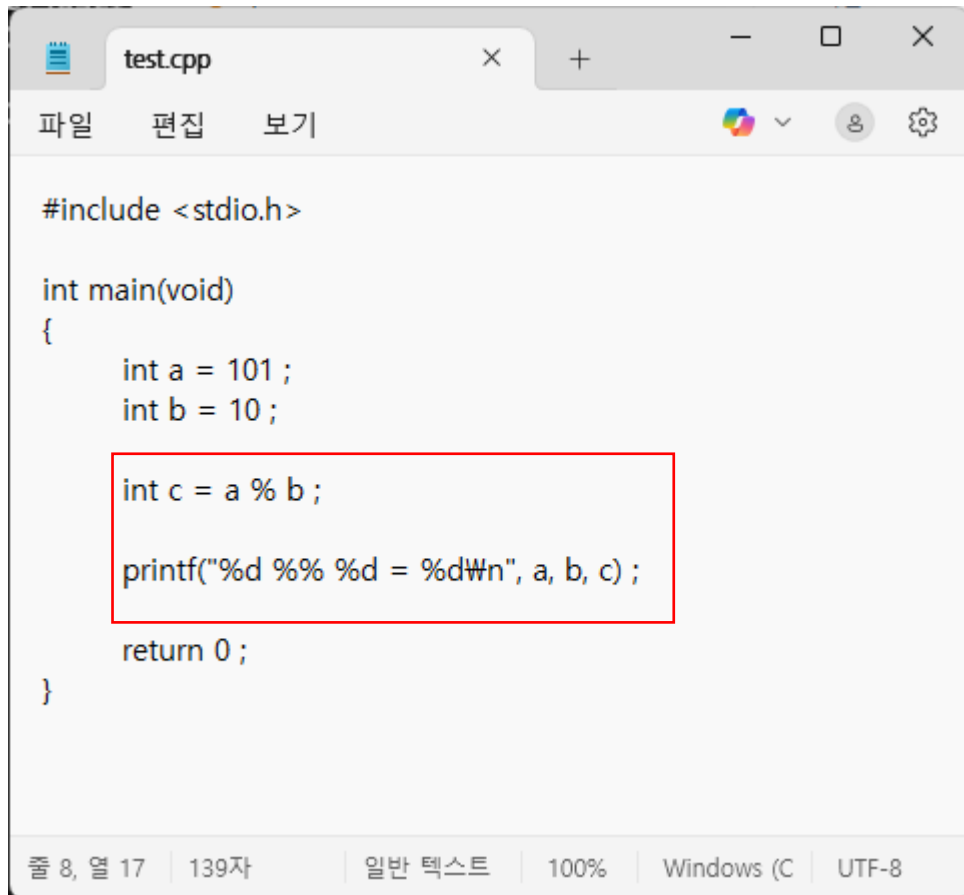
(연습) 사칙연산 + α

- 더하기 : $A = B + C ;$
- 빼 기 : $A = B - C ;$
- 곱하기 : $A = B * C ;$
- 나누기 : $A = B / C ;$
- 나머지 : $A = B \% C ;$

연습 :

2개의 숫자(정수)를 입력 받아 나머지 계산

- 아래의 예제 코드를 작성 -> 컴파일 -> 실행



```
#include <stdio.h>

int main(void)
{
    int a = 101 ;
    int b = 10 ;

    int c = a % b ;

    printf("%d %% %d = %d\n", a, b, c) ;

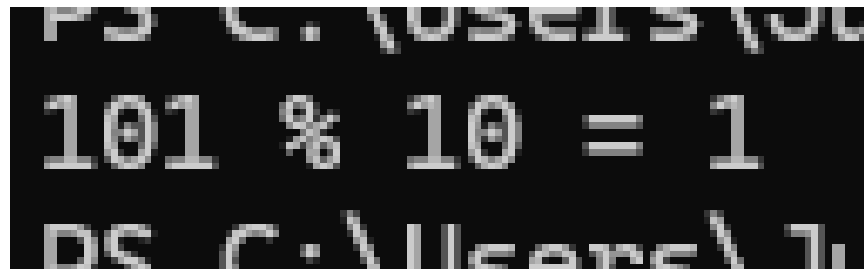
    return 0 ;
}
```

The screenshot shows a code editor window titled 'test.cpp'. The code is a C++ program that calculates the remainder of 101 divided by 10. The line `int c = a % b ;` is highlighted with a red rectangle. The status bar at the bottom indicates '줄 8, 열 17', '139자', '일반 텍스트', '100%', 'Windows (C', and 'UTF-8'.

