

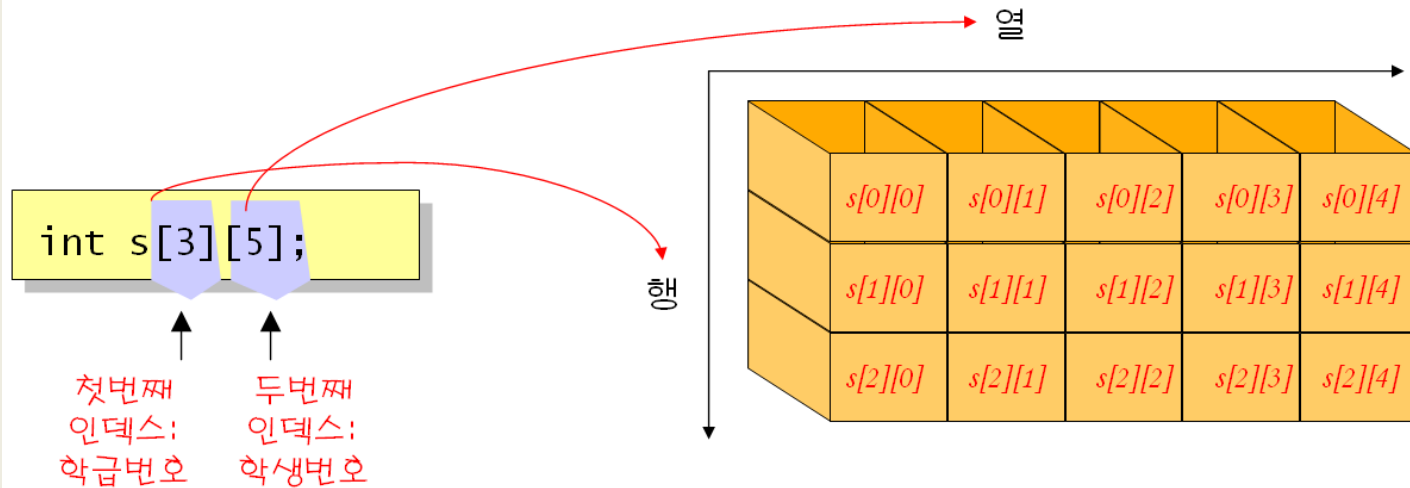


다차원 배열



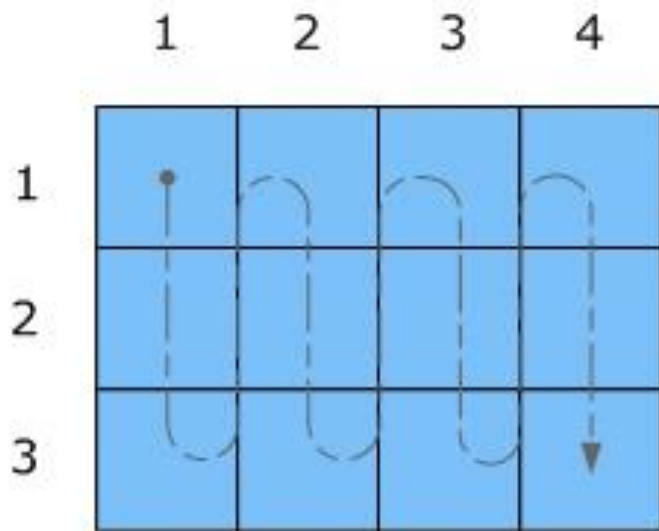
2차원 배열

```
int s[10];      // 1차원 배열  
int s[3][10];   // 2차원 배열  
int s[5][3][10]; // 3차원 배열
```

