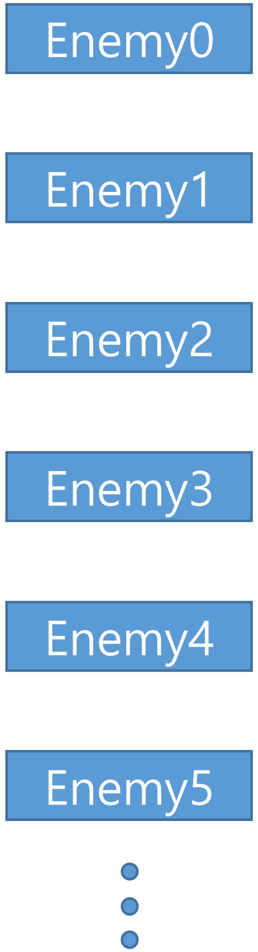
Array

DIGIPEN
DRAGONS

Jinwoo Choi

Array(배열)이란?

Same data type
Contiguous memory space



Enemy0

Enemy1

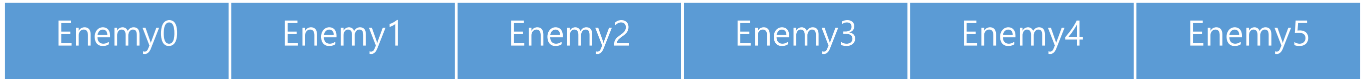
Enemy2

Enemy3

Enemy4

Enemy5

```
//Do Enemy0 Update()

//Do Enemy1 Update()

//Do Enemy2 Update()

//Do Enemy3 Update()

//Do Enemy4 Update()

//Do Enemy5 Update()
```



Enemy0	Enemy1	Enemy2	Enemy3	Enemy4	Enemy5

```
for(int i=0; i<6; i++)
{
    //Do Enemy[i] Update()
}
```

Array

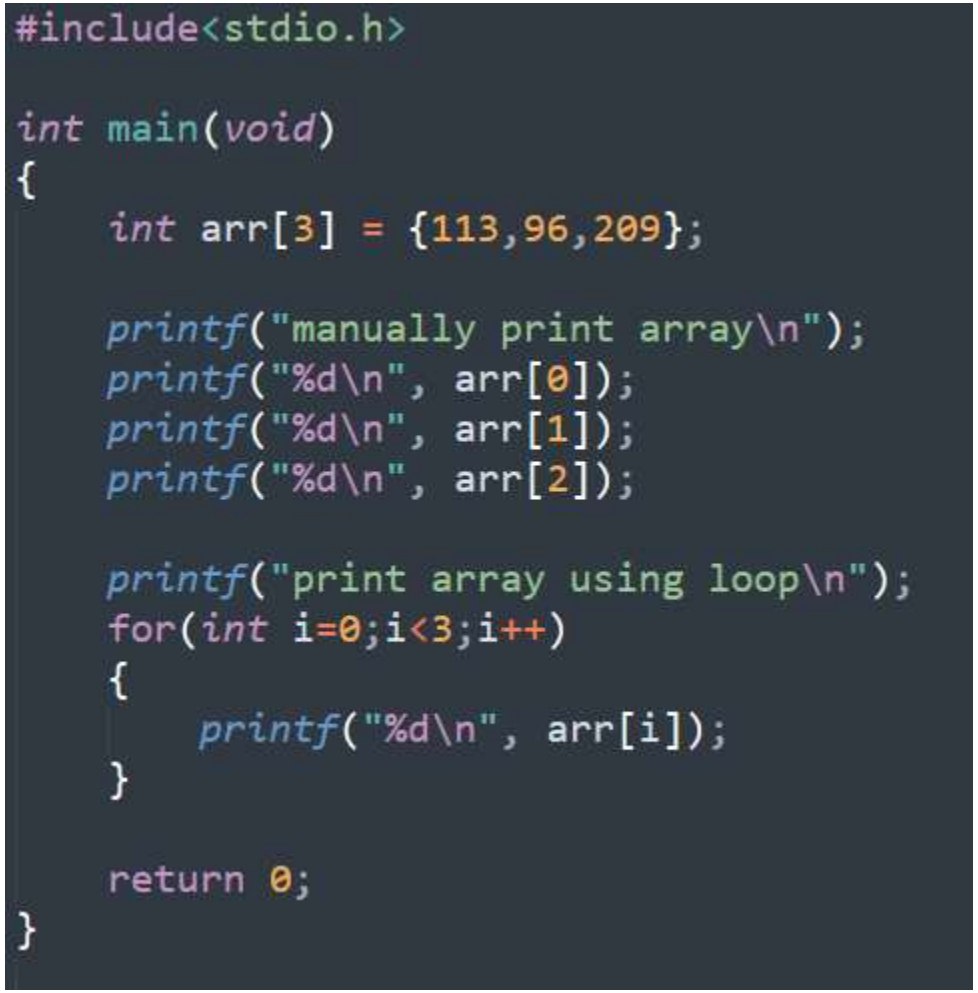
```
type arrayName[length];
```

length=constant(상수)

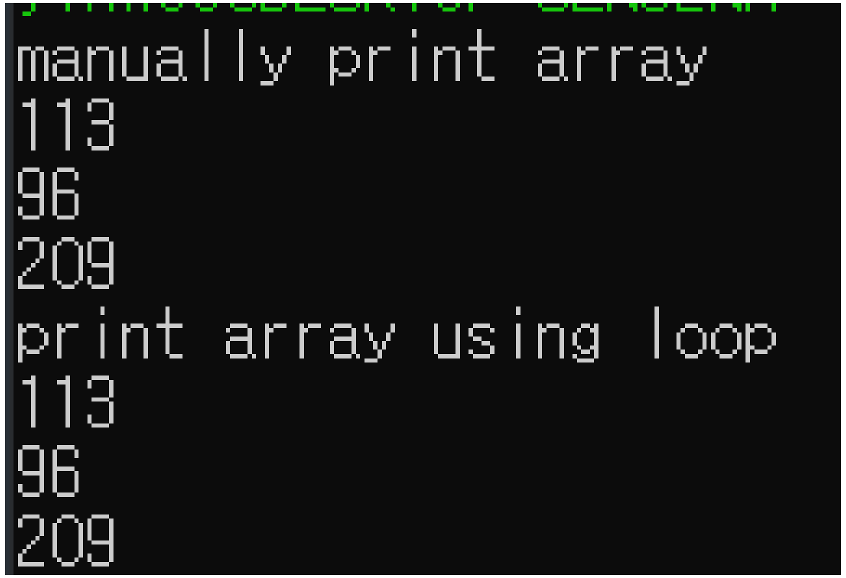
```
int length = 5;

int arrayName[length]; // X
```

```
type arrayName[length] = {element0, element1, element2, …};

type arrayName[] = {element0, element1, element2, …};
```

