

Object-Oriented Programming in C++

Arrays and Vectors - 2

Professor: 최학남, xncui@inha.ac.kr

Office: high-tech 401

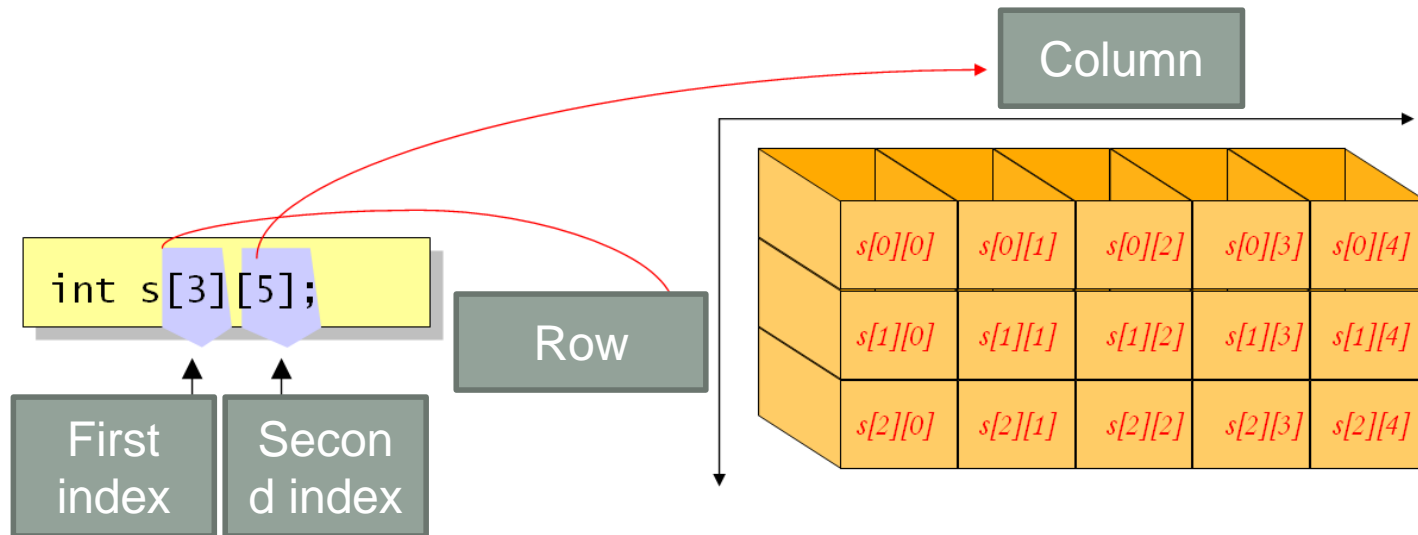


Contents

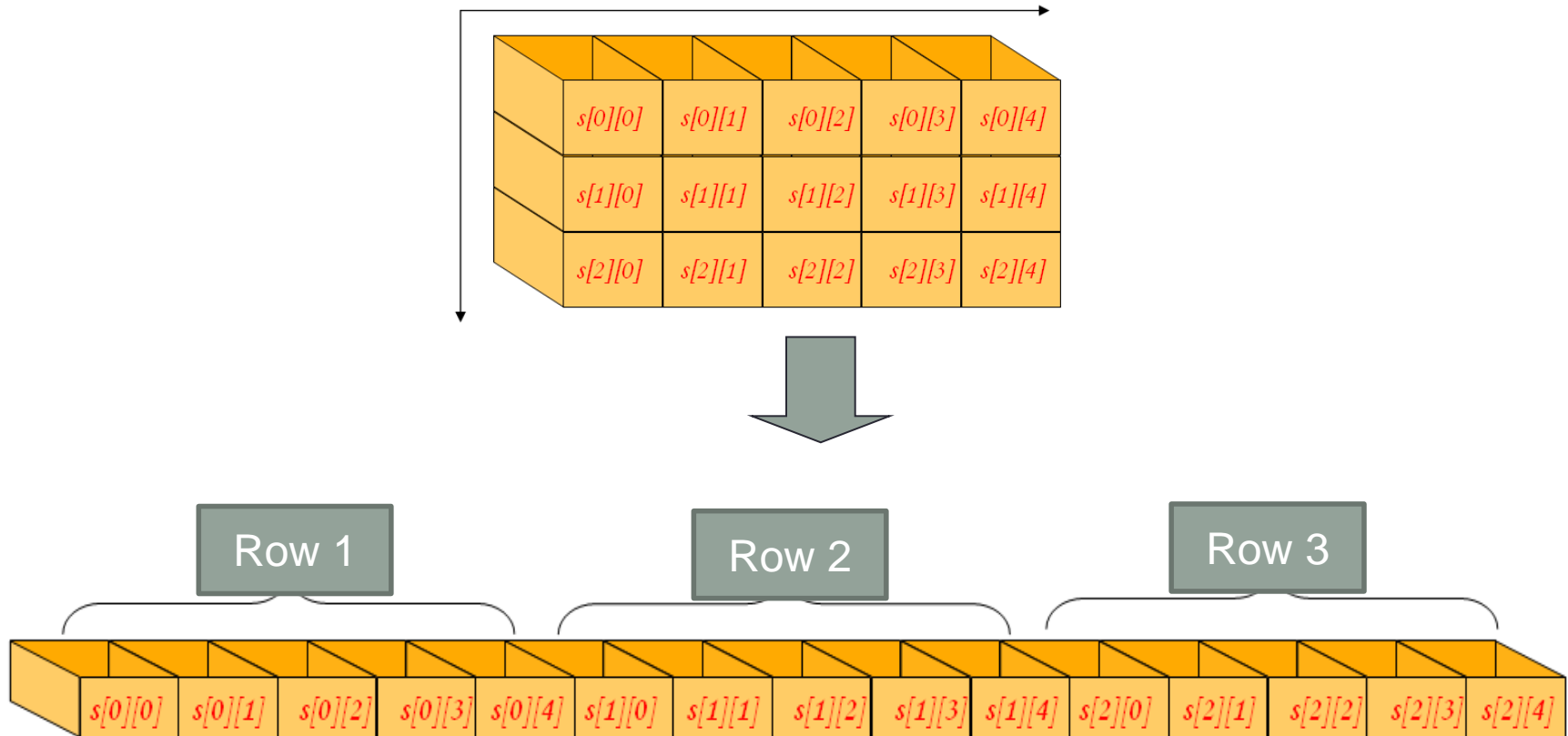
- What are arrays
- A simple program using array
 - ✓ Array declaration
 - ✓ Array initialization
- 1D-Array
- 2D-Array
- ND-Array

2D Array

```
int s[10];    // 1D Array  
int s[3][10]; // 2D Array  