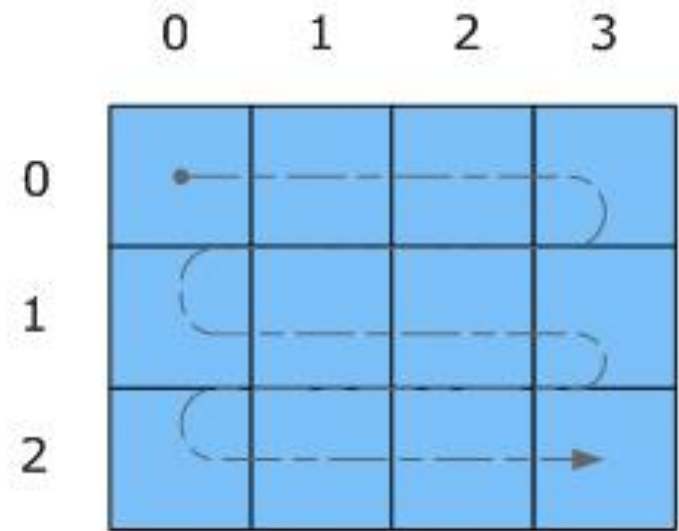




- ❑ 2차원 배열은 1차원적으로 구현된다
- ❑ Row-major vs. Column-major



A: Column-major order (Fortran-style)



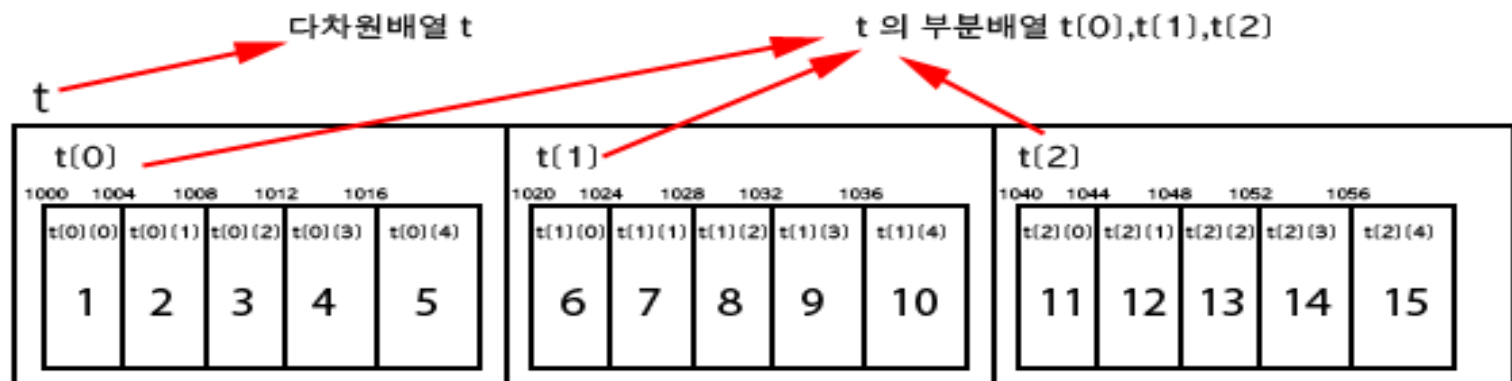
B: Row-major order (C-style)

```
int t(3)(5)={{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15}};
```

한 원소의 크기

int t (3) (5);

한원소의 크기(int형 5개)



Two Dimensional Array

1	5	3	6
3	2	38	64
22	76	82	99
0	106	345	54

User's view (abstraction)