컴파일 → `> C:\mingw64\bin\gcc -o test test.cpp`

실행 → `> ./test`

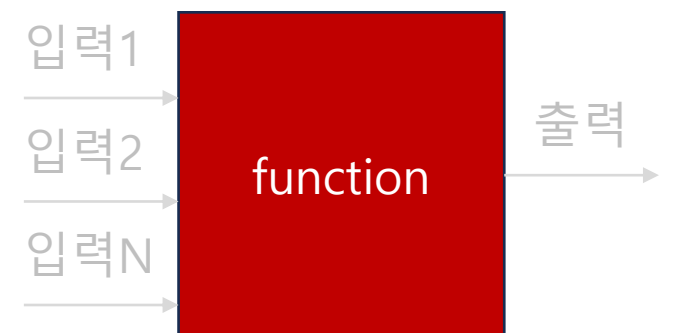
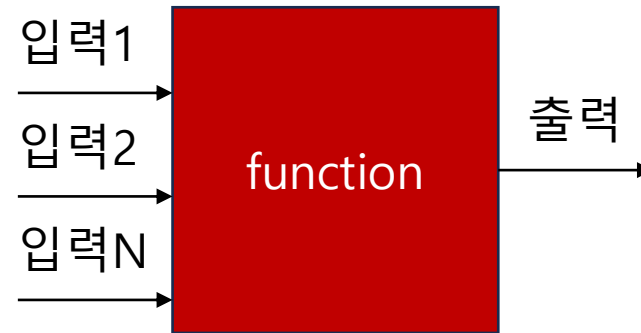
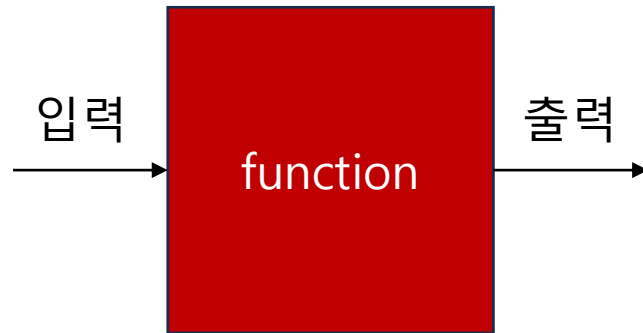
실행 결과



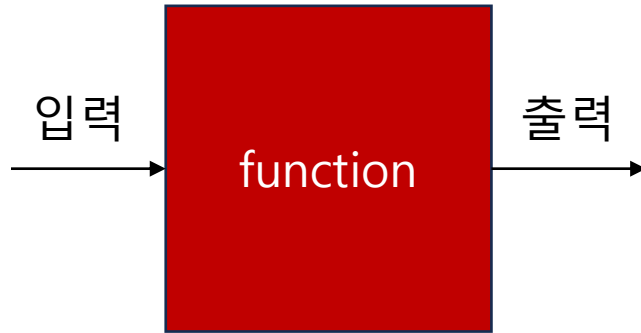
```
PS C:\Users\ju...
101 % 10 = 1
PS C:\Users\ju...
```

The screenshot shows a command prompt window with the output of the program: `101 % 10 = 1`. The prompt is `PS C:\Users\ju...`.

함수(function)

