


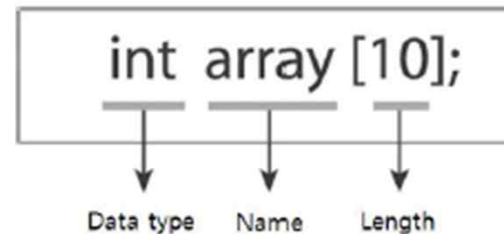
| Basic and Practice in Programming

LAB 6



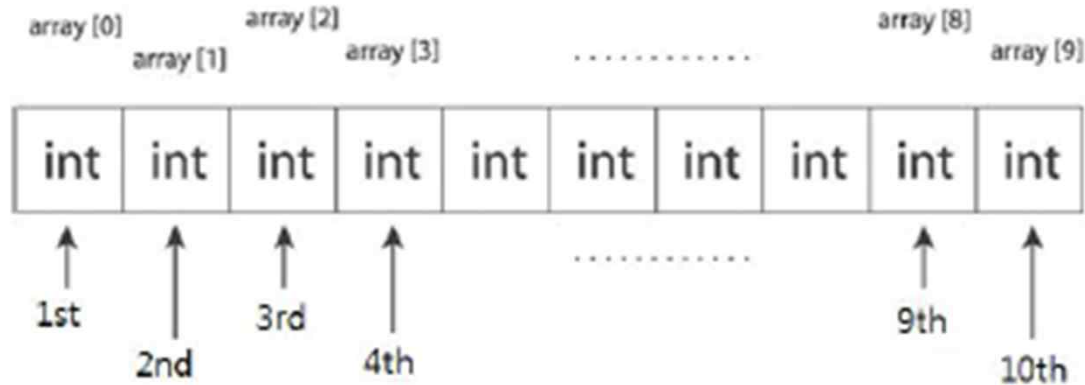
Array basic

- Declare more than one variable which are the same type
- Three elements in declaration of array
 - Length
 - Data type
 - Name



Array basic

- Access 1-dimension array
 - By index : the position of a element on array
 - Index starts with 0



Array basic

- Initialization with declaration

```
int main(void)
{
    int arr1[5] = {1, 2, 3, 4, 5};
    int arr2[ ] = {1, 3, 5, 7, 9};
    int arr3[5] = {1, 2};
    int arr4[ ];
}
```

Diagram illustrating array initialization:

- `int arr1[5] = {1, 2, 3, 4, 5};` →

1	2	3	4	5
---	---	---	---	---
- `int arr2[] = {1, 3, 5, 7, 9};` →

1	3	5	7	9
---	---	---	---	---
- `int arr3[5] = {1, 2};` →

1	2	0	0	0
---	---	---	---	---
- `int arr4[];` → **Error!**

Array with string

- We can store a string using array

```
/* LAB6 practice 1 : Array 1*/  
#include <stdio.h>  
  
int main (void)  
{  
    char str1[5] = "Good";  
    char str2[ ] = "morning";  
  
    printf("%s \n", str1);  
    printf("%s %s \n", str1, str2);  
  
    return 0;  
}
```

Array with string

- End of the string, Null character('\0') is stored

```
/* LAB6 practice 2*/  
#include <stdio.h>  
  
int main (void)  
{  
    char str2[ ] = "morning";  
  
    printf("%ld \n", sizeof(str2));  
  
    return 0;  
}
```

Array with string

- Assign value using scanf

```
/* LAB6 practice 3 */
#include <stdio.h>

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

    printf("input : ");
    scanf("%s", str);

    printf("output : %s \n", str);
    return 0;
}
```

Result

input : computer programming
output : ?