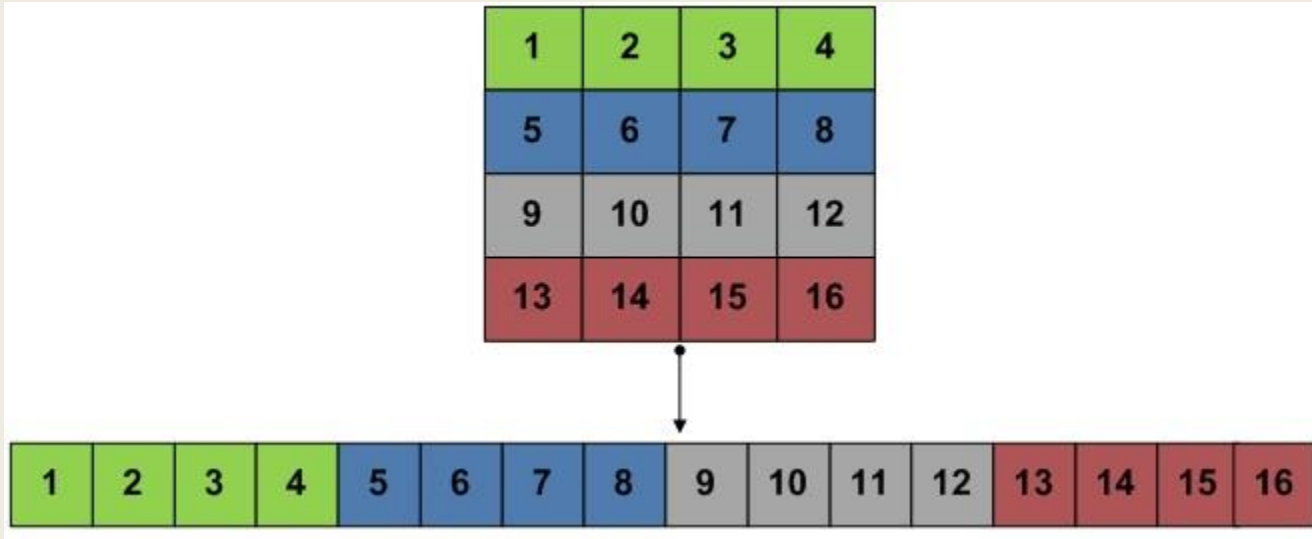
1	5	3	6	3	2	38	64	22	76	82	99	0	106	345	54
---	---	---	---	---	---	----	----	----	----	----	----	---	-----	-----	----

System's view
(implementation)

Offset of $a[i][j]$?