Array

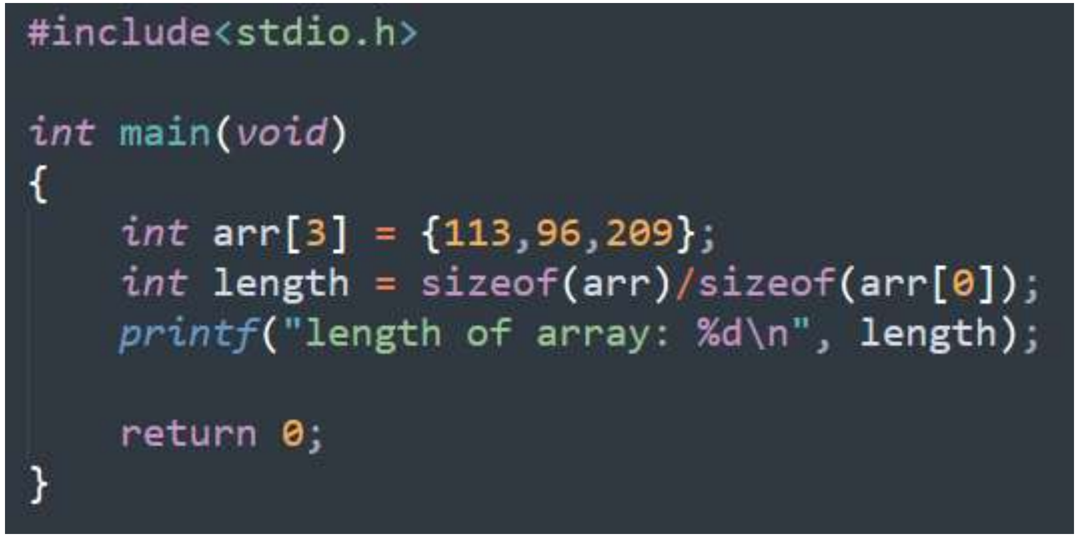


```c
#include<stdio.h>

int main(void)
{
    int arr[3] = {113,96,209};

    printf("manually print array\n");
    printf("%d\n", arr[0]);
    printf("%d\n", arr[1]);
    printf("%d\n", arr[2]);

    printf("print array using loop\n");
    for(int i=0;i<3;i++)
    {
        printf("%d\n", arr[i]);
    }

    return 0;
}
```

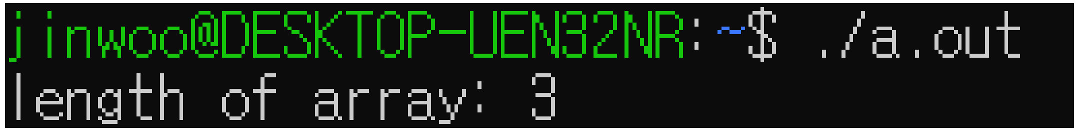


```
manually print array
113
96
209
print array using loop
113
96
209
```

Array

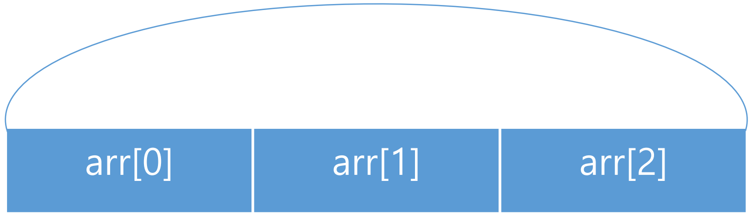


```c
#include<stdio.h>

int main(void)
{
    int arr[3] = {113,96,209};
    int length = sizeof(arr)/sizeof(arr[0]);
    printf("length of array: %d\n", length);

    return 0;
}
```

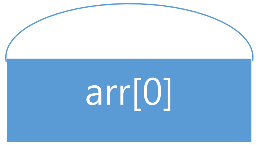


```
jinwoo@DESKTOP-UEN32NR:~$ ./a.out
length of array: 3
```

