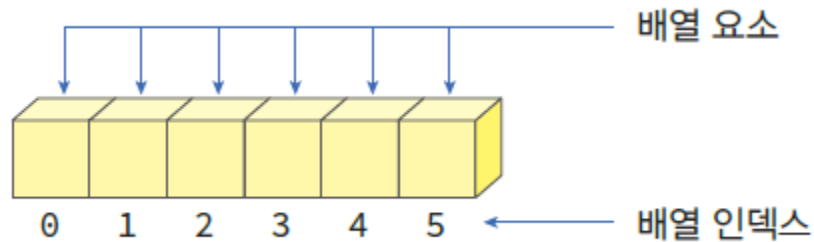


3장 배열



배열이란?

- 같은 형의 변수를 여러 개 만드는 경우에 사용
 - ▣ `int list1, list2, list3, list4, list5, list6; → int list[6];`



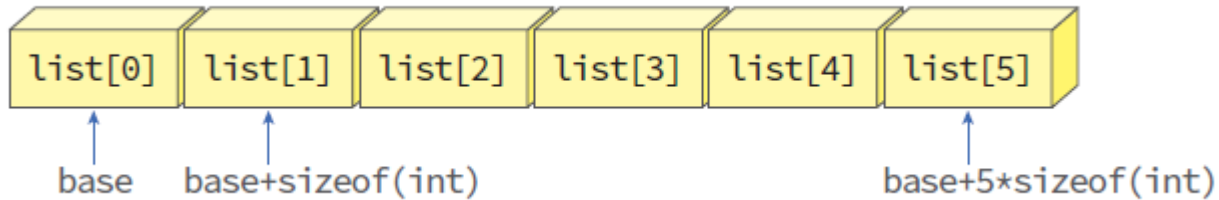


1 차원 배열

```
int list[6];
```

```
list[0] = 100;    // set 연산에 해당된다.
```

```
value = list[0];  // get 연산에 해당된다.
```





1 차원 배열의 값 정렬하기

- 선택정렬(selection sort)
- 버블정렬(bubble sort)



정렬이란?

