



누구나 즐기는 C언어 콘서트

제8장 배열





이번 장에서 학습할 내용

- 반복의 개념 이해
- 배열의 개념
- 배열의 선언과 초기화
- 일차원 배열
- 다차원 배열

배열을 사용하면
한 번에 여러
개의 값을 저장할
수 있는 공간을
할당받을 수
있다.





배열의 필요성

- 학생이 10명이 있고 이들의 평균 성적을 계산한다고 가정하자.

개별 변수를 사용
하는 방법은 학생
수가 많아지면 번
거로워집니다..



방법 #1: 개별 변수 사용

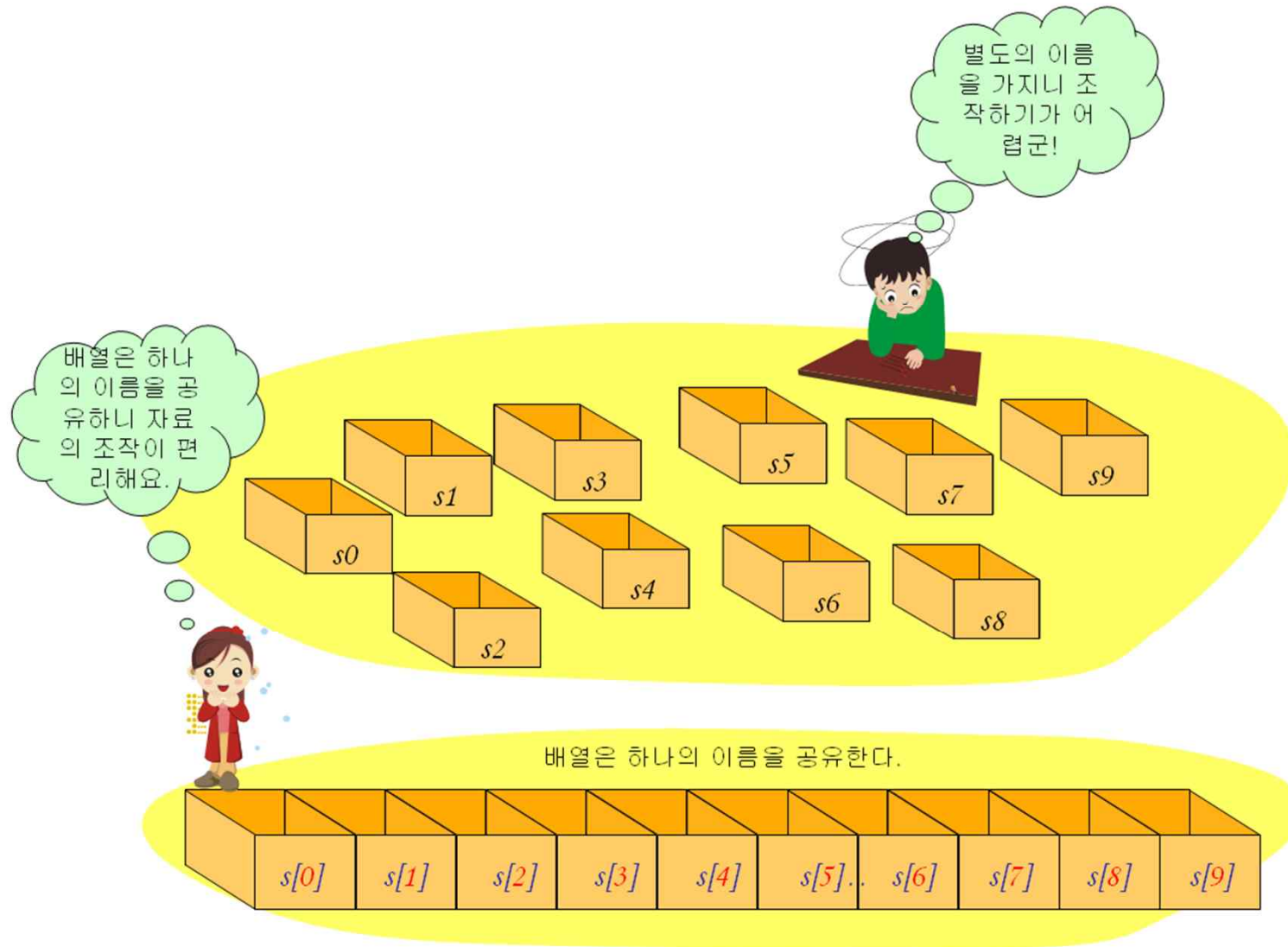
```
int s0;  
int s1;  
...  
int s9;
```

