



| Basic and Practice in Programming

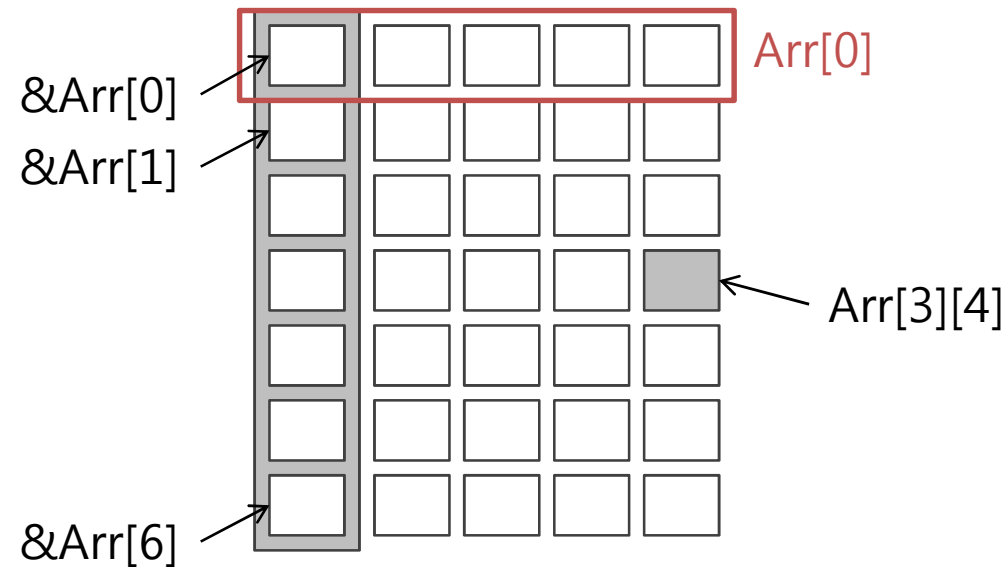
Week 10



2D Array (1/3)

- An array can have arrays as own component

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



2D Array (2/3)

```
/*Practice 1 : 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 (3/3)

```
/* Practice 2 : 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]);

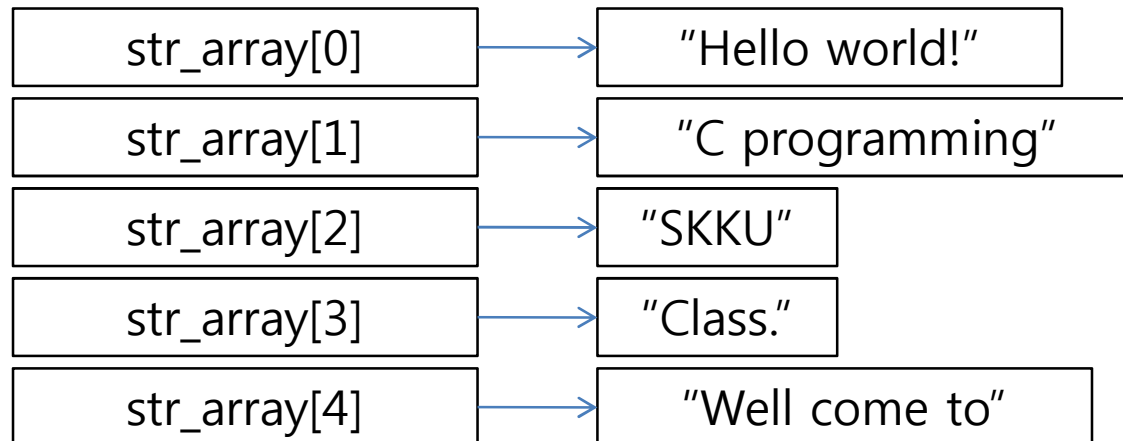
    return 0;
}
```

Pointer Array (1/3)

- Pointer can be declared as array

```
char *str_array[5]
```

- Each element of array is independent pointer



Pointer Array (2/3)

```
/*Practice 3 : Pointer array */
int main(void)
{
    char *str_array[5];
    int i;

    str_array[0] = "hello world";
    str_array[1] = "C programming";
    str_array[2] = "SKKU";
    str_array[3] = "Class";
    str_array[4] = "Well come to";