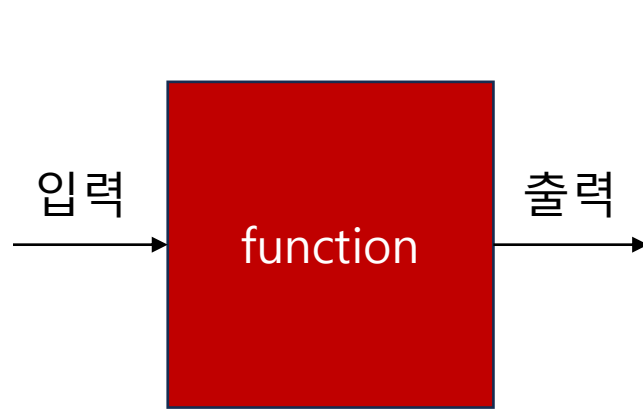


함수(function)



```
int function(int a)
{
    a = a + 10 ;
    return a ;
}
```

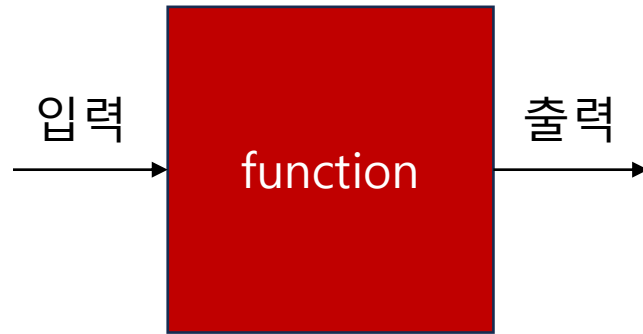
함수(function)



출력 자료형(return type)

```
int function(int a)
{
    a = a + 10 ;
    return a ;
}
```

함수(function)

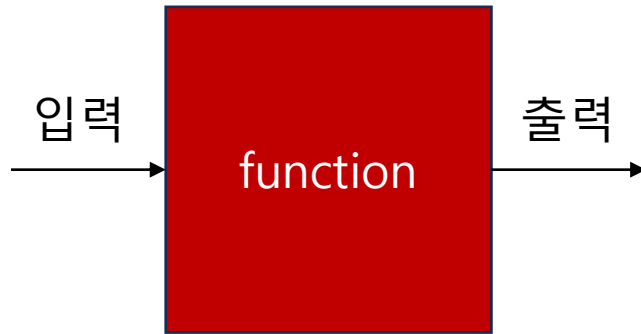


함수 이름(함수명)

```
int function(int a)
{
    a = a + 10 ;
    return a ;
}
```

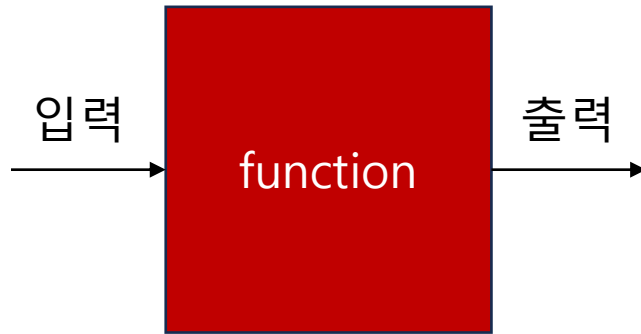
함수(function)

입력 인자(input argument)



```
int function(int a)
{
    a = a + 10 ;
    return a ;
}
```

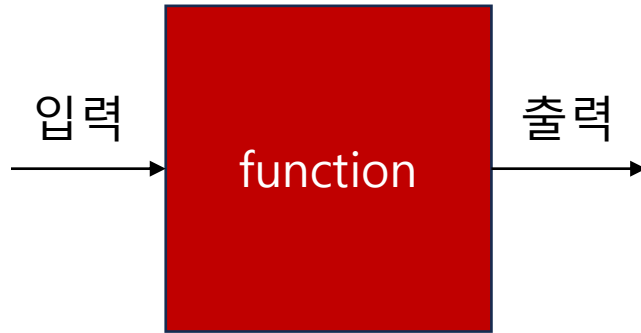
함수(function)



```
int function(int a)
{
    a = a + 10 ;
    return a ;
}
```

명령문

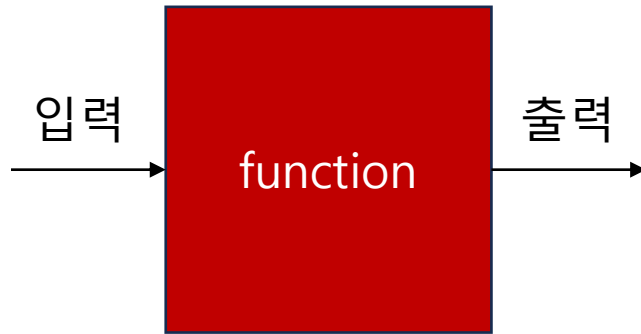
함수(function)



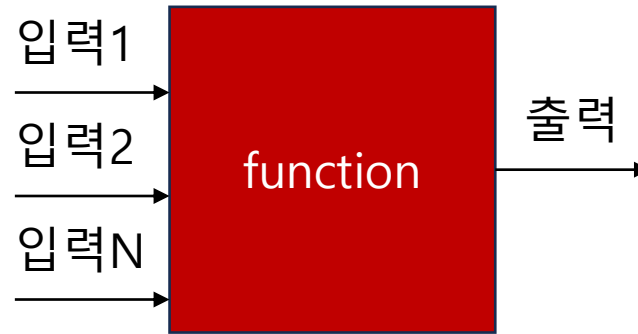
```
int function(int a)
{
    a = a + 10 ;
    return a ;
}
```

The diagram shows a C++ function definition. The return type `int` is circled in red. A dashed red line connects the `int` to the `return a ;` statement, which is also circled in red. The opening curly brace `{` is also circled in red.