방법 #2: 배열 사용

```
int s[10];
```



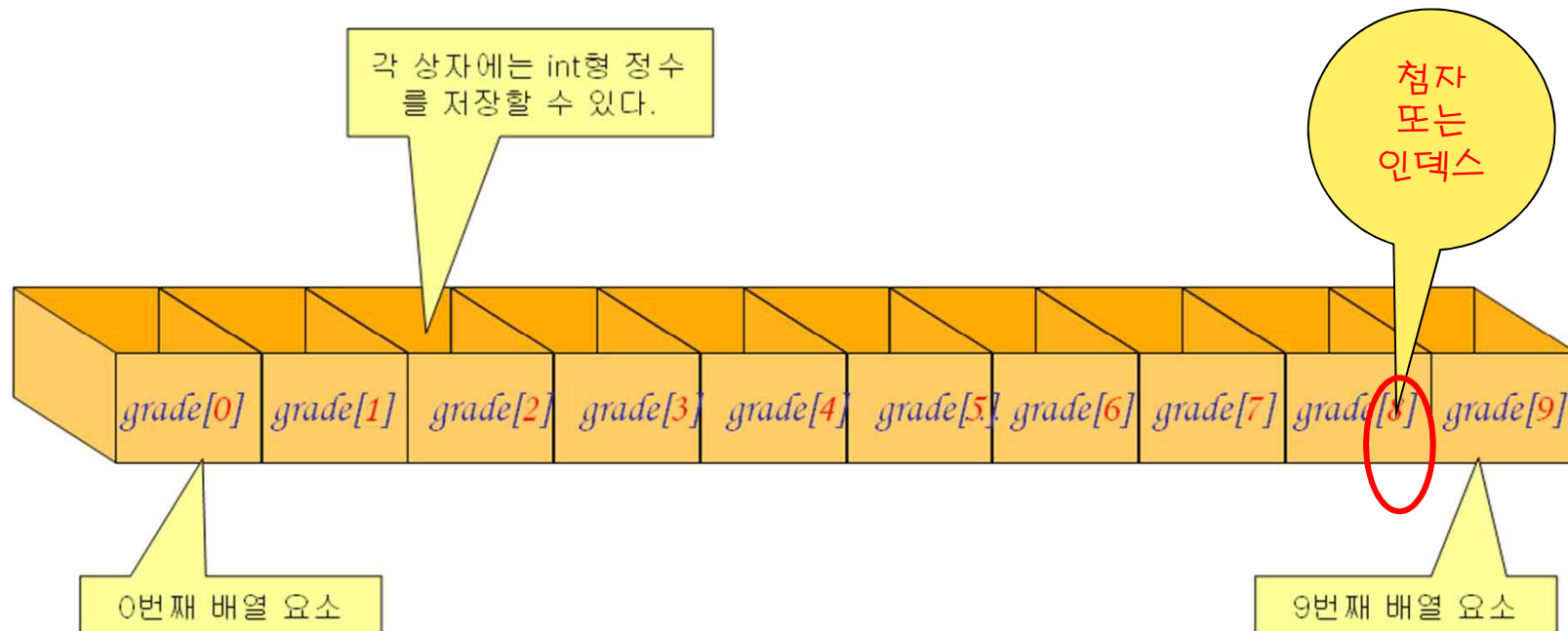
배열의 필요성





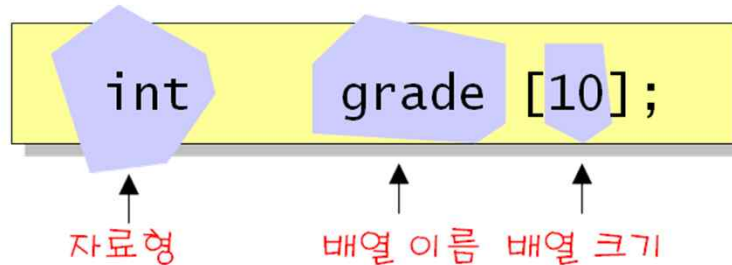
배열이란?