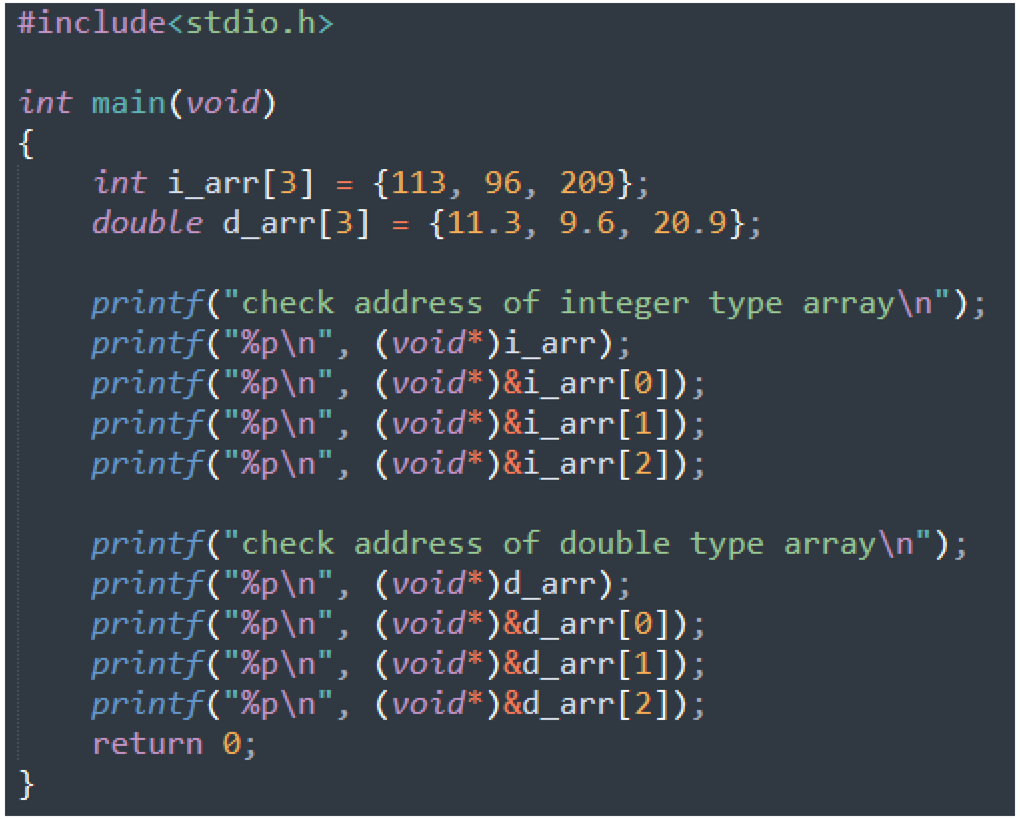
sizeof(arr) = **length * sizeof(int)**



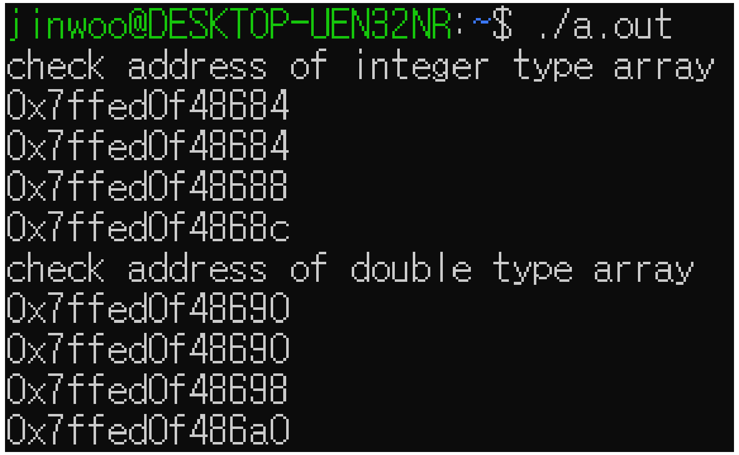
sizeof(arr[0]) = **sizeof(int)**



Array



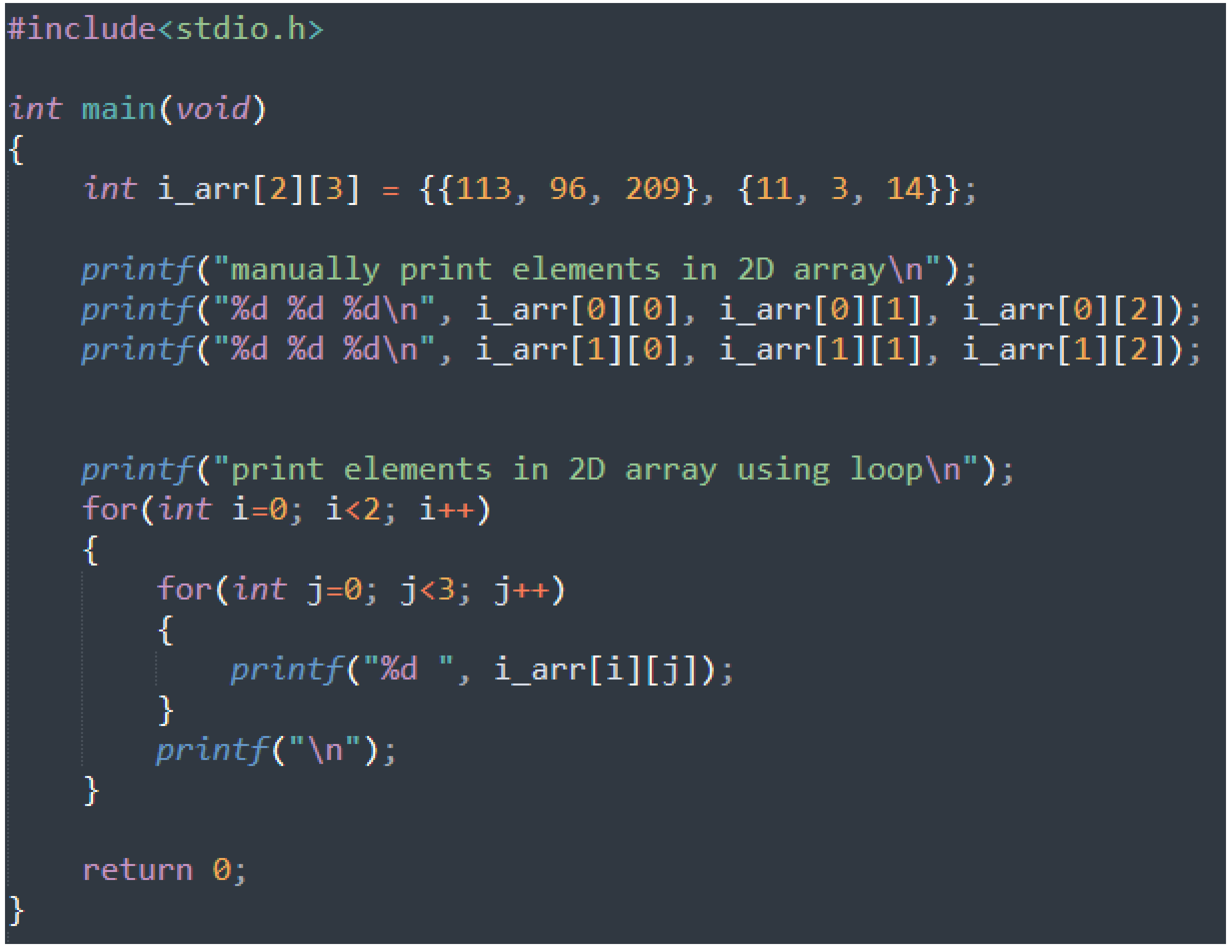
```c
#include<stdio.h>

int main(void)
{
    int i_arr[3] = {113, 96, 209};
    double d_arr[3] = {11.3, 9.6, 20.9};

    printf("check address of integer type array\n");
    printf("%p\n", (void*)i_arr);
    printf("%p\n", (void*)&i_arr[0]);
    printf("%p\n", (void*)&i_arr[1]);
    printf("%p\n", (void*)&i_arr[2]);

    printf("check address of double type array\n");
    printf("%p\n", (void*)d_arr);
    printf("%p\n", (void*)&d_arr[0]);
    printf("%p\n", (void*)&d_arr[1]);
    printf("%p\n", (void*)&d_arr[2]);
    return 0;
}
```



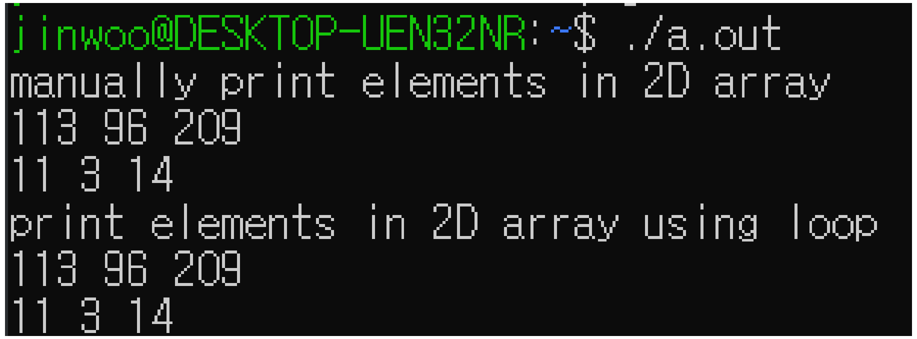
```
jinwoo@DESKTOP-UEN32NR:~$ ./a.out
check address of integer type array
0x7ffed0f48684
0x7ffed0f48684
0x7ffed0f48688
0x7ffed0f4868c
check address of double type array
0x7ffed0f48690
0x7ffed0f48690
0x7ffed0f48698
0x7ffed0f486a0
```

HEX	84	HEX	88	HEX	8C
DEC	132	DEC	136	DEC	140
HEX	90	HEX	98	HEX	A0
DEC	144	DEC	152	DEC	160

2D Array

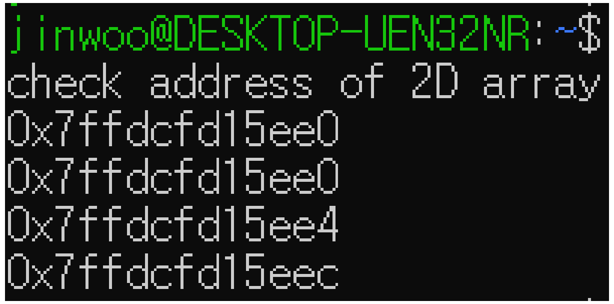


```c
#include<stdio.h>

int main(void)
{
    int i_arr[2][3] = {{113, 96, 209}, {11, 3, 14}};

    printf("manually print elements in 2D array\n");
    printf("%d %d %d\n", i_arr[0][0], i_arr[0][1], i_arr[0][2]);
    printf("%d %d %d\n", i_arr[1][0], i_arr[1][1], i_arr[1][2]);


    printf("print elements in 2D array using loop\n");
    for(int i=0; i<2; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("%d ", i_arr[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```