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

    return 0;
}
```

Pointer Array (3/3)

```
/*Practice 4 : Pointer array wrong use */
int main(void)
{
    char *str_array[5];
    int i;

    str_array[0] = "hello world";
    str_array[1] = "C programming";
    str_array[2] = "SKKU";
    str_array[3] = "Class";
    str_array[4] = "Well come to";

    str_array[0][0] = 'H';

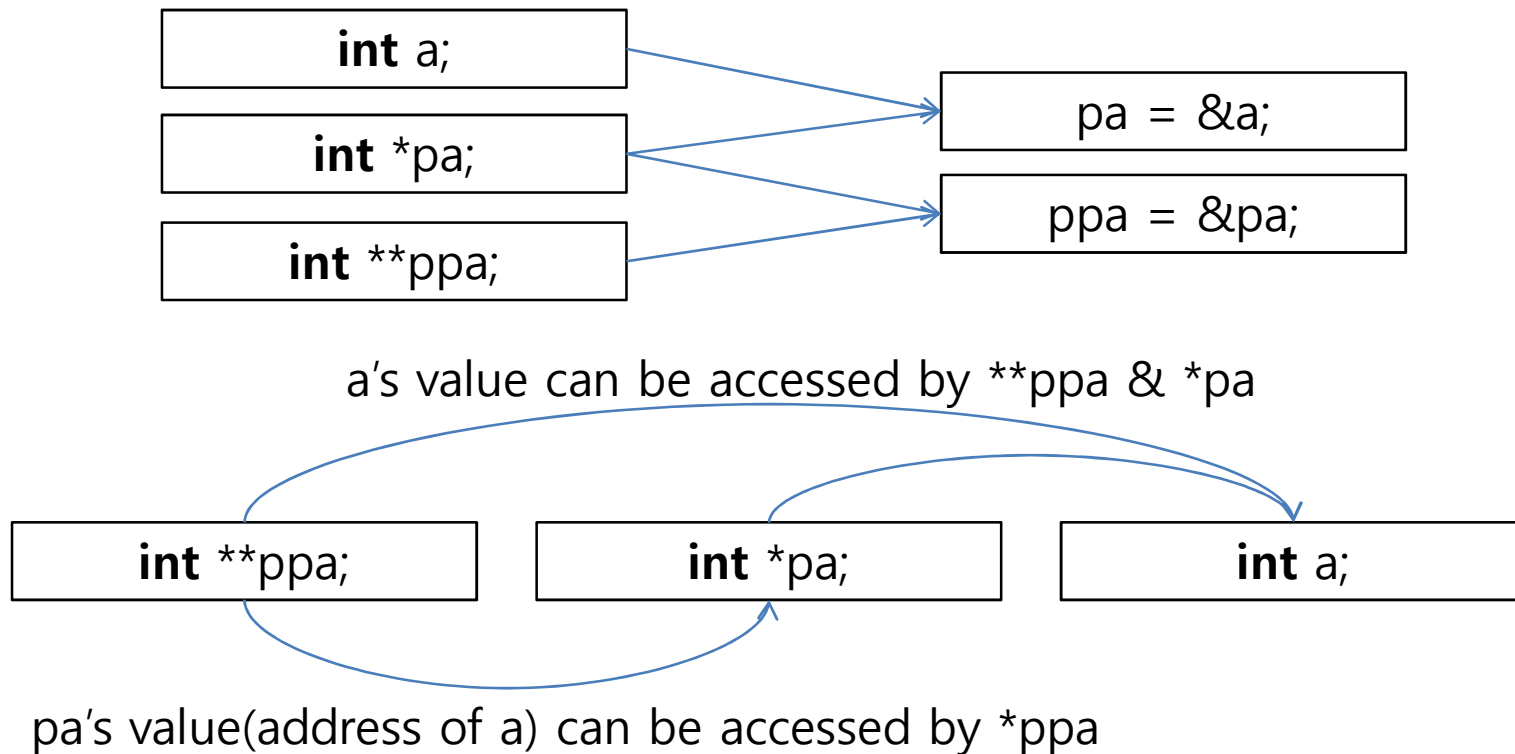
    for (i = 0;i < 5;i++)
        printf("%s\n", str_array[i]);

    return 0;
}
```

Stored at "Read-Only" region
Can't be modified

Double Pointer (1/4)

- Pointer can be pointed by pointer



Double Pointer (2/4)