- **배열(array)**: 동일한 타입의 데이터가 여러 개 저장되어 있는 데이터 저장 장소
- 배열 안에 들어있는 각각의 데이터들은 정수로 되어 있는 **번호(인덱스)**에 의하여 접근
- 배열을 이용하면 여러 개의 값을 하나의 이름으로 처리할 수 있다.





배열의 선언

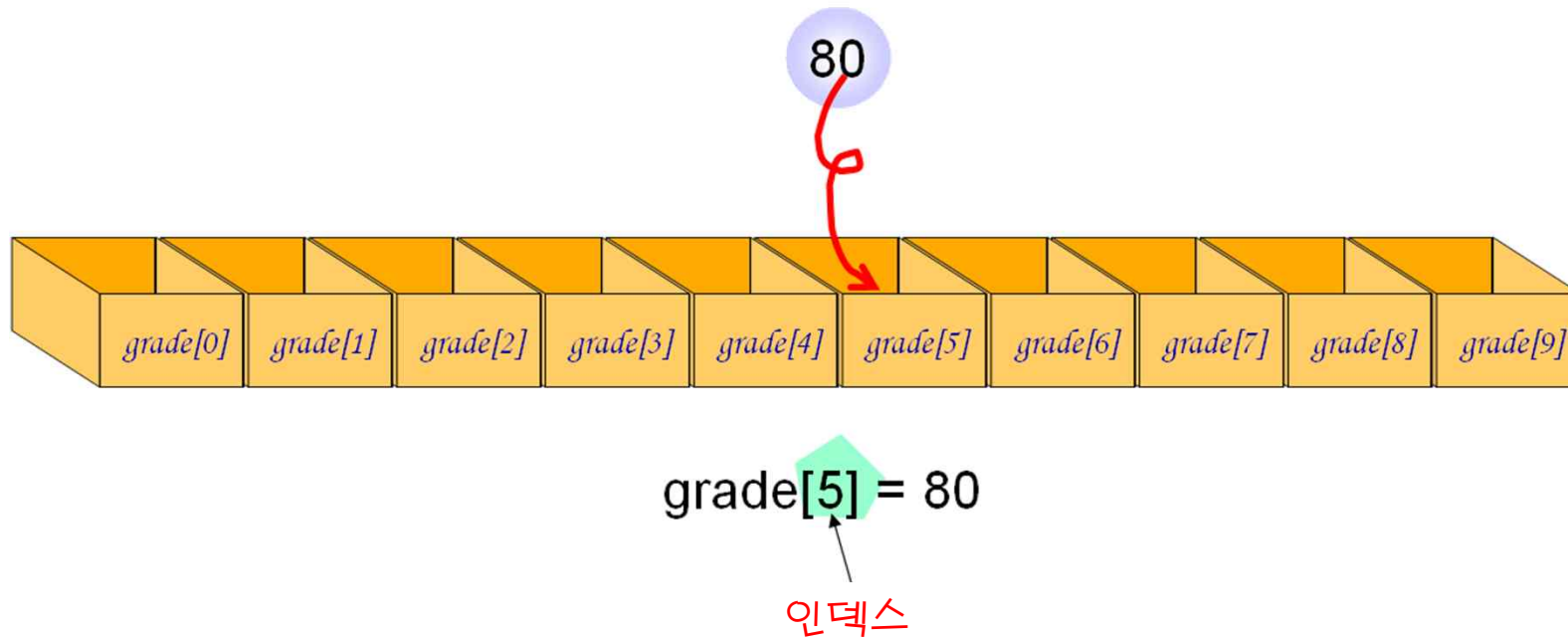


- 자료형: 배열 원소들이 **int**형라는 것을 의미
- 배열 이름: 배열을 사용할 때 사용하는 이름이 **grade**
- 배열 크기: 배열 원소의 개수가 **10**개
- 인덱스(첨자)는 항상 0부터 시작한다.

```
int score[60];           // 60개의 int형 값을 가지는 배열 grade
float cost[12];          // 12개의 float형 값을 가지는 배열 cost
char name[50];           // 50개의 char형 값을 가지는 배열 name
```



배열 원소 접근



(예)

```
grade[0] = 80;
```

```
grade[1] = grade[0];
```

```
grade[i] = 100;
```

```
grade[i+2] = 100;
```

// 0번째 원소에 80을 대입

// 0번째 원소를 1번째 원소로 복사

// i는 정수 변수

// 수식이 인덱스가 된다.