- 정렬은 물건을 크기순으로 오름차순이나 내림차순으로 나열하는 것
- 정렬은 컴퓨터 공학분야에서 가장 기본적이고 중요한 알고리즘중의 하나

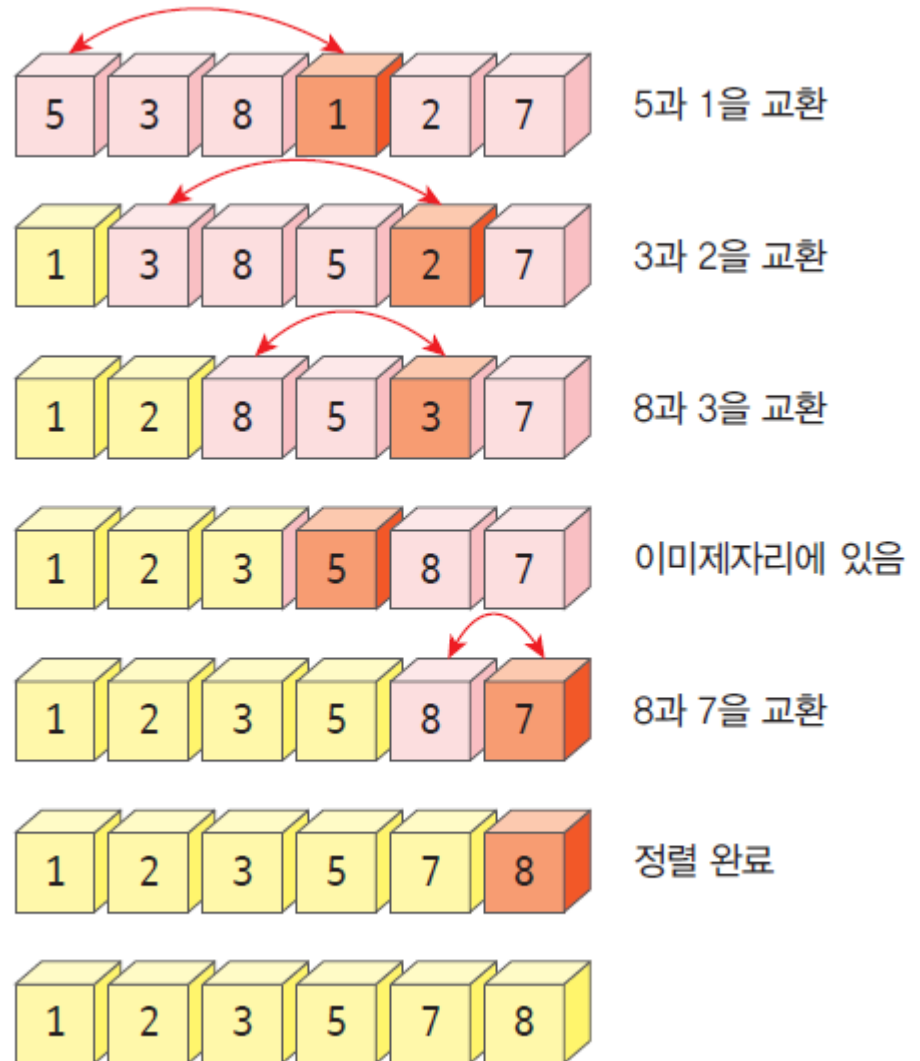




선택정렬(selection sort)

□ 선택정렬(selection sort):

정렬이 안된 숫자들중에서
최소값을 선택하여 배열의
첫번째 요소와 교환





선택정렬(selection sort)

```
#include<stdio.h>
#define SIZE 10

int main(void)
{
    int list[SIZE] = { 3,2,9,7,1,4,8,0,6,5 };
    int i, j, temp, least;

    for (i = 0; i < SIZE - 1; i++)
    {
        least = i;

        for (j = i + 1; j < SIZE; j++)
        {
            if (list[j] < list[least]) least = j;
        }

        temp = list[i];
        list[i] = list[least];
        list[least] = temp;
    }

    for (i = 0; i < SIZE; i++) printf("%d ", list[i]);

    return 0;
}
```



버블정렬(bubble sort)

- 버블정렬(bubble sort): 정렬이 안된 숫자들중에서 (첫번째, 두번째)비교(두번째, 세번째) 비교.....
- 1회전 정렬 끝나면 제일 큰수가 마지막에 자리한다.
- 마지막 수 제외하고 다시 정렬



버블정렬(bubble sort)

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

```
#define SIZE 10
```

```
int main(void)
```

```
{
```

```
    int list[SIZE] = { 3,2,9,7,1,4,8,0,6,5 };
```

```
    int i, j, temp, k;
```

```
    for (i = SIZE - 1; i > 0 ; i--)
```

```
    {
```

```
        for (j = 0; j < i; j++)
```

```
        {
```

```
            if (list[j] > list[j + 1])
```

```
            {
```

```
                temp = list[j];
```

```
                list[j] = list[j+1];
```

```
                list[j+1] = temp;
```

```
            }
```

```
        }
```

```
        for (k = 0; k < SIZE; k++) printf("%d ", list[k]);
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

문제

20 이하의 정수 n 을 입력받고 n 명의 점수를 입력받아 높은 점수부터 차례로 출력하는 프로그램을 작성하시오.

입력 예

```
5
35 10 35 100 64
```

출력 예

```
100
64
35
35
10
```

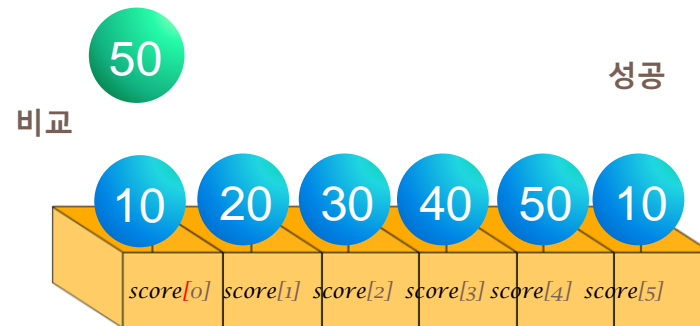


1 차원 배열의 값 탐색하기

- 순차탐색
- 이진탐색



- 순차 탐색은 배열의 원소를 순서대로 하나씩 꺼내서 탐색
키와 비교하여 원하는 값을 찾아가는 방법





```
#include <stdio.h>
#define SIZE 10
```

```
int main(void)
{
    int key, i;
    int list[SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

    printf("탐색할 값을 입력하시오:");
    scanf("%d", &key);