int s[5][3][10]; // 3D Array
```



2D Array



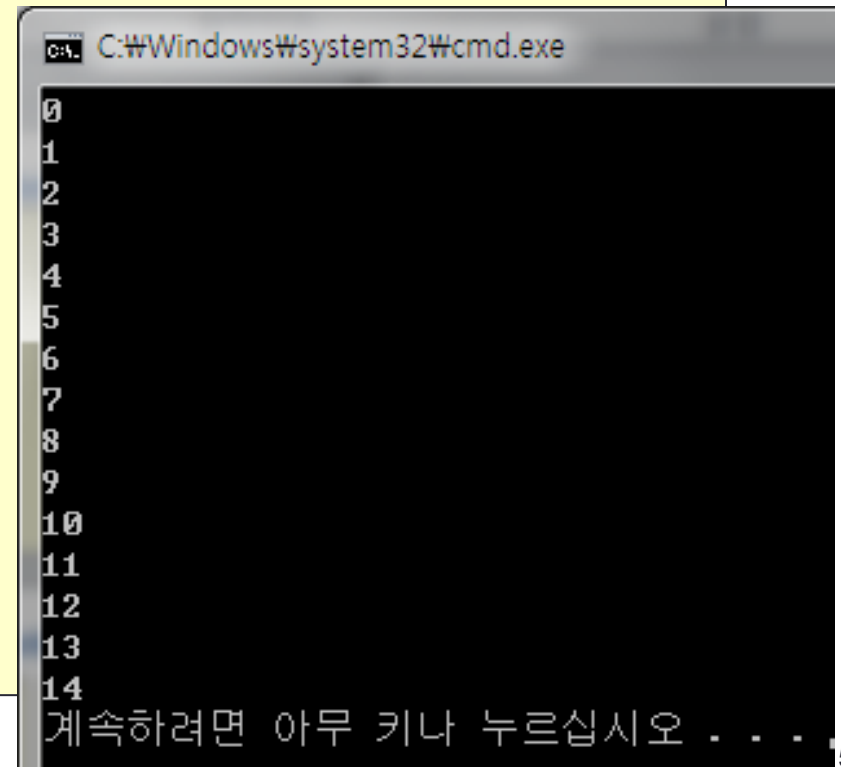
EX #11-1 2D Array

```
#include <stdio.h>

int main(void)
{
    int s[3][5];      // declaration of 2D array
    int i, j;         // for array index
    int value = 0;

    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;
}
```



```
C:\Windows\system32\cmd.exe
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
계속하려면 아무 키나 누르십시오 . . .
```