배열 선언 예제



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

```
int main(void)  
{
```

```
    int i;
```

```
    int grade[5];
```

```
    grade[0] = 10;
```

```
    grade[1] = 20;
```

```
    grade[2] = 30;
```

```
    grade[3] = 40;
```

```
    grade[4] = 50;
```

```
    for(i=0; i < 5; i++)
```

```
        printf("grade[%d]=%d\n", i, grade[i]);
```

```
    return 0;
```

```
}
```



```
grade[0]=10  
grade[1]=20  
grade[2]=30  
grade[3]=40  
grade[4]=50
```




배열 선언 예제



```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 5
```

```
int main(void)
{
```

```
    int i;
    int grade[SIZE];
```

```
    for(i = 0; i < SIZE; i++)
        grade[i] = rand() % 100;
```

```
    for(i = 0; i < SIZE; i++)
        printf("grade[%d]=%d\n", i, grade[i]);
```

```
    return 0;
```

```
}
```



```
grade[0]=41
grade[1]=67
grade[2]=34
grade[3]=0
grade[4]=69
```



배열 선언 예제



```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 5
```

```
int main(void)
{
    int i;
    int grade[SIZE];
    printf("5명의 점수를 입력하시오\n");
```

```
    for(i = 0; i < SIZE; i++)
        scanf("%d", &grade[i]);
```

```
    for(i = 0; i < SIZE; i++)
        printf("grade[%d]=%d\n", i, grade[i]);
    return 0;
```

```
}
```



```
grade[0]=41
grade[1]=67
grade[2]=34
grade[3]=0
grade[4]=69
```



배열 선언 예제



```
#include <stdio.h>
#define STUDENTS 5
int main(void)
{
    int grade[STUDENTS];
    int sum = 0;
    int i, average;
    for(i = 0; i < STUDENTS; i++)
    {
        printf("학생들의 성적을 입력하시오: ");
        scanf("%d", &grade[i]);
    }
    for(i = 0; i < STUDENTS; i++)
        sum += grade[i];
    average = sum / STUDENTS;
    printf("성적 평균= %d\n", average);

    return 0;
}
```



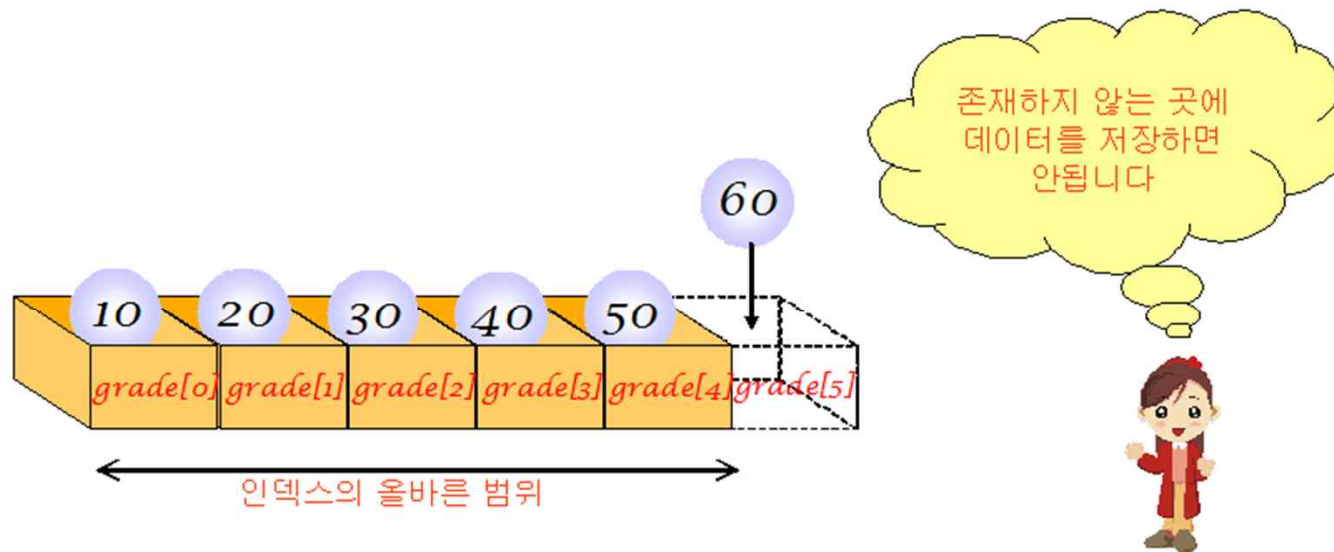
학생들의 성적을 입력하시오: 10
학생들의 성적을 입력하시오: 20
학생들의 성적을 입력하시오: 30
학생들의 성적을 입력하시오: 40
학생들의 성적을 입력하시오: 50
성적 평균 = 30