함수(function)



```
int function(int a)
{
    a = a + 10 ;
    return a ;
}
```



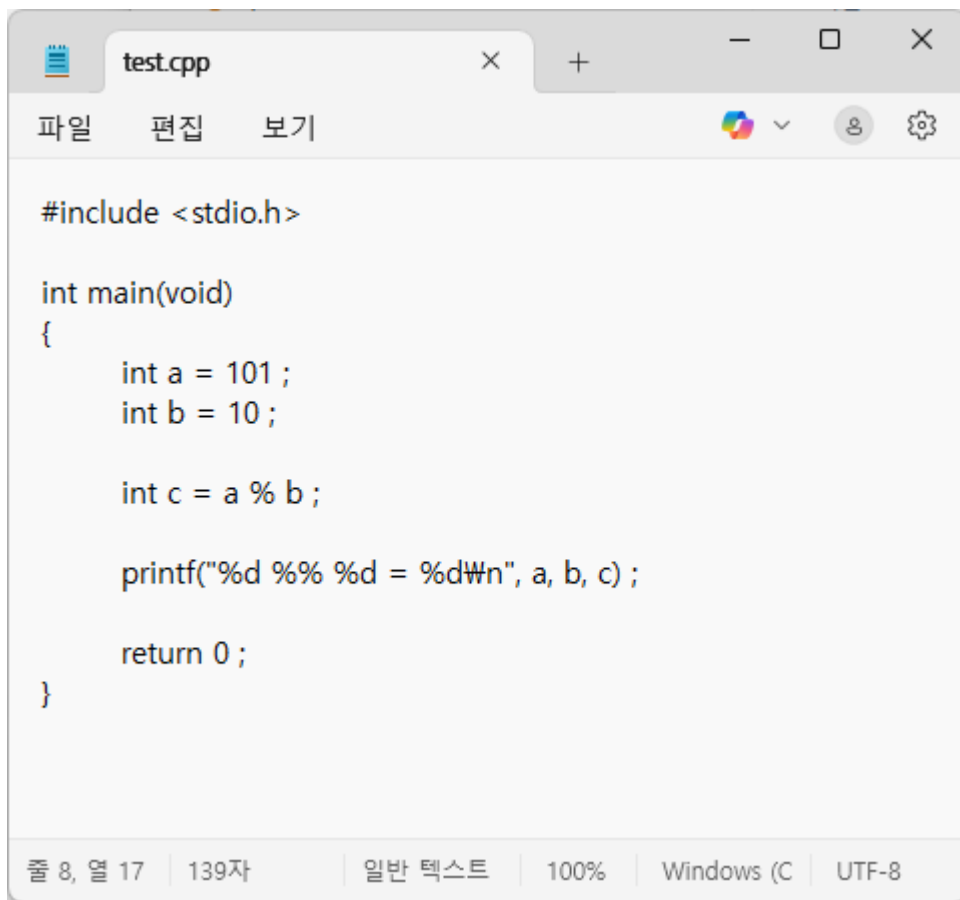
```
int function(int a, int b)
{
    int c = a + b ;
    return c ;
}
```



```
void function(void)
{
    //함수 명령
}
```

함수(function)