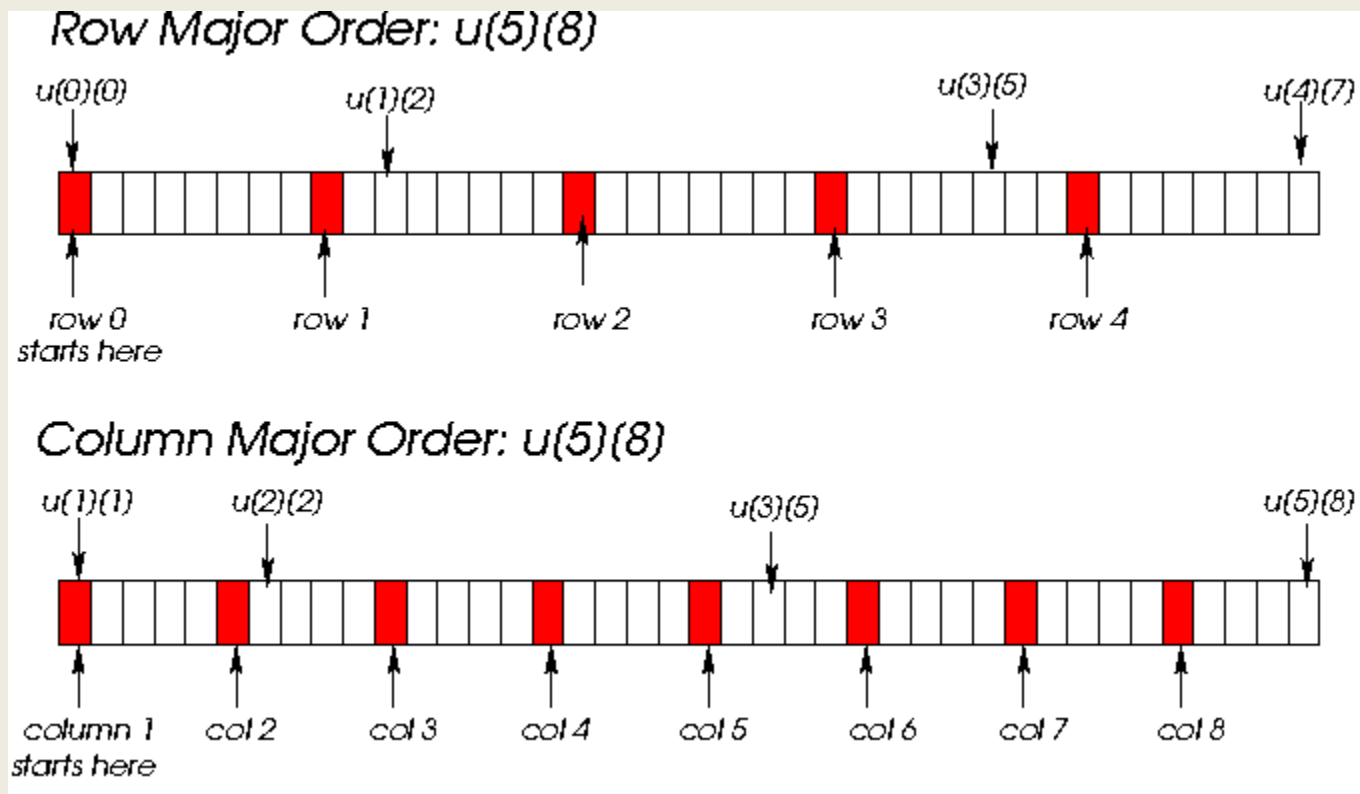


Example 1





□ Example 2



a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]

	주소의 배열식 표기법	주소의 배열 & 포인터식 표기법	주소의 포인터식 표기법	메모리	값의 배열식 표기법	값의 배열 & 포인터식 표기법	값의 포인터식 표기법
a	&a[0][0]	a[0]	*(a+0)+0	1000	a[0][0]	*(a[0])	*(*(a+0)+0)
	&a[0][1]	a[0]+1	*(a+0)+1	1004	a[0][1]	*(a[0]+1)	*(*(a+0)+1)
	&a[0][2]	a[0]+2	*(a+0)+2	1008	a[0][2]	*(a[0]+2)	*(*(a+0)+2)
	&a[0][3]	a[0]+3	*(a+0)+3	1012	a[0][3]	*(a[0]+3)	*(*(a+0)+3)
a+1	&a[1][0]	a[1]	*(a+1)+0	1016	a[1][0]	*(a[1])	*(*(a+1)+0)
	&a[1][1]	a[1]+1	*(a+1)+1	1020	a[1][1]	*(a[1]+1)	*(*(a+1)+1)
	&a[1][2]	a[1]+2	*(a+1)+2	1024	a[1][2]	*(a[1]+2)	*(*(a+1)+2)
	&a[1][3]	a[1]+3	*(a+1)+3	1028	a[1][3]	*(a[1]+3)	*(*(a+1)+3)
a+2	&a[2][0]	a[2]	*(a+2)+0	1032	a[2][0]	*(a[2])	*(*(a+2)+0)
	&a[2][1]	a[2]+1	*(a+2)+1	1036	a[2][1]	*(a[2]+1)	*(*(a+2)+1)
	&a[2][2]	a[2]+2	*(a+2)+2	1040	a[2][2]	*(a[2]+2)	*(*(a+2)+2)
	&a[2][3]	a[2]+3	*(a+2)+3	1044	a[2][3]	*(a[2]+3)	*(*(a+2)+3)

```

C:\WINDOWS\system32\cmd.exe

&array      : 0012FF40      &array+1    : 0012FF70
array       : 0012FF40      array+1     : 0012FF50
array[0]    : 0012FF40      array[0]+1 : 0012FF44
  
```


2차원 배열의 활용

```
#include <stdio.h>

int main(void)
{
    int s[3][5];    // 2차원 배열 선언
    int i, j;       // 2개의 인덱스 변수
    int value = 0;   // 배열 원소에 저장되는 값

    for(i=0;i<3;i++)
        for(j=0;j<5;j++)
            s[i][j] = value++;

    for(i=0;i<3;i++)
        for(j=0;j<5;j++)
            printf("%d\\n", s[i][j]);

    return 0;
}
```