```
jinwoo@DESKTOP-UEN32NR:~$ ./a.out
manually print elements in 2D array
113 96 209
11 3 14
print elements in 2D array using loop
113 96 209
11 3 14
```

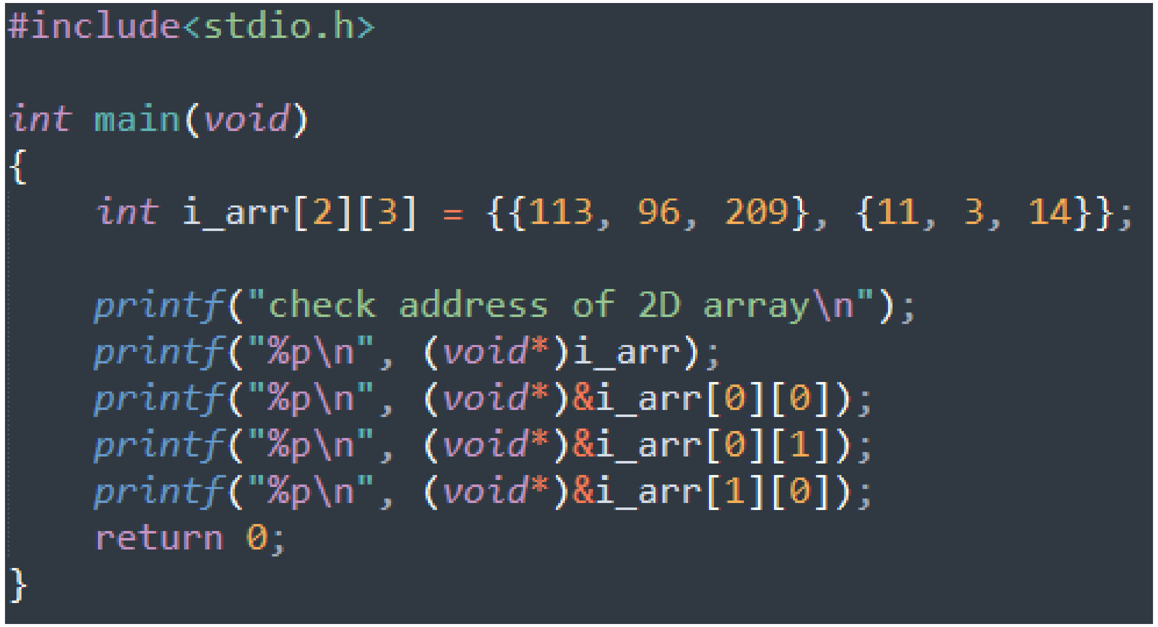
i_arr[0][0]	i_arr[0][1]	i_arr[0][2]
i_arr[1][0]	i_arr[1][1]	i_arr[1][2]



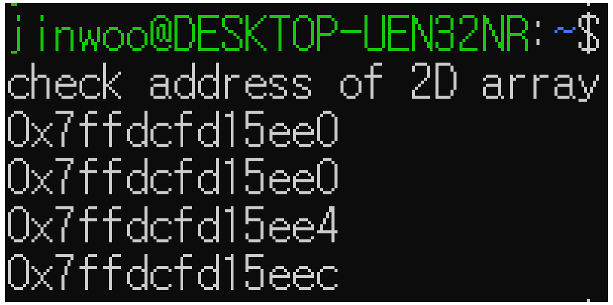
```
jinwoo@DESKTOP-UEN32NR:~$
check address of 2D array
0x7ffdcfd15ee0
0x7ffdcfd15ee0
0x7ffdcfd15ee4
0x7ffdcfd15eec
```

i_arr[0][0]	i_arr[0][1]	i_arr[0][2]	i_arr[1][0]	i_arr[1][1]	i_arr[1][2]

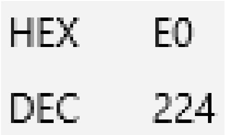
2DArray



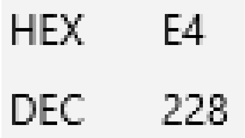
```c
#include<stdio.h>

int main(void)
{
    int i_arr[2][3] = {{113, 96, 209}, {11, 3, 14}};

    printf("check address of 2D array\n");
    printf("%p\n", (void*)i_arr);
    printf("%p\n", (void*)&i_arr[0][0]);
    printf("%p\n", (void*)&i_arr[0][1]);
    printf("%p\n", (void*)&i_arr[1][0]);
    return 0;
}
```