EX #11-2 2D Array

```
#include <stdio.h>

int main(void)
{
    int s[3][5];    // declaration of 2D array
    int i, j;       // for array index
    int value = 0;

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

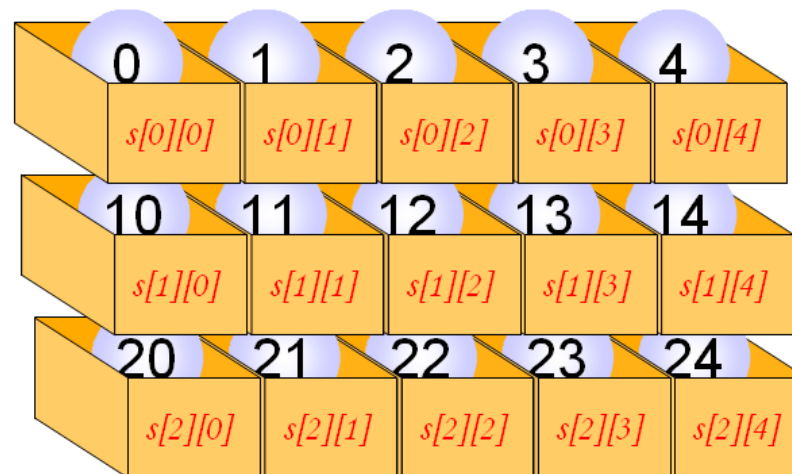
    for(i=0;i<3;i++) {
        for(j=0;j<5;j++) {
            printf("%3d", s[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

C:\WINDOWS\system32\cmd.exe

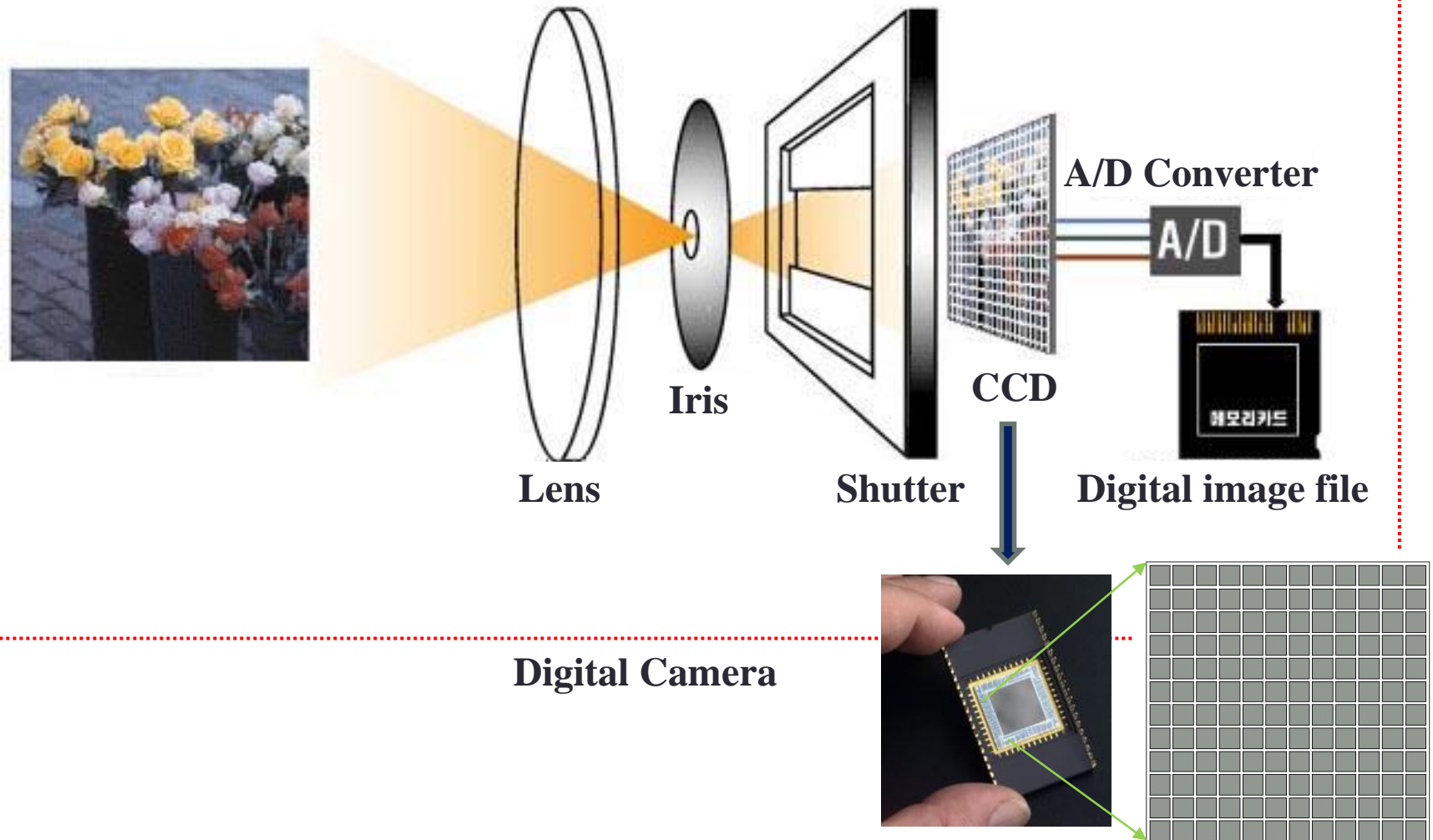
```
0  1  2  3  4
5  6  7  8  9
10 11 12 13 14
계속하려면 아무 키나 누르십시오
```

2D Array initialization