```
/*Practice 5 : Double pointer */
int main(void)
{
    int a;
    int *pa, **ppa;

    a = 10;
    pa = &a;
    ppa = &pa;
    printf("%d %d %d\n", a, *pa, **ppa);

    *pa = 20;
    printf("%d %d %d\n", a, *pa, **ppa);

    **ppa = 30;
    printf("%d %d %d\n", a, pa, ppa);

    return 0;
}
```

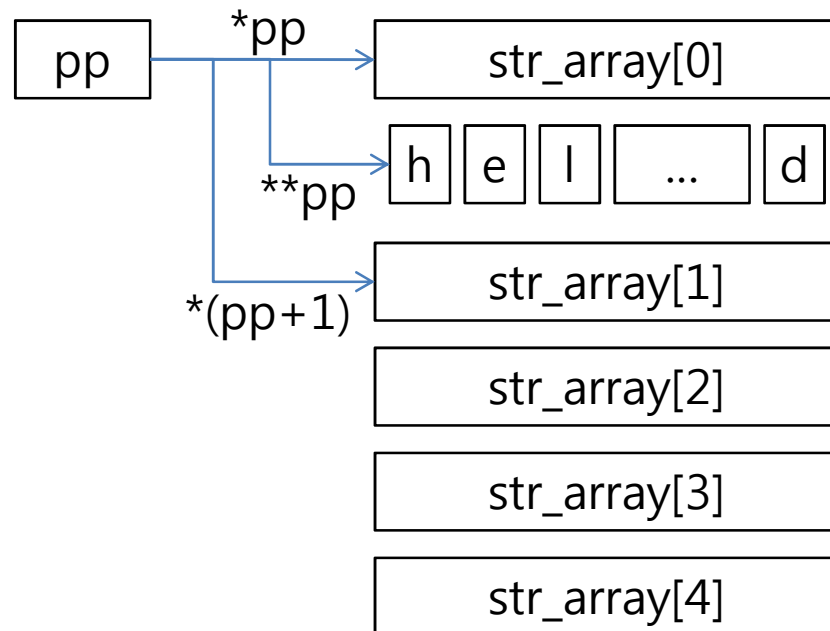
Double Pointer (3/4)

```
/*Practice 6 : Double pointer */
int main(void)
{
    char *str_array[5];
    char **p;
    int i;

    str_array[0] = "hello world";
    str_array[1] = "C programming";
    str_array[2] = "SKKU";
    str_array[3] = "Class";
    str_array[4] = "Well come to";
    p = str_array;

    for (i = 0;i < 5;i++)
        printf("%s\n", *p++);

    return 0;
}
```



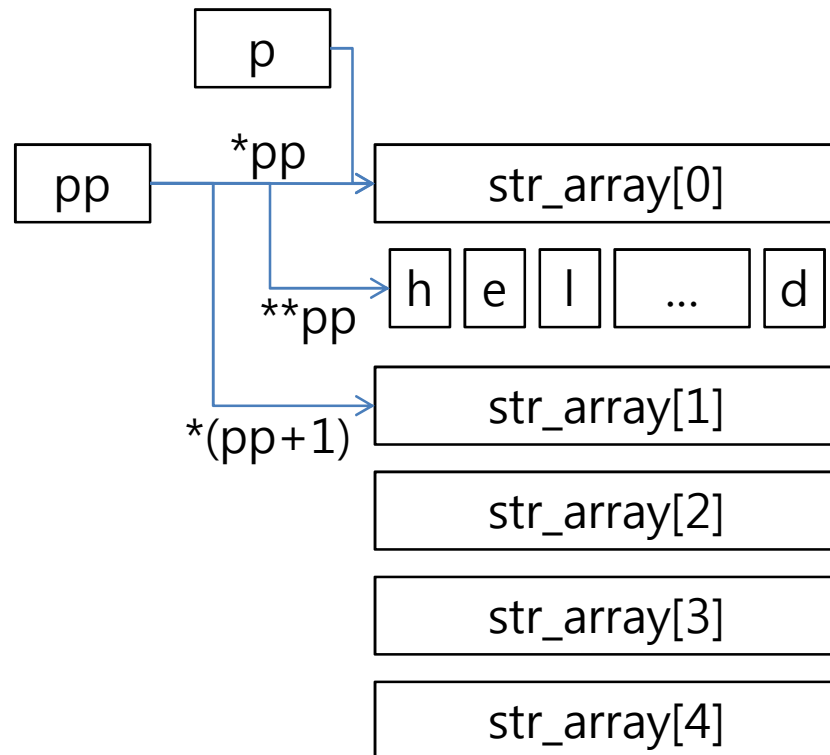
Double Pointer (4/5)