```
jinwoo@DESKTOP-UEN32NR:~$
check address of 2D array
0x7ffdcfd15ee0
0x7ffdcfd15ee0
0x7ffdcfd15ee4
0x7ffdcfd15eec
```

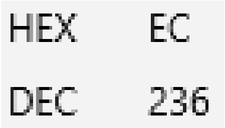


HEX	E0
DEC	224

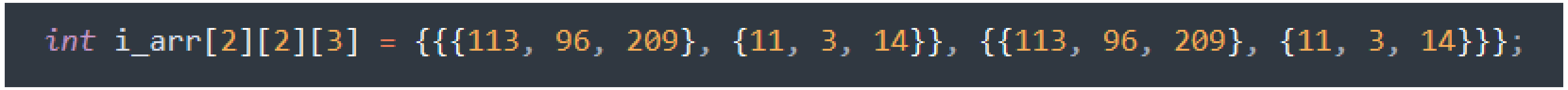


HEX	E4
DEC	228

i_arr[0][0]	i_arr[0][1]	i_arr[0][2]
i_arr[1][0]	i_arr[1][1]	i_arr[1][2]



HEX	EC
DEC	236



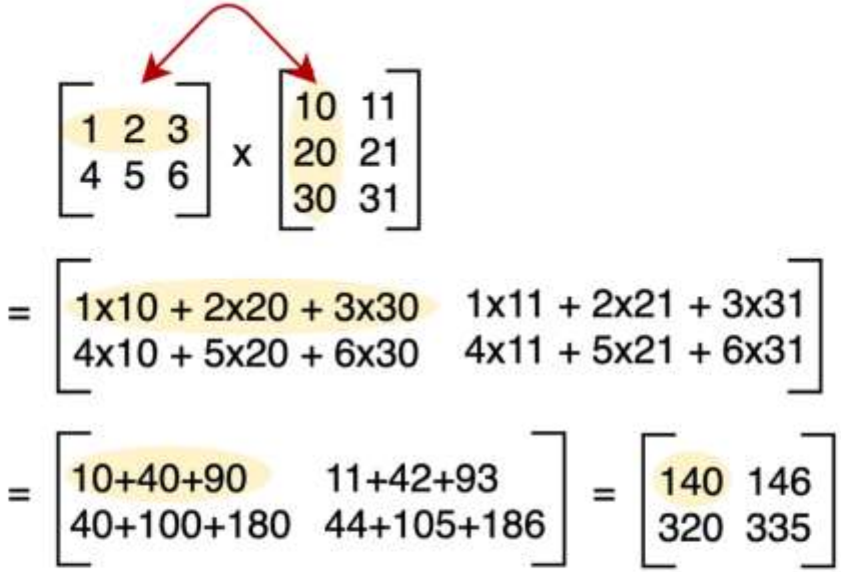
```c
int i_arr[2][2][3] = {{{113, 96, 209}, {11, 3, 14}}, {{113, 96, 209}, {11, 3, 14}}};
```

LAB – FindMaxNum

- Create a file named 'FindMaxNum_YourName.c'.
- Your program should
    - prompt the user to enter 10 integers.
    - Store the numbers in an array
    - Find the maximum number from the array
    - Display the maximum number to the user.
- Use a loop to input and process the numbers.
- Display the maximum number after all inputs have been entered.

LAB – MatrixX

- Create a file named 'MatrixX_YourName.c'.
- Your program should
  - prompt the user to enter values for 2x3 and 3x2 matrices
  - Store the numbers in two separate arrays.
  - Multiply the two matrices
  - Store the result in a third 2x2 array
- Display the resulting matrix to the user.
  - output should have same form as a matrix



$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{x} \begin{bmatrix} 10 & 11 \\ 20 & 21 \\ 30 & 31 \end{bmatrix}$$

$$= \begin{bmatrix} 1\text{x}10 + 2\text{x}20 + 3\text{x}30 & 1\text{x}11 + 2\text{x}21 + 3\text{x}31 \\ 4\text{x}10 + 5\text{x}20 + 6\text{x}30 & 4\text{x}11 + 5\text{x}21 + 6\text{x}31 \end{bmatrix}$$

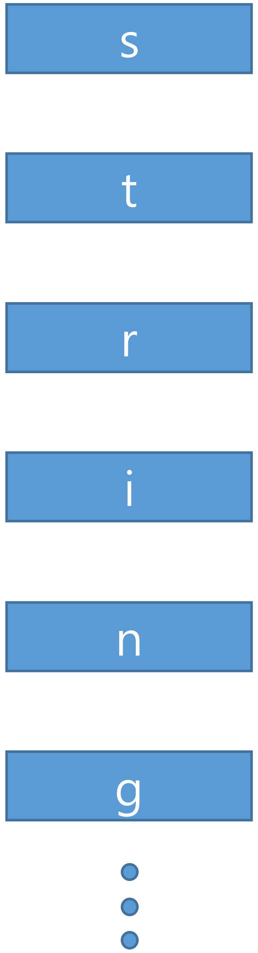
$$= \begin{bmatrix} 10+40+90 & 11+42+93 \\ 40+100+180 & 44+105+186 \end{bmatrix} = \begin{bmatrix} 140 & 146 \\ 320 & 335 \end{bmatrix}$$

String

DIGIPEN
DRAGONS

Jinwoo Choi

String(문자열)이란?


s
t
r
i
n
g

```
type arrayName[length];
char stringName[length];
```

length=constant(상수)

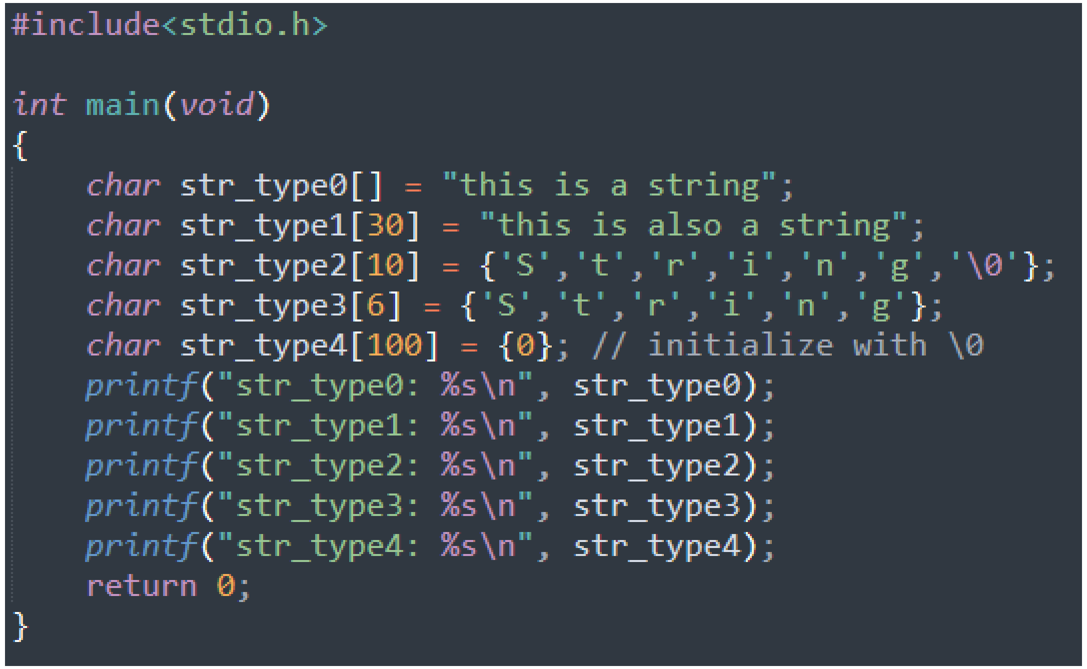
s	t	r	i	n	g

String(문자열)이란?

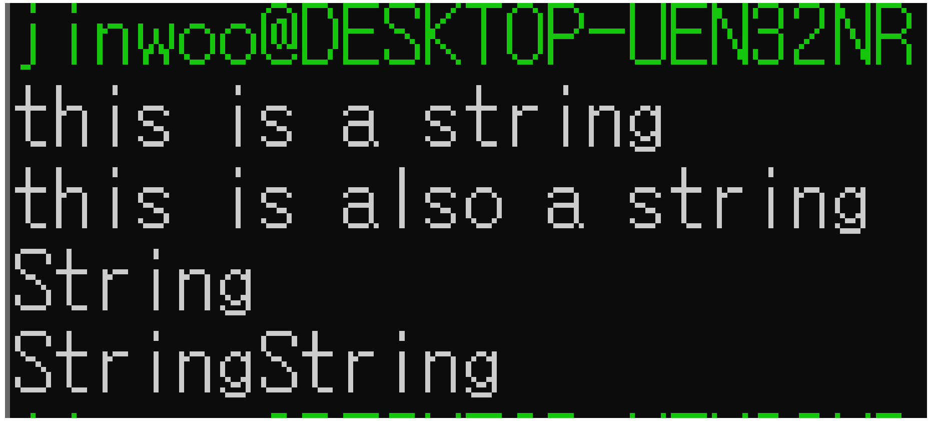
```
char str[10] = “string”;
```

s	t	r	i	n	g	\0	?	?	?

String - output

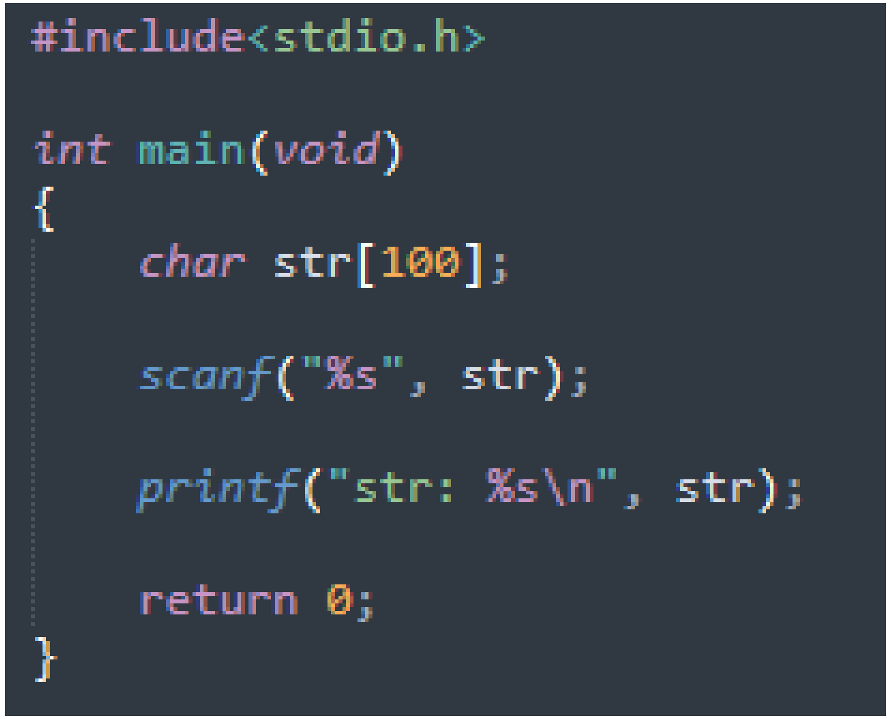