```
int s[3][5] = {  
    { 0, 1, 2, 3, 4}, // first row  
    { 10, 11, 12, 13, 14}, // second row  
    { 20, 21, 22, 23, 24} // third row  
};
```



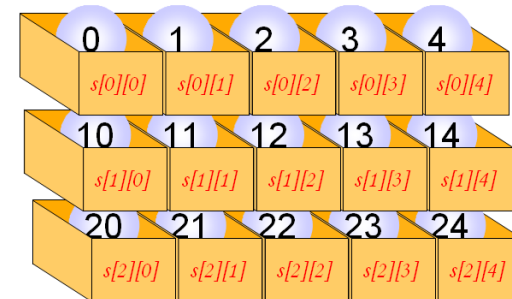
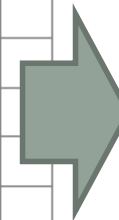
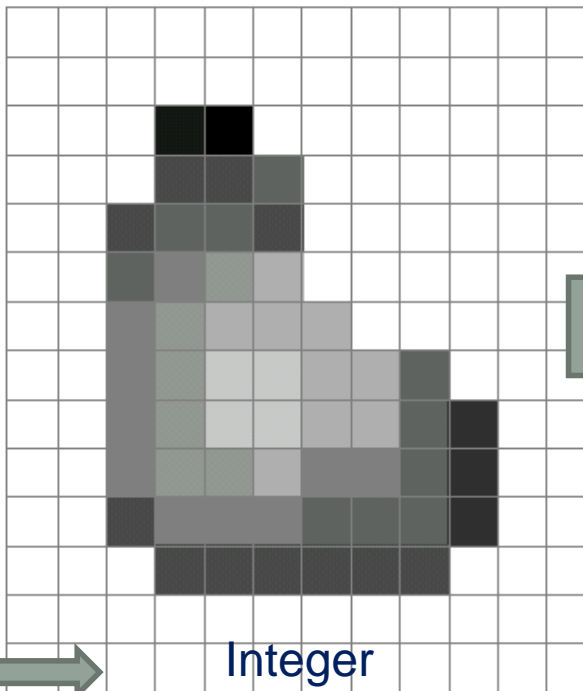
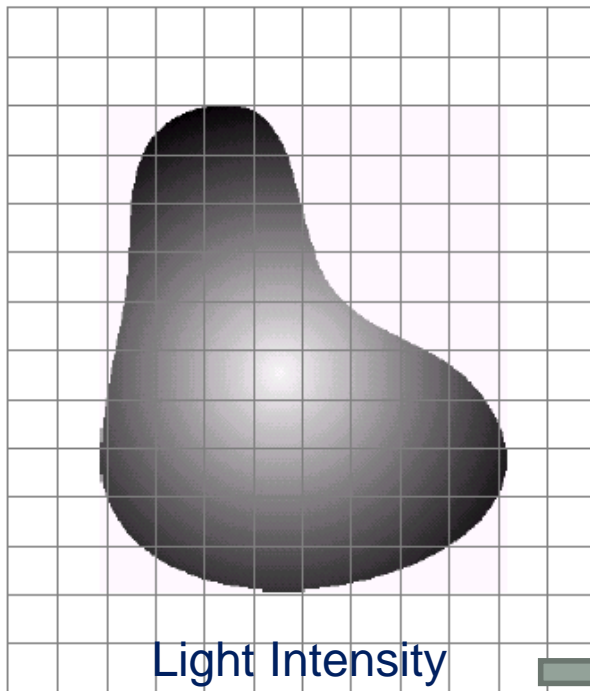
Digital Camera System



Digital Image

Natural Scene

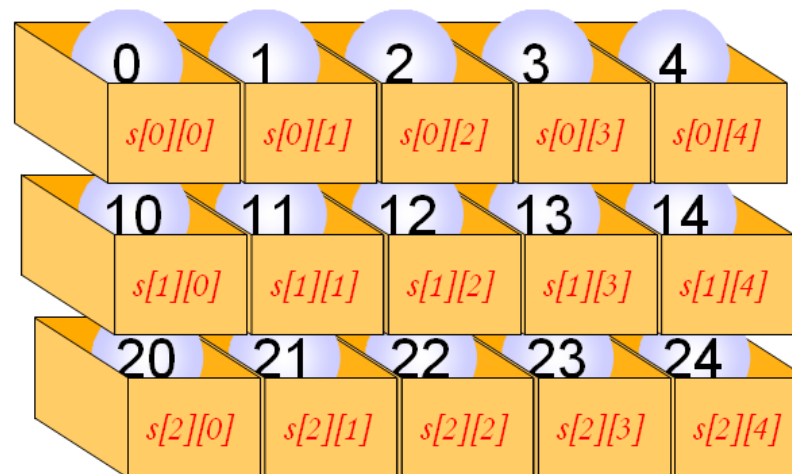
B/W(or Gray) Image



Sampling Quantization

2D Array initialization