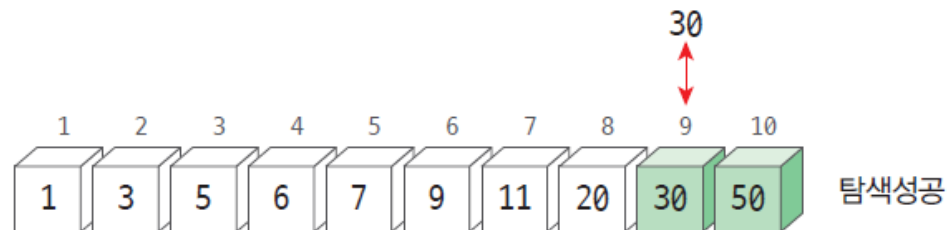
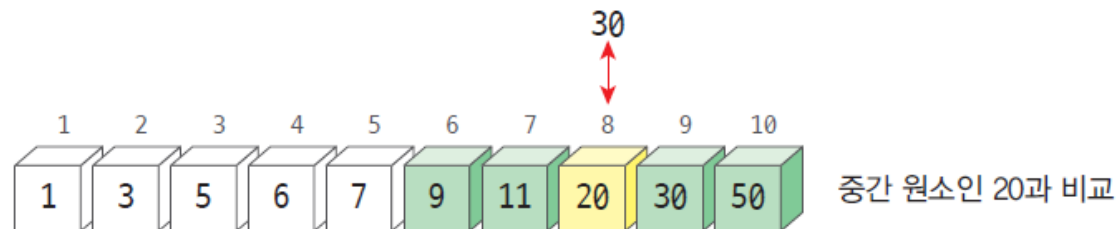
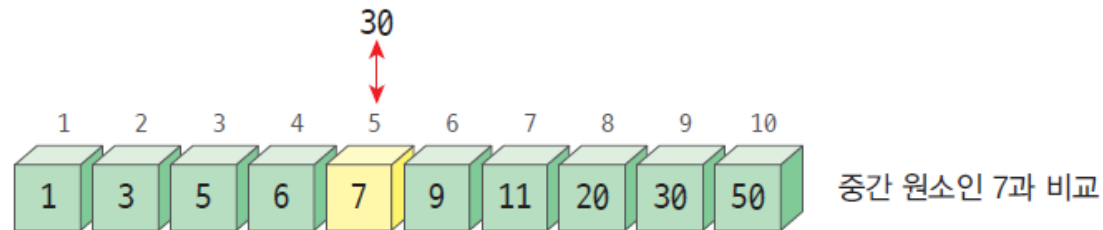
    for(i = 0; i < SIZE; i++)
        if(list[i] == key)
            printf("탐색 성공 인덱스= %d\n", i);

    printf("탐색 종료\n");
    return 0;
}
```

for 루프를 이용하여 list[i]와 key를 비교하는 연산을 배열의 크기만큼 반복한다. 만약 list[i]와 key가 같으면 탐색은 성공되고 키값이 발견된 배열의 인덱스를 출력한다.

탐색할 값을 입력하시오:7
탐색 성공 인덱스 = 6
탐색 종료

- 이진 탐색(binary search): 정렬된 배열의 중앙에 위치한 원소와 비교 되풀이





```
#include <stdio.h>
#define SIZE 16
int binary_search(int list[], int n, int key);

int main(void)
{
    int key;
    int grade [SIZE] = { 2,6,11,13,18,20,22,27,29,30,34,38,41,42,45,47 };
    printf("탐색할 값을 입력하시오:");
    scanf("%d", &key);
    printf("탐색 결과= %d\n", binary_search(grade, SIZE, key));