잘못된 인덱스 문제

- 인덱스가 배열의 크기를 벗어나게 되면 프로그램에 치명적인 오류를 발생시킨다.
- **C**에서는 프로그래머가 인덱스가 범위를 벗어나지 않았는지를 확인하고 책임을 져야 한다.

```
int grade[5];  
...  
grade[5] = 60;    // 치명적인 오류!
```





잘못된 인덱스 예제



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

```
int main(void)
```

```
{
```

```
    int grade[5];
```

```
    int i;
```

```
    grade[0]=10;
```

```
    grade[1]=20;
```

```
    grade[2]=30;
```

```
    grade[3]=40;
```

```
    grade[4]=50;
```

```
    grade[5]=60;
```

```
    for(i = 0; i <= 5; i++)
```

```
        printf("grade[%d]=%d\n", i, grade[i]);
```

```
    return 0;
```

```
}
```



시스템에 심각한 오류가 발생할 수도 있다.



중간 점검

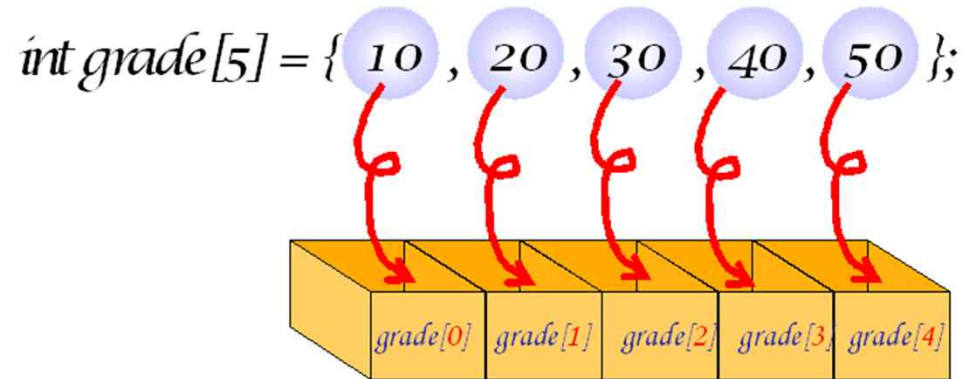
1. n 개의 원소를 가지는 배열의 경우, 첫 번째 원소의 인덱스는 무엇인가?
2. n 개의 원소를 가지는 배열의 경우, 마지막 원소의 인덱스는 무엇인가?
3. 범위를 벗어나는 인덱스를 사용하면 어떻게 되는가? 즉 `int a[10];`과 같이 선언된 배열이 있는 경우, `a[10]`에 6을 대입하면 어떻게 되는가?