```
int s[ ][5] = {  
    { 0, 1, 2, 3, 4}, // first row  
    { 10, 11, 12, 13, 14}, // second row  
    { 20, 21, 22, 23, 24} // third row  
};
```



EX 12: 3D Array

```
#include<iostream>
using namespace std;
int main(void)
{
    int s[6][3][5]={0};
    int x, y, z;
    int i = 1;

    for (z = 0; z < 6; z++) {
        for (y = 0; y < 3; y++) {
            for (x = 0; x < 5; x++) {
                s[z][y][x] = i++;
                cout << s[z][y][x] << " ";
            }
            cout << endl;
        }
        cout << endl;
    }

    return 0;
}
```

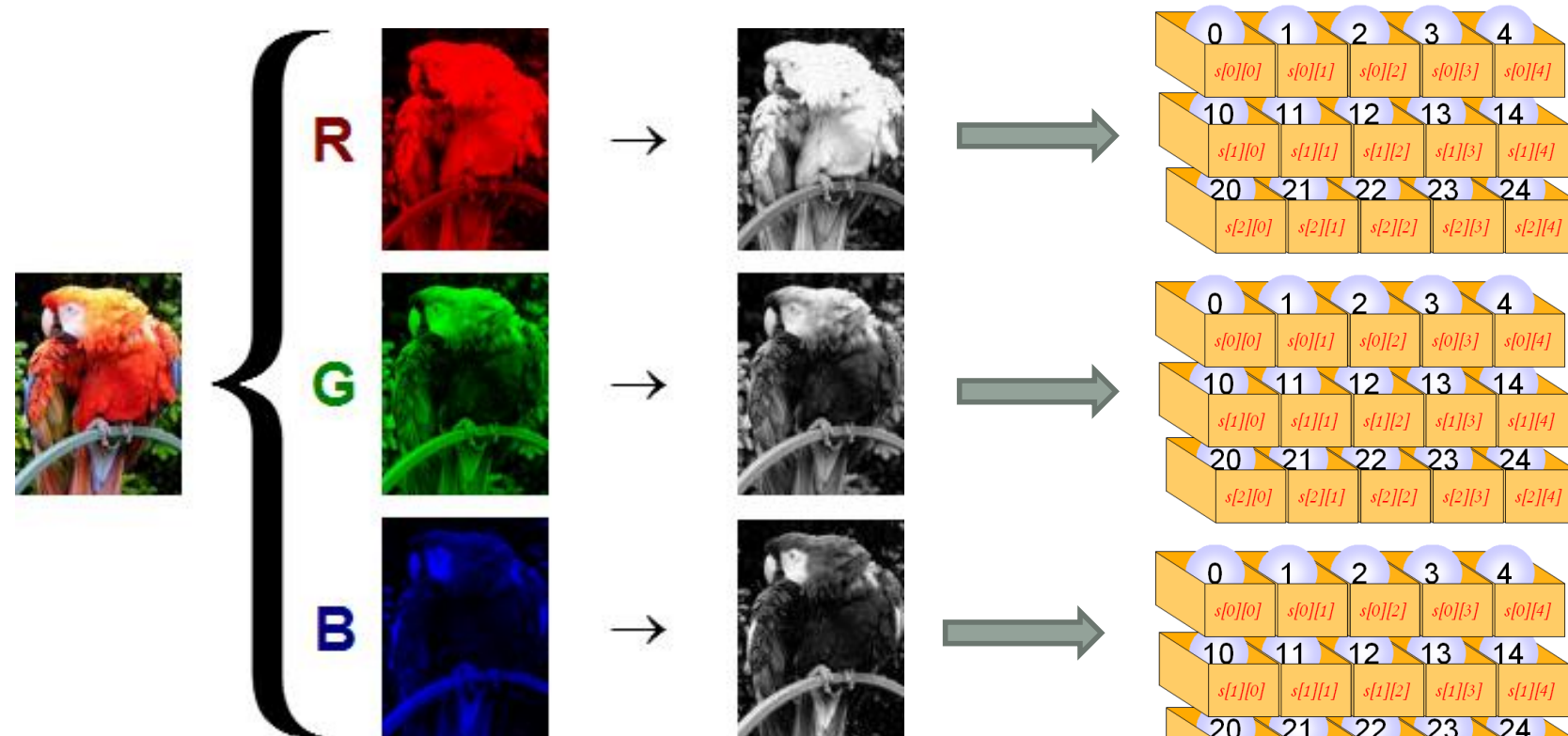
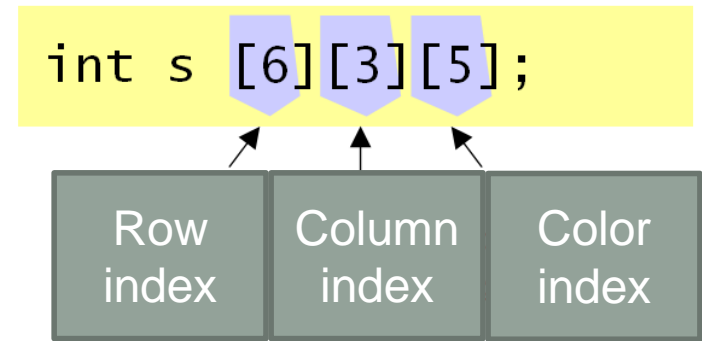
int s [6][3][5];

First
index

Second
index

Third
index

Color Image



Ex 13: Reverse