Exercise 1

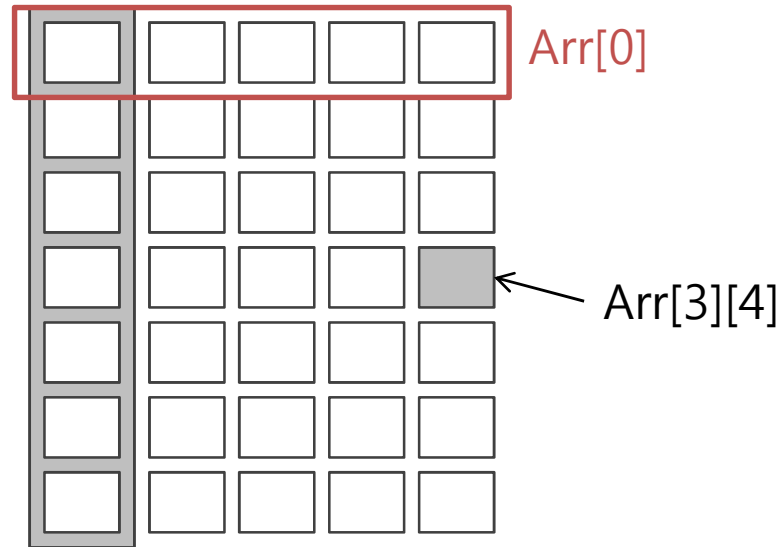
- Reverse the string
 - Input : Length n string ($n < 20$)
 - Output : Reversed string

Ex) input : Semiconductor
output : rotcudnocimeS

2D Array (1/4)

- An array can have arrays as own component

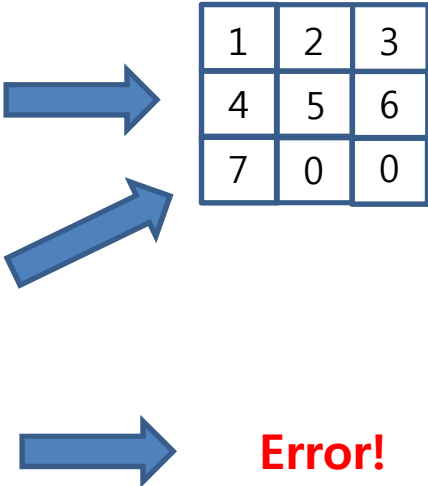
```
int Arr[7][5];
```



2D Array (2/4)

- Initialization with declaration

```
int main(void)
{
    int arr1[3][3] = {1, 2, 3, 4, 5, 6, 7};
    int arr2[ ][3] = {1, 2, 3, 4, 5, 6, 7};
    int arr3[ ][ ] = {1, 2, 3, 4, 5, 6, 7};
}
```



1	2	3
4	5	6
7	0	0

Error!

2D Array (2/4)

```
/* LAB6 practice 4 : 2D array */
int main(void)
{
    int arr[5][5];
    int i, j;

    for (i = 0; i < 5; i++)
        for (j = 0; j < 5; j++)
            arr[i][j] = i * j;

    for (i = 0; i < 5; i++) {
        for (j = 0; j < 5; j++)
            printf("%d ", arr[i][j]);
        printf("\n");
    }
    return 0;
}
```

2D Array (3/4)

```
/* LAB6 practice 5 : 2D array 2 */
int main(void)
{
    char arr[5][5] = {"apple", "pies\n", "array", "point", "er\n.\n0"};
    int i, j;

    printf("loop1 \n");

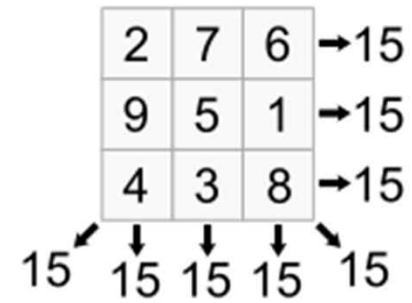
    for (i = 0; i < 5; i++){
        for (j = 0; j < 5; j++){
            printf(" %c", arr[i][j]);
            printf("\n");
        }
    }
    printf("loop2 \n");

    for (i = 0; i < 5; i++)
        printf("%d:%s\n", i, arr[i]);

    return 0;
}
```

Exercise 2

- Identifying Magic Square
 - Input : 5 x 5 integer matrix
 - Output : yes/no



2	7	6
9	5	1
4	3	8



yes

2	5	6
9	7	1
4	3	8



no