함수를 찾아라!



```
#include <stdio.h>

int main(void)
{
    int a = 101 ;
    int b = 10 ;

    int c = a % b ;

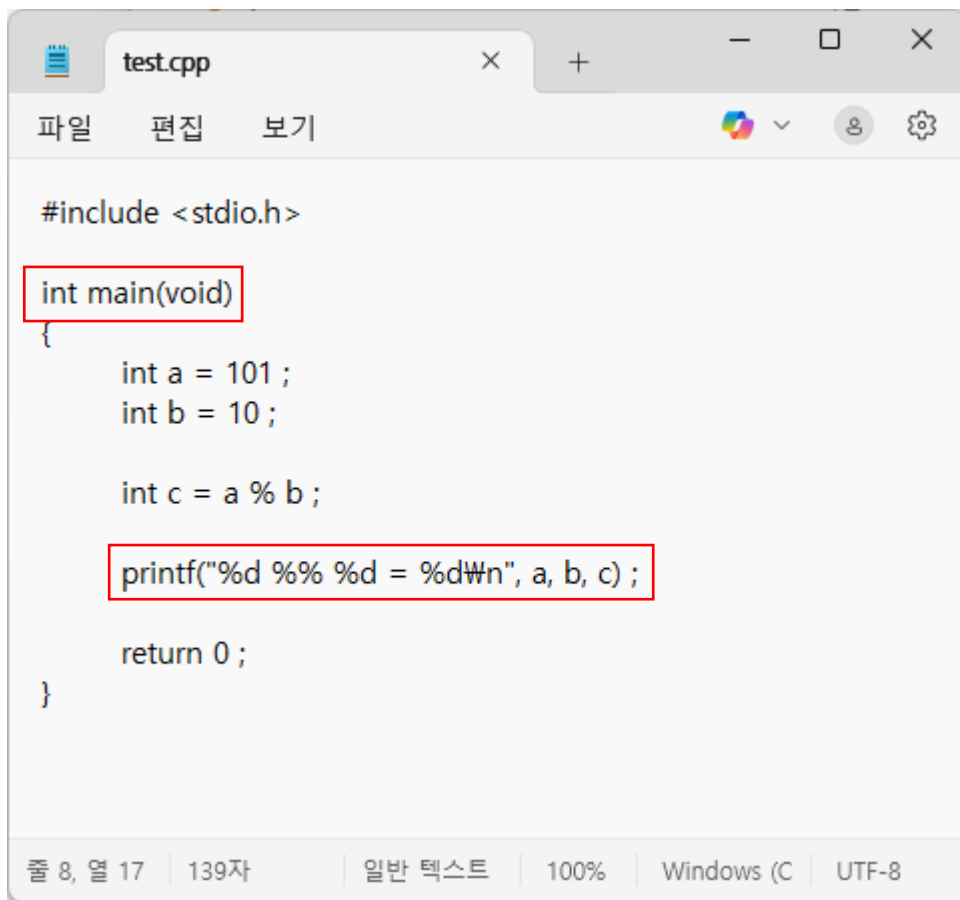
    printf("%d %% %d = %d\n", a, b, c) ;

    return 0 ;
}
```

줄 8, 열 17 | 139자 | 일반 텍스트 | 100% | Windows (C | UTF-8

함수(function)

함수를 찾아라!



```
test.cpp
파일 편집 보기
#include <stdio.h>
int main(void)
{
    int a = 101 ;
    int b = 10 ;

    int c = a % b ;

    printf("%d %% %d = %d\n", a, b, c) ;

    return 0 ;
}
```

줄 8, 열 17 | 139자 | 일반 텍스트 | 100% | Windows (C | UTF-8

함수(function)

나만의 함수 만들기

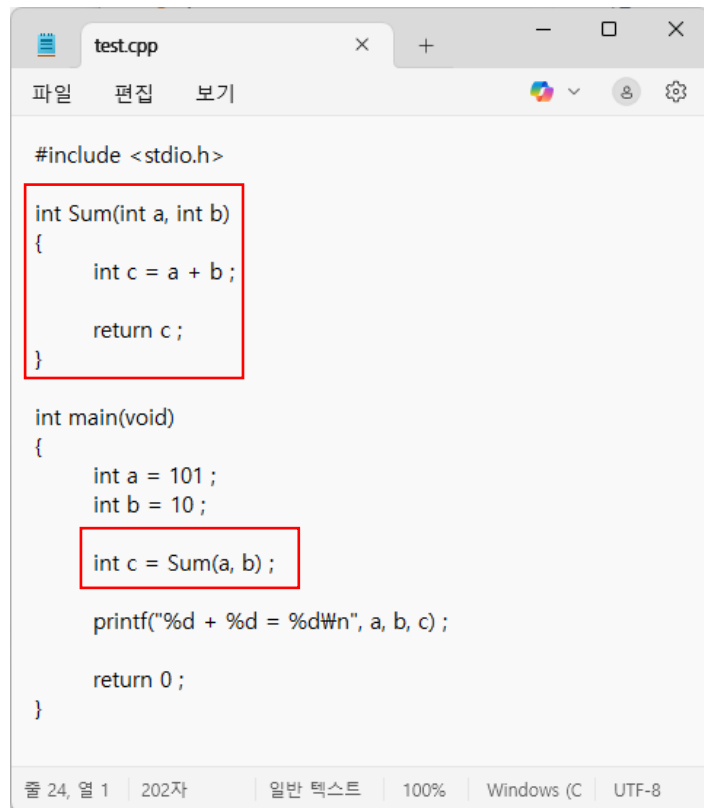
- 2개의 정수를 입력 받아 더한 결과를 출력하는 함수

```
int Sum(int a, int b)  
{  
    int c = a + b ;  
  
    return c ;  
}
```