2차원 배열의 초기화

```
int s[3][5] = {  
    { 0, 1, 2, 3, 4 }, // 첫 번째 행의 원소들의 초기값  
    { 10, 11, 12, 13, 14 }, // 두 번째 행의 원소들의 초기값  
    { 20, 21, 22, 23, 24 } // 세 번째 행의 원소들의 초기값  
};
```

```
int s[ ][5] = {  
    { 0, 1, 2, 3, 4 }, // 첫 번째 행의 원소들의 초기값  
    { 10, 11, 12, 13, 14 }, // 두 번째 행의 원소들의 초기값  
    { 20, 21, 22, 23, 24 }, // 세 번째 행의 원소들의 초기값  
};
```

```
int s[ ][5] = {  
    { 0, 1, 2 },           // 첫 번째 행의 원소들의 초기값  
    { 10, 11, 12 },       // 두 번째 행의 원소들의 초기값  
    { 20, 21, 22 }        // 세 번째 행의 원소들의 초기값  
};
```

```
int s[ ][5] = {  
    0, 1, 2, 3, 4,        // 첫 번째 행의 원소들의 초기값  
    5, 6, 7, 8, 9,        // 두 번째 행의 원소들의 초기값  
};
```

3차원 배열

```
int s [6][3][5];
```

첫번째 두번째 세번째
인덱스: 인덱스: 인덱스:
학년번호 학급번호 학생번호

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

```
int main(void)
```

```
{
```

```
    int s[3][3][3];    // 3차원 배열 선언  
    int x, y, z;       // 3개의 인덱스 변수  
    int i = 1;         // 배열 원소에 저장되는 값
```

```
    for(z=0;z<3;z++)  
        for(y=0;y<3;y++)  
            for(x=0;x<3;x++)  
                s[z][y][x] = i++;
```

```
    return 0;
```

```
}
```

다차원 배열 인수

```
#include <stdio.h>
#define YEARS    3
#define PRODUCTS 5

int sum(int grade[][PRODUCTS]);

int main(void)
{
    int sales[YEARS][PRODUCTS] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };
    int total_sale;
    total_sale = sum(sales);
    printf("총매출은 %d입니다.\n", total_sale);
    return 0;
}

int sum(int grade[][PRODUCTS])
{
    int y, p;
    int total = 0;
    for(y = 0; y < YEARS; y++)
        for(p = 0; p < PRODUCTS; p++)
            total += grade[y][p];
    return total;
}
```

첫번째 인덱스의 크기는
적지 않아도 된다.

다차원 배열 예제

학급 0의 평균 성적 = 2
학급 1의 평균 성적 = 12
학급 2의 평균 성적 = 22
전체 학생들의 평균 성적 = 12

```
#include <stdio.h>
#define CLASSES 3
#define STUDENTS 5

int main(void)
{
    int s[CLASSES][STUDENTS] = {
        { 0, 1, 2, 3, 4 },    // 첫번째 행의 원소들의 초기값
        { 10, 11, 12, 13, 14 }, // 두번째 행의 원소들의 초기값
        { 20, 21, 22, 23, 24 }, // 세번째 행의 원소들의 초기값
    };
    int clas, student, total, subtotal;
    total = 0;
    for(clas = 0; clas < CLASSES; clas++)
    {
        subtotal = 0;
        for(student = 0; student < STUDENTS; student++)
            subtotal += s[clas][student];
        printf("학급 %d의 평균 성적 = %d\n", clas, subtotal / STUDENTS);
        total += subtotal;
    }
    printf("전체 학생들의 평균 성적 = %d\n", total/(CLASSES * STUDENTS));
    return 0;
}
```

다차원 배열을 이용한 행렬의 표현

```
#include <stdio.h>
#define ROWS 3
#define COLS 3

int main(void)
{
    int A[ROWS][COLS] = { { 2,3,0 }, { 8,9,1 }, { 7,0,5 } };
    int B[ROWS][COLS] = { { 1,0,0 }, { 1,0,0 }, { 1,0,0 } };
    int C[ROWS][COLS];
    int r,c;
    // 두개의 행렬을 더한다.
    for(r = 0; r < ROWS; r++)
        for(c = 0; c < COLS; c++)
            C[r][c] = A[r][c] + B[r][c];
    // 행렬을 출력한다.
    for(r = 0; r < ROWS; r++)
    {
        for(c = 0; c < COLS; c++)
            printf("%d ", C[r][c]);
        printf("\n");
    }
    return 0;
}
```

3	3	0
9	9	1
8	0	5