```
/*Practice 7 : Double pointer */
int main(void)
{
    char *str_array[5], **pp, *p;
    int i;

    str_array[0] = "hello world";
    str_array[1] = "C programming";
    str_array[2] = "SKKU";
    str_array[3] = "Class";
    str_array[4] = "Well come to";
    pp = str_array;
    p = *pp;

    for (i = 0; i < 5; i++)
        printf("%c\n", *p++);
    for (i = 0; i < 5; i++)
        printf("%s\n", *pp++);

    return 0;
}
```



Double Pointer (5/5)

```
/*Practice 8 : Double pointer practice*/
```

```
void swap (char **str1, char **str2)
{
    char *temp;

    temp = *str1;
    *str1 = *str2;
    *str2 = temp;
}
```

```
int main(void)
{
    char *str_array[5];
    int i;

    str_array[0] = "hello world";
    str_array[1] = "C programming";
    str_array[2] = "SKKU";
    str_array[3] = "Class";
    str_array[4] = "Well come to";

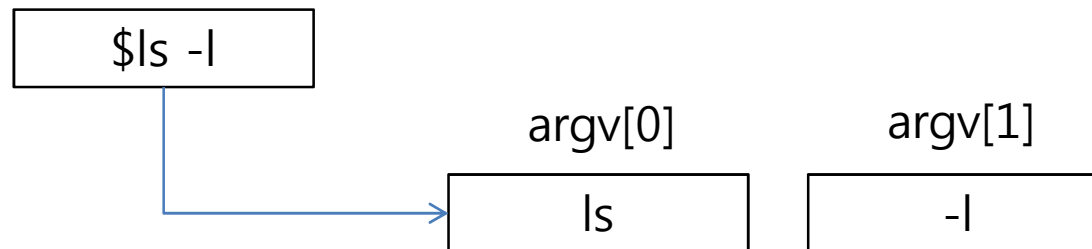
    swap(&str_array[4], &str_array[0]);
    swap(&str_array[1], &str_array[2]);