함수(function)

나만의 함수 만들기

- 2개의 정수를 입력 받아 더한 결과를 출력하는 함수



```
#include <stdio.h>

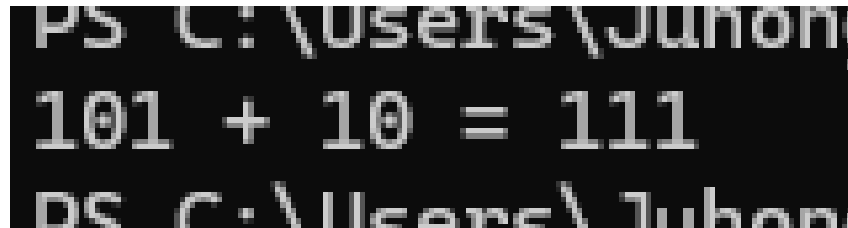
int Sum(int a, int b)
{
    int c = a + b ;
    return c ;
}

int main(void)
{
    int a = 101 ;
    int b = 10 ;
    int c = Sum(a, b) ;
    printf("%d + %d = %d\n", a, b, c) ;
    return 0 ;
}
```

컴파일 → `> C:\mingw64\bin\gcc -o test test.cpp`

실행 → `> ./test`

실행 결과



```
PS C:\Users\Junhong>
101 + 10 = 111
PS C:\Users\Junhong>
```



아래의 코드를 완성 하시오

```
test.cpp
파일 편집 보기
#include <stdio.h>

int Sum(int a, int b)
{
    int c = a + b;

    return c;
}

int Minus(int a, int b)
{
}

int Multiple(int a, int b)
{
}

int Divid(int a, int b)
{
}

int main(void)
{
    int a = 101;
    int b = 10;

    int sum = Sum(a, b);
    int minus = Minus(a, b);
    int multiple = Multiple(a, b);
    int divid = Divid(a, b);

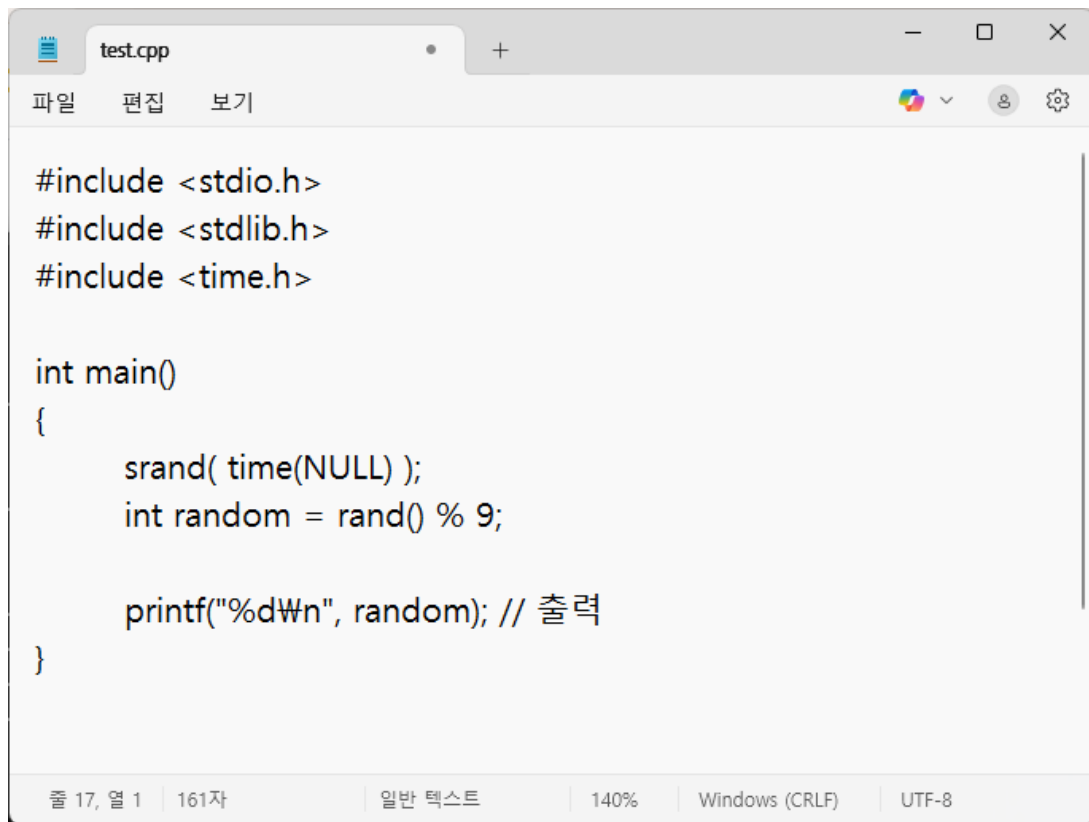
    printf("%d + %d = %d\n", a, b, sum);
    printf("%d - %d = %d\n", a, b, minus);
    printf("%d * %d = %d\n", a, b, multiple);
    printf("%d / %d = %d\n", a, b, divid);

    return 0;
}
```

```
PS C:\Users\Junhong\On
101 + 10 = 111
101 - 10 = 91
101 * 10 = 1010
101 / 10 = 10
PS C:\Users\Junhong\On
```