```c
#include<stdio.h>

int main(void)
{
    char str_type0[] = "this is a string";
    char str_type1[30] = "this is also a string";
    char str_type2[10] = {'S','t','r','i','n','g','\0'};
    char str_type3[6] = {'S','t','r','i','n','g'};
    char str_type4[100] = {0}; // initialize with \0
    printf("str_type0: %s\n", str_type0);
    printf("str_type1: %s\n", str_type1);
    printf("str_type2: %s\n", str_type2);
    printf("str_type3: %s\n", str_type3);
    printf("str_type4: %s\n", str_type4);
    return 0;
}
```

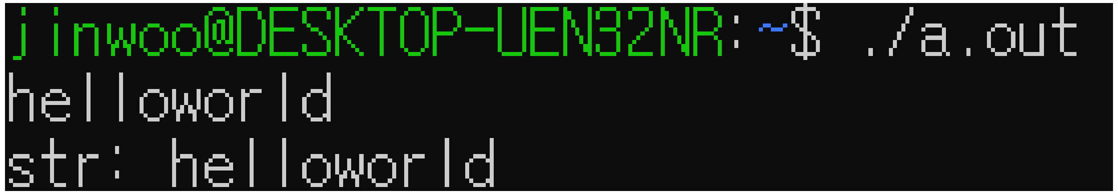


```
jinwoo@DESKTOP-UEN32NR
this is a string
this is also a string
String
StringString
```

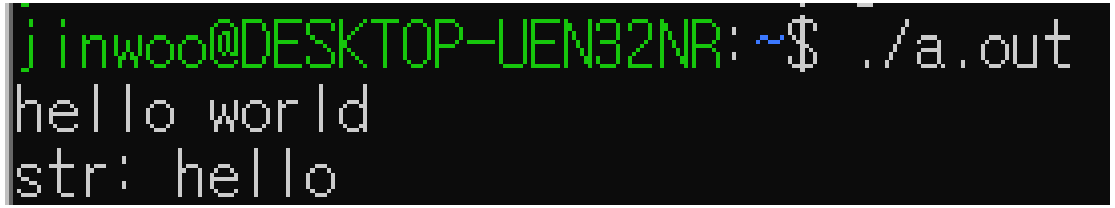
String – input(scanf)



```c
#include<stdio.h>

int main(void)
{
    char str[100];

    scanf("%s", str);

    printf("str: %s\n", str);

    return 0;
}
```

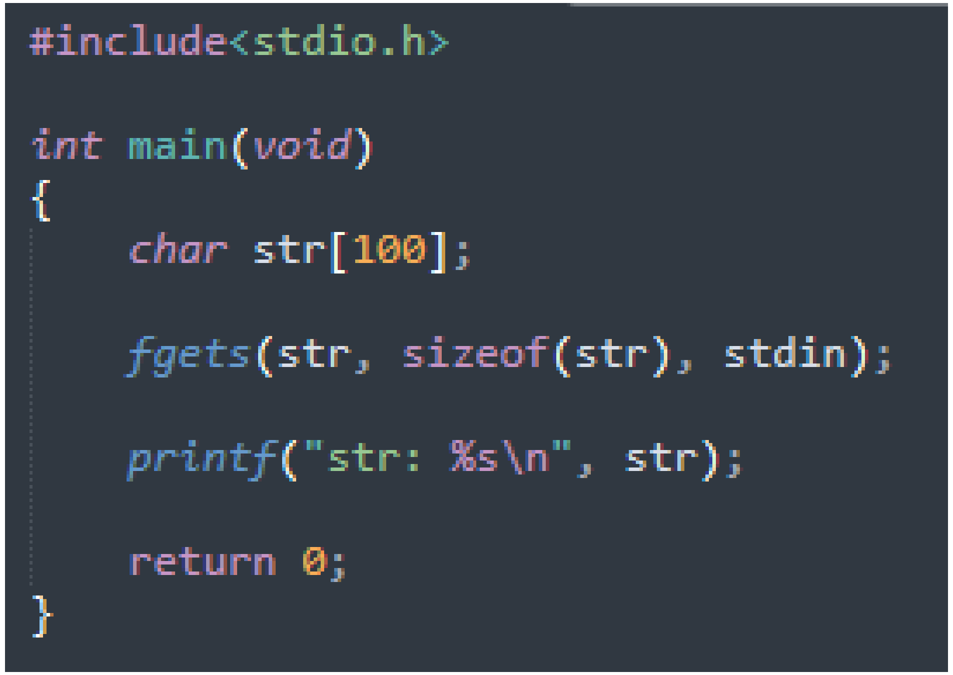


```
jinwoo@DESKTOP-UEN32NR:~$ ./a.out
helloworld
str: helloworld
```

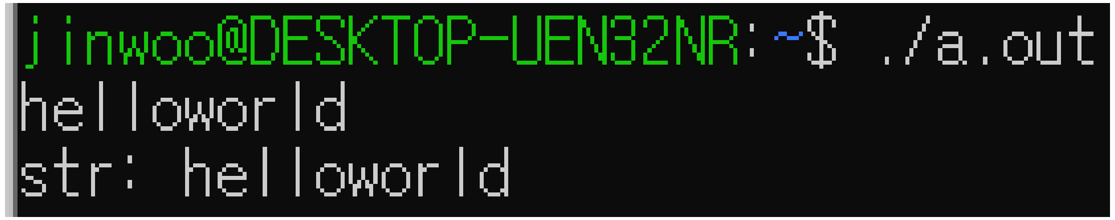