    return 0;
}
```



```
int binary_search(int list[], int n, int key)
{
    int low, high, middle;
    low = 0;
    high = n-1;
    while( low <= high ){
        printf("[%d %d]\n", low, high); // 아직 숫자들이 남아있으면
        middle = (low + high)/2;        // 하한과 상한을 출력한다.
        if( key == list[middle] )       // 중간 위치를 계산한다.
            return middle;             // 일치하면 탐색 성공
        else if( key > list[middle] ) // 중간 원소보다 크다면
            low = middle + 1;          // 새로운 값으로 low 설정
        else
            high = middle - 1;         // 새로운 값으로 high 설정
    }
    return -1;
}
```


2차원 배열

□ `int list[3][5];`

	0열	1열	2열	3열	4열
0행	<code>list[0][0]</code>	<code>list[0][1]</code>	<code>list[0][2]</code>	<code>list[0][3]</code>	<code>list[0][4]</code>
1행	<code>list[1][0]</code>	<code>list[1][1]</code>	<code>list[1][2]</code>	<code>list[1][3]</code>	<code>list[1][4]</code>
2행	<code>list[2][0]</code>	<code>list[2][1]</code>	<code>list[2][2]</code>	<code>list[2][3]</code>	<code>list[2][4]</code>



2차원 배열의 활용

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

#define ROWS 3
#define COLS 5

int main(void)
{
    int s[ROWS][COLS];
    int i, j;

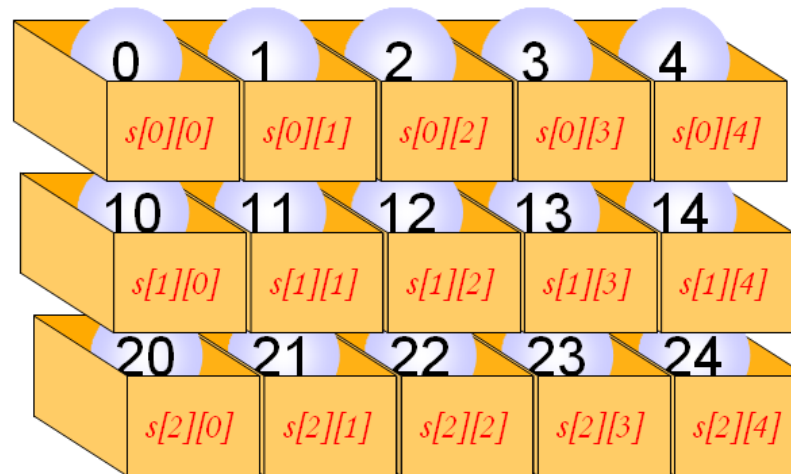
    for (i = 0; i < ROWS; i++)
    {
        for (j = 0; j < COLS; j++)
        {
            s[i][j] = rand() % 100;
        }
    }

    for (i = 0; i < ROWS; i++)
    {
        for (j = 0; j < COLS; j++)
        {
            printf("% 2d", s[i][j]);
        }
        printf("\n");
    }
}
```



2차원 배열의 초기화

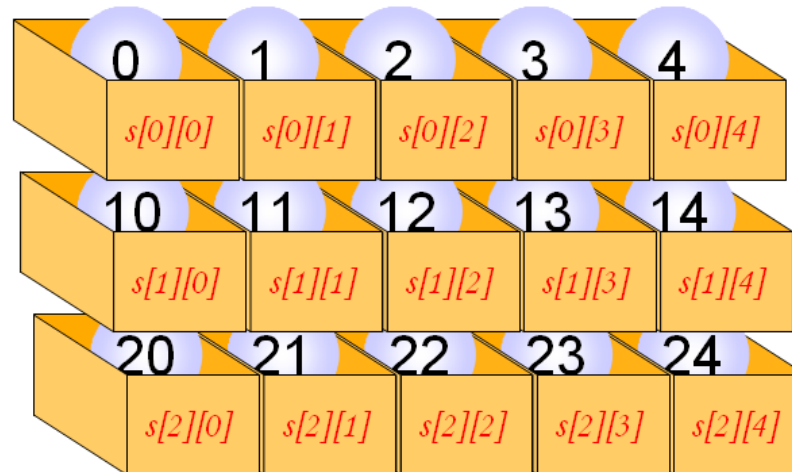
```
int s[3][5] = {  
    { 0, 1, 2, 3, 4}, // 첫 번째 행의 원소들의 초기값  
    { 10, 11, 12, 13, 14}, // 두 번째 행의 원소들의 초기값  
    { 20, 21, 22, 23, 24} // 세 번째 행의 원소들의 초기값  
};
```