```
1  #include <stdio.h>
2  #define SIZE 5
3
4  int main(void)
5  {
6      int data[SIZE];
7      int i;
8
9      for(i = 0; i < SIZE; i++)    // 정수를 입력받는 루프
10     {
11         printf("정수를 입력하시오:");
12         scanf("%d", &data[i]);
13     }
14
15     for(i = SIZE - 1; i >= 0; i--) // 역순으로 출력하는 루프
16     {
17         printf("%d\n", data[i]);
18     }
19
20     return 0;
21 }
```



Ex 14: frequency

```
1  #include <stdio.h>
2  #define SIZE 101
3
4  int main(void)
5  {
6      int freq[SIZE];
7      int i, score;
8
9      for(i = 0; i < SIZE; i++)
10         freq[i] = 0;
11
12     while(1)
13     {
14         printf("숫자를 입력하시오(종료-1): ");
15         scanf("%d", &score);
16         if (score < 0) break;
17         freq[score]++;
18     }
19
20     printf("값  빈도\n");
21
22     for(i = 0; i < SIZE; i++)
23         printf("%3d  %3d \n", i, freq[i]);
24
25     return 0;
26 }
```



Ex 15: dice

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define SIZE 6
5
6  int main(void)
7  {
8      int freq[SIZE] = { 0 };    // 주사위의 면의 빈도를 0으로 한다.
9      int i;
10
11     for(i = 0; i < 10000; i++)    // 주사위를 10000번 던진다.
12         ++freq[ rand() % 6 ];    // 해당면의 빈도를 하나 증가한다.
13
14     printf("=====\n");
15     printf("면   빈도\n");
16     printf("=====\n");
17
18     for(i = 0; i < SIZE; i++)
19         printf("%3d   %3d\n", i, freq[i]);
20
21     return 0;
22 }
```



Ex 16: sorting - selection sort

```

1  #include <stdio.h>
2  #define SIZE 10
3
4  void selection_sort(int list[], int n);
5  void print_list(int list[], int n);
6
7  int main(void)
8  {
9      int grade[SIZE] = { 3, 2, 9, 7, 1, 4, 8, 0, 6, 5 };
10
11     // 원래의 배열 출력
12     printf("원래의 배열\n");
13     print_list(grade, SIZE);
14
15     selection_sort(grade, SIZE);
16
17     // 정렬된 배열 출력
18     printf("정렬된 배열\n");
19     print_list(grade, SIZE);
20
21     return 0;
22 }
23

```

```

24 void print_list(int list[], int n)
25 {
26     int i;
27     for(i = 0; i < n; i++)
28         printf("%d ", list[i]);
29     printf("\n");
30 }
31
32 void selection_sort(int list[], int n)
33 {
34     int i, j, temp, least;
35
36     for(i = 0; i < n-1; i++)
37     {
38         least = i;
39
40         for(j = i + 1; j < n; j++) // 최소값 탐색
41             if(list[j] < list[least])
42                 least = j;
43         // i번째 원소와 least 위치의 원소를 교환
44         temp = list[i];
45         list[i] = list[least];
46         list[least] = temp;
47     }
48 }