```
jinwoo@DESKTOP-UEN32NR:~$ ./a.out
hello world
str: hello
```

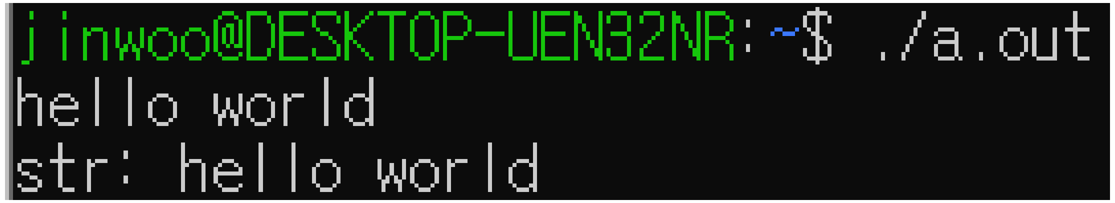
String – input(fgets)



```c
#include<stdio.h>

int main(void)
{
    char str[100];

    fgets(str, sizeof(str), stdin);

    printf("str: %s\n", str);

    return 0;
}
```



```
jinwoo@DESKTOP-UEN32NR:~$ ./a.out
helloworld
str: helloworld
```

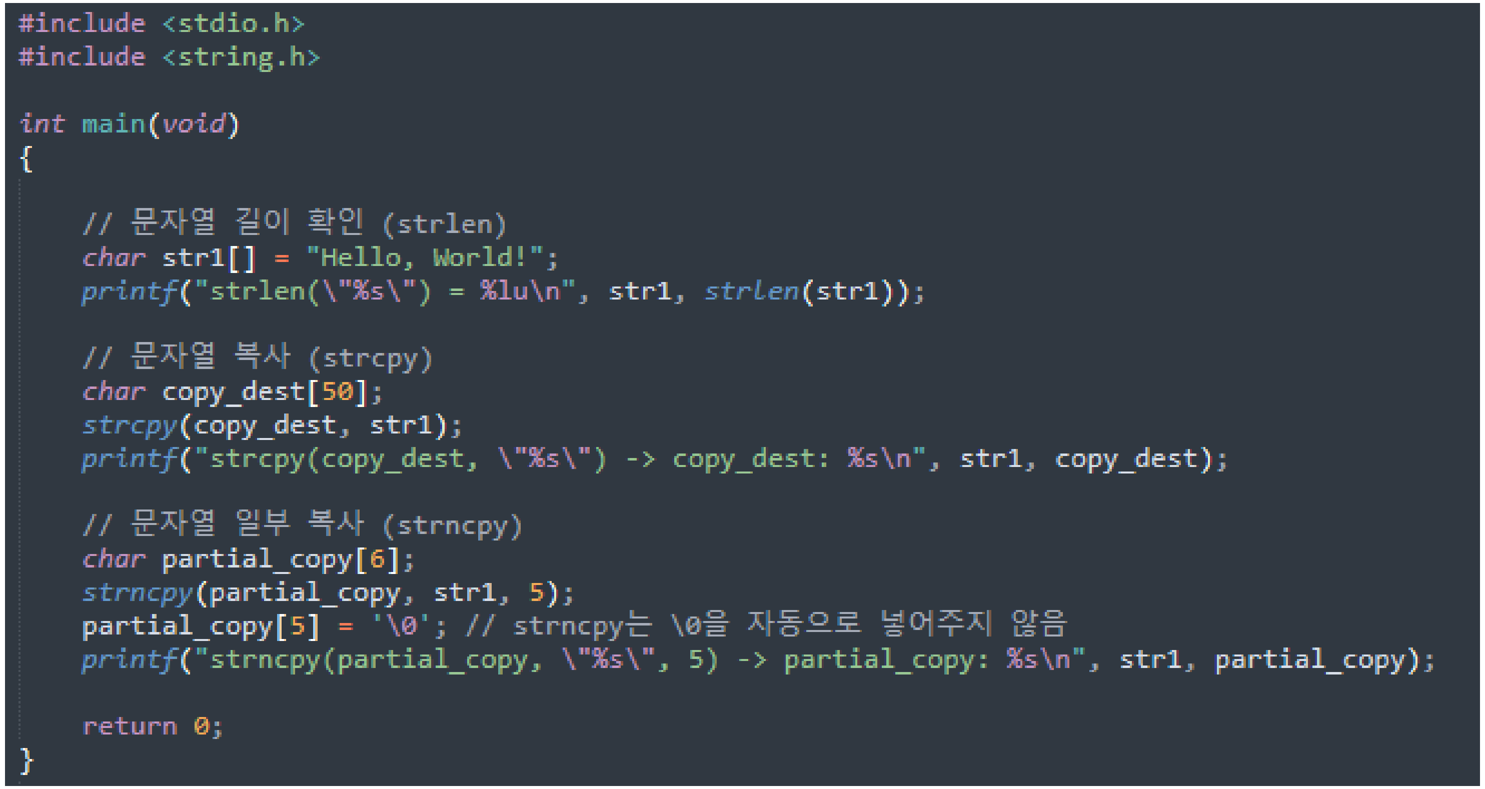


```
jinwoo@DESKTOP-UEN32NR:~$ ./a.out
hello world
str: hello world
```

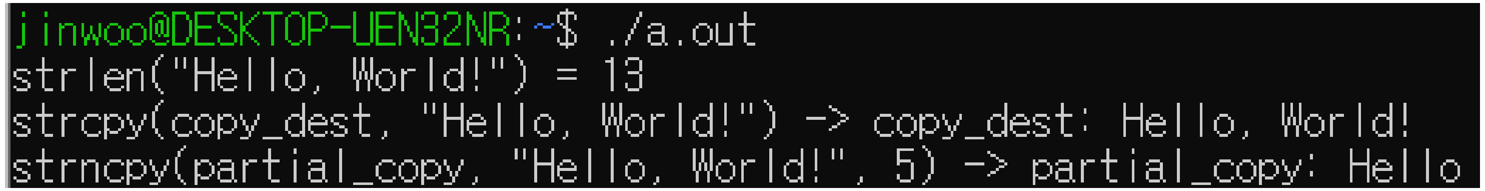
<string.h>

https://www.geeksforgeeks.org/c-library-string-h/

- **strlen() = string Length**
- **strcpy() = string Copy**
- **strncpy() = string n개 Copy**



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

int main(void)
{

    // 문자열 길이 확인 (strlen)
    char str1[] = "Hello, World!";
    printf("strlen(\"%s\") = %lu\n", str1, strlen(str1));

    // 문자열 복사 (strcpy)
    char copy_dest[50];
    strcpy(copy_dest, str1);
    printf("strcpy(copy_dest, \"%s\") -> copy_dest: %s\n", str1, copy_dest);

    // 문자열 일부 복사 (strncpy)
    char partial_copy[6];
    strncpy(partial_copy, str1, 5);
    partial_copy[5] = '\0'; // strncpy는 \0을 자동으로 넣어주지 않음
    printf("strncpy(partial_copy, \"%s\", 5) -> partial_copy: %s\n", str1, partial_copy);

    return 0;
}
```