RANDOM

임의의 숫자 만들기(난수 생성기)



```
test.cpp
파일 편집 보기
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    srand( time(NULL) );
    int random = rand() % 9;

    printf("%d\n", random); // 출력
}
```

줄 17, 열 1 | 161자 | 일반 텍스트 | 140% | Windows (CRLF) | UTF-8

컴파일 → **> C:\mingw64\bin\gcc -o test test.cpp**

실행 → **> ./test**

실행 결과

```
PS C:\Users\Juhong\OneDrive - 주식회사 랑데뷰\바탕 화면\work
2
PS C:\Users\Juhong\OneDrive - 주식회사 랑데뷰\바탕 화면\work
3
PS C:\Users\Juhong\OneDrive - 주식회사 랑데뷰\바탕 화면\work
0
PS C:\Users\Juhong\OneDrive - 주식회사 랑데뷰\바탕 화면\work
```



0~9사이의 임의의 숫자 3개 생성

```
PS C:\Users\Jul  
5, 9, 7  
PS C:\Users\Jul  
PS C:\Users\Jul  
7, 9, 5  
PS C:\Users\Jul  
PS C:\Users\Jul  
7, 6, 1  
PS C:\Users\Jul
```

숫자야구 게임 만들기



- 1) 세 자리 수를 정합니다.
 - 2) 짝이 말한 숫자와 내가 정한 숫자를 비교해봅니다.
 - 3) 다음 규칙에 따라 짝에게 말합니다. 짝과 번갈아가며 숫자를 맞혀봅니다.
 - * 스트라이크(S): 자리와 숫자가 모두 일치하는 경우
 - * 볼(B): 숫자가 같되 자리가 틀리는 경우
 - * 아웃(O): 그 어떤 숫자도 내가 정한 숫자와 다른 경우
- 예) 내가 정한 숫자가 287이라면 다음과 같습니다.

| | | | | | | | | | |
|---|---|---|--------------|---|----|---|-----|----|---|
| 3 | 6 | 9 | 아웃 | S | ○○ | B | ○○○ | 아웃 | ● |
| 6 | 7 | 2 | 2볼 | S | ○○ | B | ●●○ | 아웃 | ○ |
| 2 | 8 | 0 | 2스트라이크 | S | ●● | B | ○○○ | 아웃 | ○ |
| 2 | 7 | 8 | 1스트라이크 2볼 | S | ●○ | B | ●●○ | 아웃 | ○ |
| 2 | 8 | 7 | 홈런! (승리) | S | ○○ | B | ○○○ | 아웃 | ○ |
| | | | | | | | | 홈런 | ● |

- 4) 먼저 숫자를 맞힌 사람이 이깁니다.