2차원 배열의 초기화

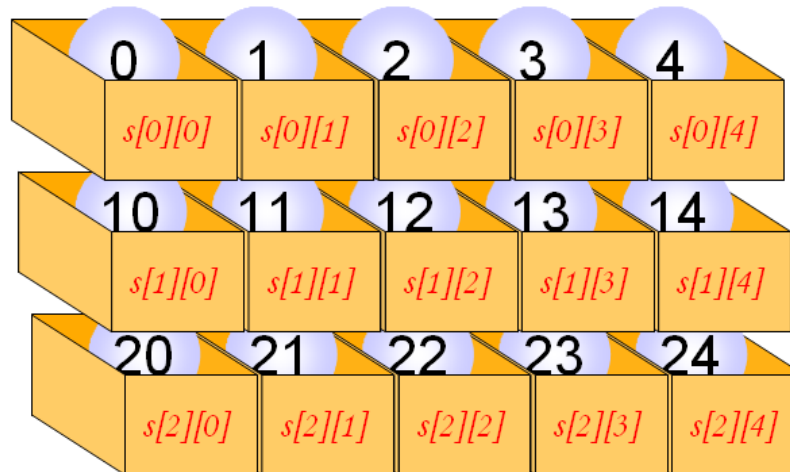
```
int s[ ][5] = {  
    { 0, 1, 2, 3, 4}, // 첫 번째 행의 원소들의 초기값  
    { 10, 11, 12, 13, 14}, // 두 번째 행의 원소들의 초기값  
    { 20, 21, 22, 23, 24}, // 세 번째 행의 원소들의 초기값  
};
```





2차원 배열의 초기화

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





- 학생들의 성적 기록표를 2차원 배열에 저장하고 각 학생의 최종 성적을 계산해보자.

학번	중간고사(30%)	기말고사(40%)	기말과제(20%)	퀴즈점수(10%)	결석횟수(감점)
1	87	98	80	76	3
2	99	89	90	90	0
3	65	68	50	49	0

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

```
#define ROWS 3
```

```
#define COLS 5
```

```
int main(void)
```

```
{
```

```
    int a[ROWS][COLS] = {
```

```
        {87,98,80,76,3},
```

```
        {99,89,90,90,0},
```

```
        {65,68,50,49,0}
```

```
    };
```

```
    int i;
```

```
    double scores;
```

```
    for (i = 0; i < ROWS; i++)
```

```
    {
```

```
        scores = a[i][0] * 0.3 + a[i][1] * 0.4 + a[i][2] * 0.2 + a[i][3] * 0.1 - a[i][4];
```

```
        printf("학생 %d 의 최종성적 = %10.2f 입니다. \n", (i + 1), scores);
```

```
    }
```

```
    return 0;
```

```
}
```

```
#include<stdio.h>

#define ROWS 3
#define COLS 5

void show(int s[ROWS][COLS])
{
    int i;
    double scores;
    for (i = 0; i < ROWS; i++)
    {
        scores = s[i][0] * 0.3 + s[i][1] * 0.4 + s[i][2] * 0.2 + s[i][3] * 0.1 - s[i][4];
        printf("학생 %d 의 최종성적 = %10.2f 입니다. \n", (i + 1), scores);
    }
}

int main(void)
{
    int a[ROWS][COLS] = {
        {87,98,80,76,3},
        {99,89,90,90,0},
        {65,68,50,49,0}
    };

    show(a);
    return 0;
}
```




2차원 배열 문제 1

3행 5열의 배열을 아래 출력 예와 같이 초기화 한 후 출력하는 프로그램을 작성하시오.

숫자사이의 공백은 3칸으로하고 각 숫자는 2칸으로 오른쪽에 맞추어 출력한다. 예, "%2d "

4

출력 예

5	8	10	6	4
11	20	1	13	2
7	9	14	22	3

4



2차원 배열 문제2

2행 4열의 배열 두 개를 만들어서 입력을 받고 두 배열의 곱을 구하여 출력하는 프로그램을 작성하시오.

입력 예

```
first array
1 2 3 4
5 6 7 8
second array
1 4 7 8
3 6 9 8
```

출력 예

```
1 8 21 32
15 36 63 64
```



2차원 배열 문제3

문제

4행 3열의 배열을 선언하고 아래의 값으로 초기화 한 후 차례대로 출력한 후 배열의 모든 값을 더한 결과를 출력하는 프로그램을 작성하시오.

```
{{3, 5, 9}, {2, 11, 5}, {8, 30, 10}, {22, 5, 1}}
```

출력 예

```
3 5 9
2 11 5
8 30 10
22 5 1
111
```