```



Ex 17: search - sequential search

```
1  #include <stdio.h>
2  #define SIZE 10
3
4  int seq_search(int list[], int n, int key);
5
6  int main(void)
7  {
8      int key;
9      int grade[SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
10
11     printf("탐색할 값을 입력하시오:");
12     scanf("%d", &key);
13     printf("탐색 결과 = %d\n", seq_search(grade, SIZE, key));
14
15     return 0;
16 }
17
18 int seq_search(int list[], int n, int key)
19 {
20     int i;
21
22     for(i = 0; i < SIZE; i++)
23         if(list[i] == key)
24             return i; // 탐색이 성공하면 인덱스 반환
25     return -1;        // 탐색이 실패하면 -1 반환
26 }
```



Ex 18: search - binary search

```

1  #include <stdio.h>
2  #define SIZE 10
3  int binary_search(int list[], int n, int key);
4
5  int main(void)
6  {
7      int key;
8      int grade[SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
9
10     printf("탐색할 값을 입력하시오:");
11     scanf("%d", &key);
12     printf("탐색 결과= %d\n", binary_search(grade, SIZE, key));
13
14     return 0;
15 }
16

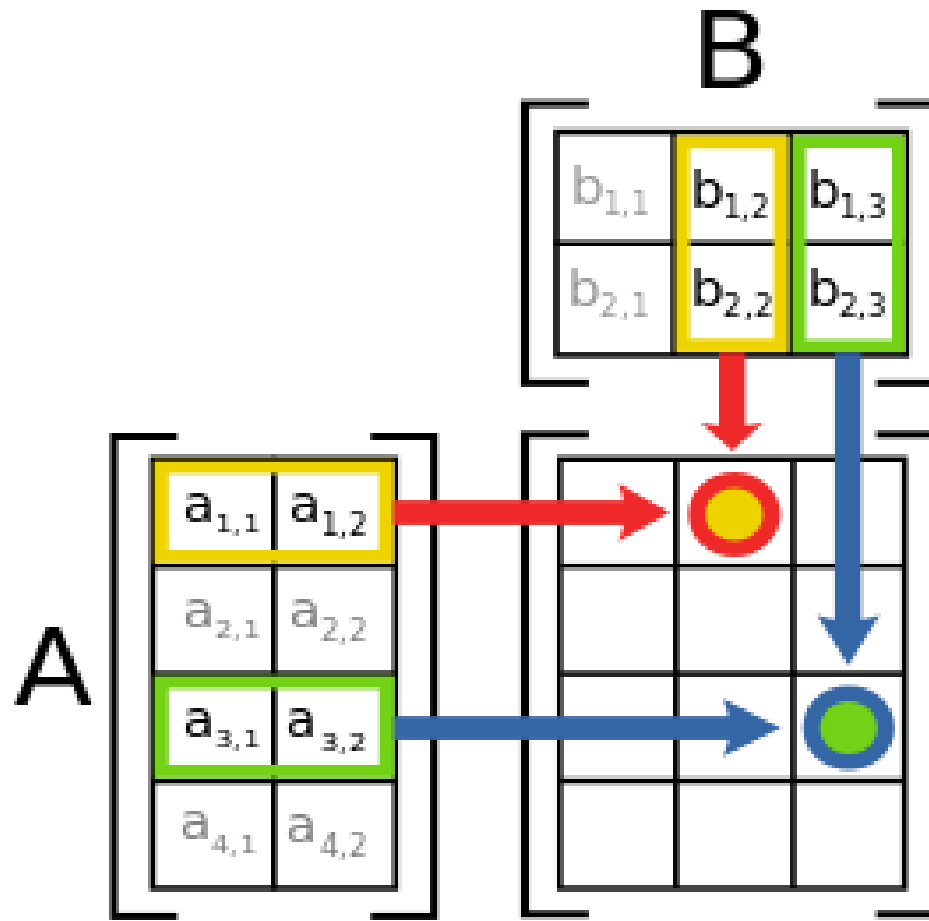
```

```

17 int binary_search(int list[], int n, int key)
18 {
19     int low, high, middle;
20
21     low = 0;
22     high = n-1;
23
24     while( low <= high ){    // 아직 숫자들이 남아있으면
25         middle = (low + high)/2;    // 중간 요소 결정
26         if( key == list[middle] ) // 일치하면 탐색 성공
27             return middle;
28         else if( key > list[middle] )// 중간 원소보다 크다면
29             low = middle + 1; // 새로운 값으로 low 설정
30         else
31             high = middle - 1; // 새로운 값으로 high 설정
32     }
33     return -1;
34 }
35