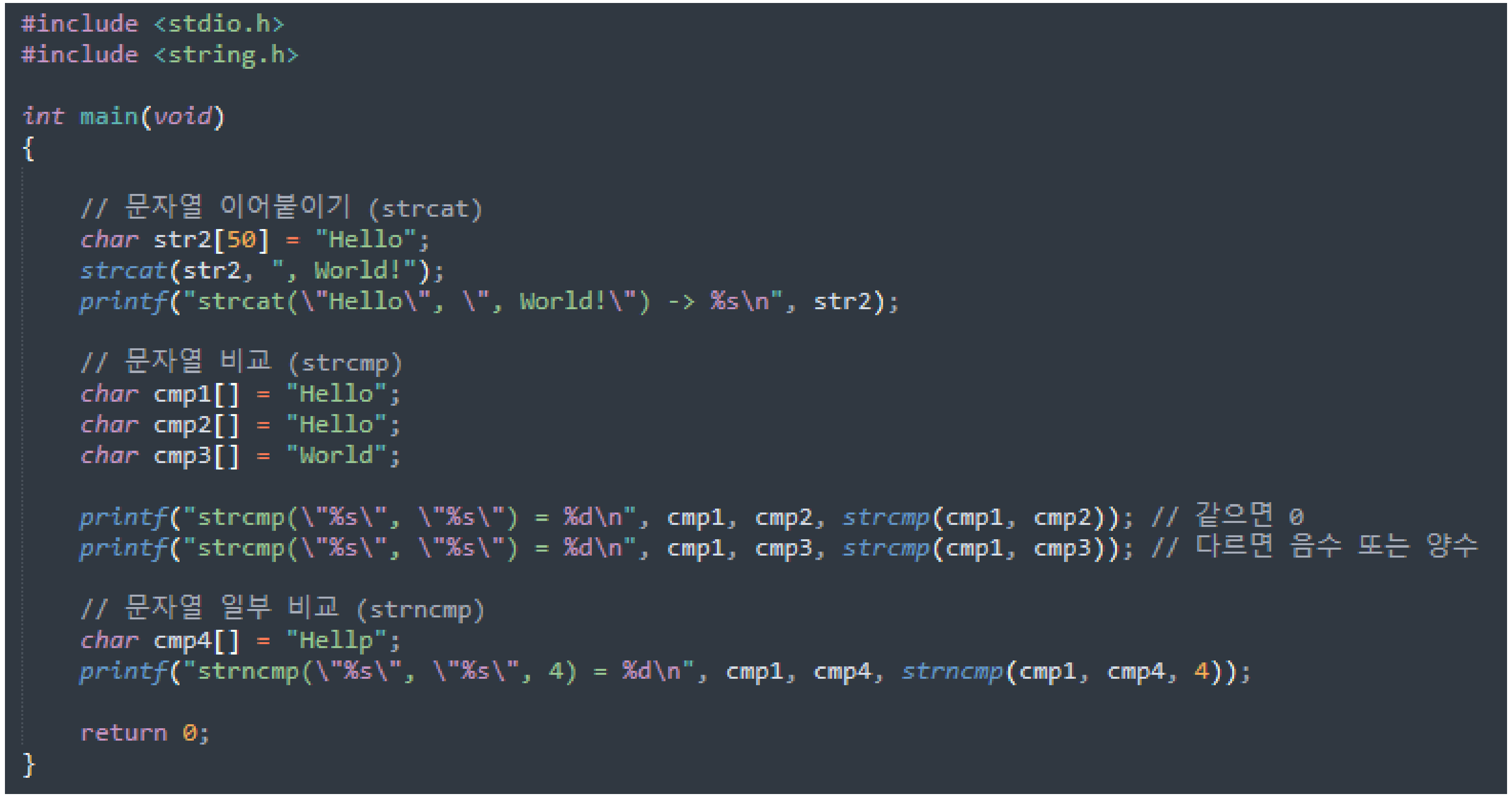


```
jinwoo@DESKTOP-UEN32NR:~$ ./a.out
strlen("Hello, World!") = 13
strcpy(copy_dest, "Hello, World!") -> copy_dest: Hello, World!
strncpy(partial_copy, "Hello, World!", 5) -> partial_copy: Hello
```

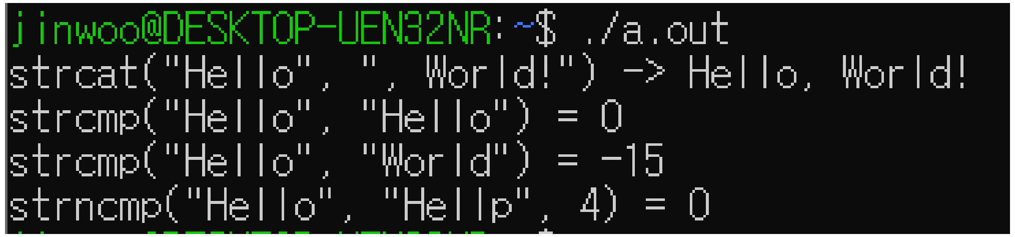
<string.h>

https://www.geeksforgeeks.org/c-library-string-h/

- **strcat() = string Concatenates**
- **strcmp() = string Compare**
- **strncmp() = string n개 Compare**



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

int main(void)
{

    // 문자열 이어붙이기 (strcat)
    char str2[50] = "Hello";
    strcat(str2, ", World!");
    printf("strcat(\"Hello\", \", World!\") -> %s\n", str2);

    // 문자열 비교 (strcmp)
    char cmp1[] = "Hello";
    char cmp2[] = "Hello";
    char cmp3[] = "World";

    printf("strcmp(\"%s\", \"%s\") = %d\n", cmp1, cmp2, strcmp(cmp1, cmp2)); // 같으면 0
    printf("strcmp(\"%s\", \"%s\") = %d\n", cmp1, cmp3, strcmp(cmp1, cmp3)); // 다르면 음수 또는 양수

    // 문자열 일부 비교 (strncmp)
    char cmp4[] = "Hellp";
    printf("strncmp(\"%s\", \"%s\", 4) = %d\n", cmp1, cmp4, strncmp(cmp1, cmp4, 4));

    return 0;
}
```



```
jinwoo@DESKTOP-UEN32NR:~$ ./a.out
strcat("Hello", ", World!") -> Hello, World!
strcmp("Hello", "Hello") = 0
strcmp("Hello", "World") = -15
strncmp("Hello", "Hellp", 4) = 0
```

LAB – String

- Create a file named 'String_YourName.c'.
- Your program should
  - Prompt the user for input and store it in a char array.
  - Declare and manipulate strings using <string.h> functions.
    - Try multiple functions.