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

    return 0;
}
```

Command-line Argument

- C-language supports command-line args
 - You can give a number of arguments to program



- All the program has at least one argument that is program's name

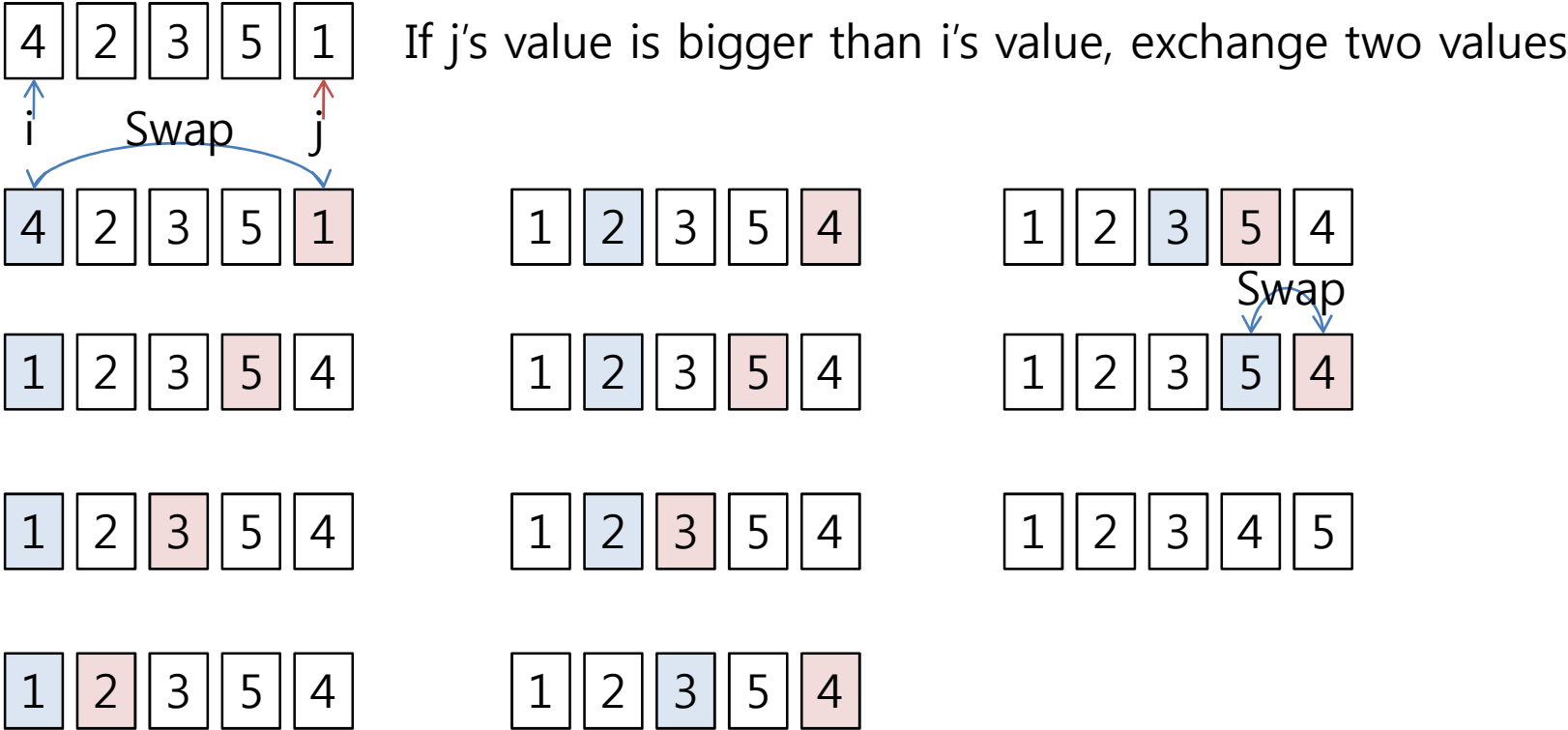
Command-line Argument

```
/*Practice 9 : Command line args*/  
  
int main (int argc, char *argv[])  
{  
    int i;  
  
    if (argc >= 1) {  
        for (i = 0; i < argc; i++)  
            printf("%d's argument : %s\n", i, argv[i]);  
    }  
  
    return 0;  
}
```

```
$a.out test 1 2 3
```

Pointer Practice – Bubble Sort

- Bubble sort
 - The most simple sorting algorithm



Pointer Practice – Bubble Sort

```
/*Practice 10 : Command line args*/
int main(void)
{
    char *str_array[5];
    int i, j;

    str_array[0] = "Hello world";
    str_array[1] = "C programming";
    str_array[2] = "SKKU";
    str_array[3] = "Class";
    str_array[4] = "Well come to";

    for (i = 0; i < 5; i++)
        for (j = 4; j > i; j--)
            if (str_array[i][0] > str_array[j][0])
                swap(&str_array[i], &str_array[j]);

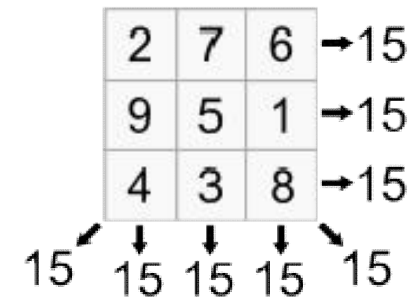
    for (i = 0; i < 5; i++)
        printf("%s\n", str_array[i]);
}
```

Using practice8's swap function



Exercise

- Identifying Magic Square (Due date : Today's 11:59 PM)
 - 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