```

Matrix multiplication (행렬곱)



Matrix multiplication (행렬곱)

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix}$$

$$\mathbf{C} = \mathbf{AB}$$

$$\mathbf{C} = \begin{pmatrix} a_{11}b_{11} + \cdots + a_{1n}b_{n1} & a_{11}b_{12} + \cdots + a_{1n}b_{n2} & \cdots & a_{11}b_{1p} + \cdots + a_{1n}b_{np} \\ a_{21}b_{11} + \cdots + a_{2n}b_{n1} & a_{21}b_{12} + \cdots + a_{2n}b_{n2} & \cdots & a_{21}b_{1p} + \cdots + a_{2n}b_{np} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} + \cdots + a_{mn}b_{n1} & a_{m1}b_{12} + \cdots + a_{mn}b_{n2} & \cdots & a_{m1}b_{1p} + \cdots + a_{mn}b_{np} \end{pmatrix}$$

EX #19 Matrix expression using array

```

#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;
    // matrix addition.
    for(r = 0; r < ROWS; r++)
        for(c = 0; c < COLS; c++)
            C[r][c] = A[r][c] + B[r][c];
    // print the results.
    for(r = 0; r < ROWS; r++)
    {
        for(c = 0; c < COLS; c++)
            printf("%d ", C[r][c]);
        printf("\n");
    }
    // matrix multiplication.
    int D[ROWS][COLS] = { 0 };
    //add matrix multiplication code here

    // print the results.
    for(r = 0; r < ROWS; r++)
    {
        for(c = 0; c < COLS; c++)
            printf("%d ", D[r][c]);
        printf("\n");
    }
    return 0;
}

```

$$A = \begin{bmatrix} 2 & 3 & 0 \\ 8 & 9 & 1 \\ 7 & 0 & 5 \end{bmatrix}$$

3	3	0
9	9	1
8	0	5

5	0	0
18	0	0
12	0	0



HW #7

Declare a 20x20 array and provide following services. Use as many functions as possible.

1. fill: fill the array using `rand()%100`
2. show: show the array
3. row sum: read a row number and show the sum of that row
4. col sum: read a column number and show the sum of that column
5. diagonal sum: show the diagonal sum
6. backward diagonal sum: show the backward diagonal sum
7. row max: read a row number and show the max of that row
8. col max: read a column number and show the max of that column
9. max row: find the row whose sum is the greatest
10. max col: find the column whose sum is the greatest



Project #1

1. For the following set of n data points (x, y) , compute the correlation coefficient r , given by

$$r = \frac{n \sum xy - \sum x \sum y}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}}$$

x	y
34.22	102.43
39.87	100.93
41.85	97.43
43.23	97.81
40.06	98.32
53.29	98.32
53.29	100.07
54.14	97.08
49.12	91.59
40.71	94.85
55.15	94.65

Have to use the function and array

1. Compute the **correlation coefficient r**
2. Draw the function call structures of the developed source code.

Project #1

2. For the following set of point given by (x, y) fit a straight line given by

$$y = a + bx$$

where,

$$a = \bar{y} - b\bar{x} \quad \text{and}$$

$$b = \frac{n \sum yx - \sum x \sum y}{[n \sum x^2 - (\sum x)^2]}$$

x	y
3.0	1.5
4.5	2.0
5.5	3.5
6.5	5.0
7.5	6.0
8.5	7.5
8.0	9.0
9.0	10.5
9.5	12.0
10.0	14.0

Have to use the function and array

1. Compute the **a** and **b**
2. Draw the function call structures of the developed source code.

Deadline: two weeks from today (refer the e-class)



저작권 안내 / Copyright Notice

본 사이트에서 수업 자료로 이용되는 저작물은 저작권법 제25조 수업목적저작물 이용 보상금제도의 의거, 한국복제전송저작권협회와 약정을 체결하고 적법하게 이용하고 있습니다. **약정범위를 초과하는 사용은 저작권법에 저촉될 수 있으므로 수업자료의 대중 공개-공유 및 수업 목적외의 사용을 금지합니다. (일부 공개로 제공되는 유튜브 강의를 불법으로 공유하거나 배포하는 행위를 금함)**

The teaching materials provided on this site are copyrighted and signed legally by the Korea Reproduction and Transmission Rights Association. Using these materials beyond the scope of the agreement is a violation against copyright laws. **Hence, one is not allowed to open or share the materials in public. Using them outside of the class is strictly prohibited.**

2020. 3. 11.

인하대학교·한국복제전송저작권협회

Inha University · Korea Reproduction and Transmission Rights Association