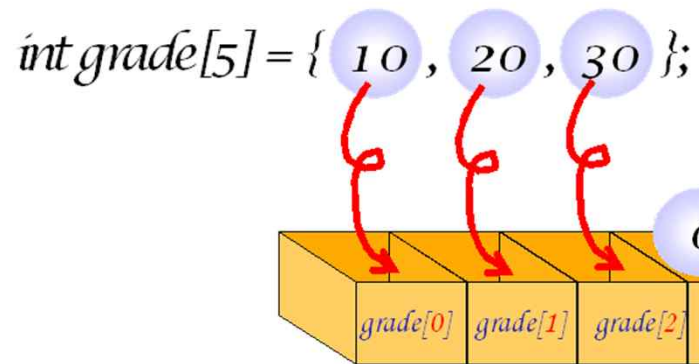


배열의 초기화

- `int grade[5] = { 10,20,30,40,50 };`



- `int grade[5] = { 10,20,30 };`



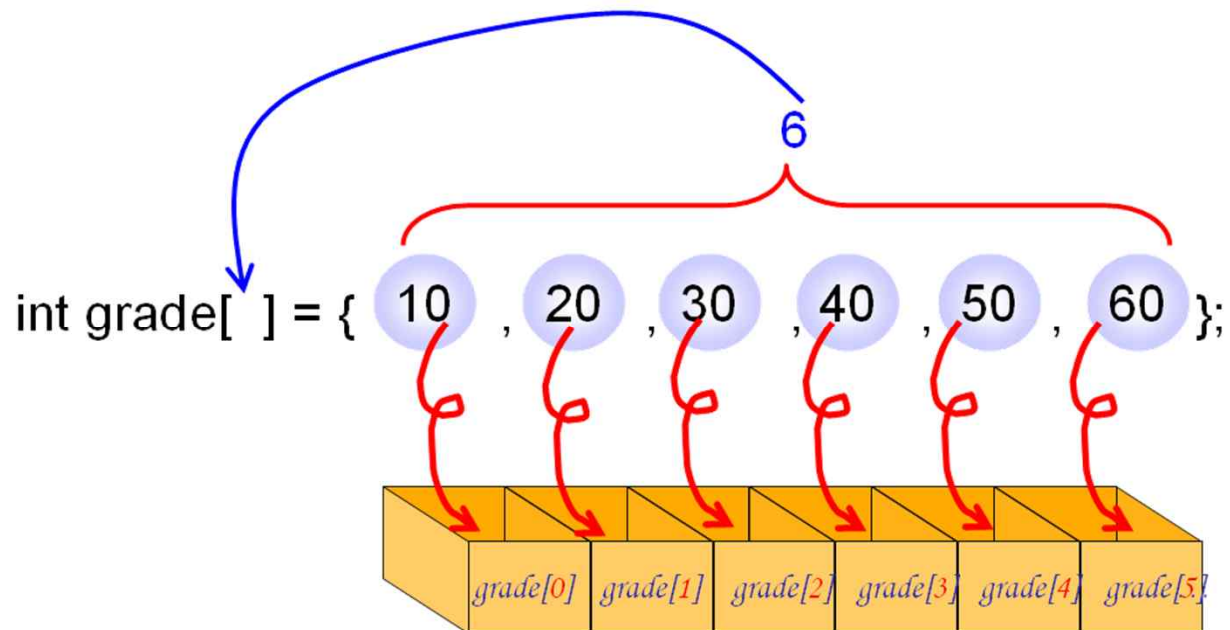
초기값을 일부만
주면 나머지 원소
들은 0으로 초기화
됩니다.





배열의 초기화

- 배열의 크기가 주어지지 않으면 자동적으로 초기값의 개수만큼이 배열의 크기로 잡힌다.





배열 초기화 예제



```
#include <stdio.h>
int main(void)
{
    int grade[5] = { 31, 63, 62, 87, 14 };
    int i;

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

    return 0;
}
```



```
grade[0] = 31
grade[1] = 63
grade[2] = 62
grade[3] = 87
grade[4] = 14
```



배열 초기화 예제



```
#include <stdio.h>
int main(void)
{
    int grade[5] = { 31, 63 };
    int i;

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

    return 0;
}
```



```
grade[0] = 31
grade[1] = 63
grade[2] = 0
grade[3] = 0
grade[4] = 0
```



배열 초기화 예제



```
#include <stdio.h>
int main(void)
{
    int grade[5] ;
    int i;

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

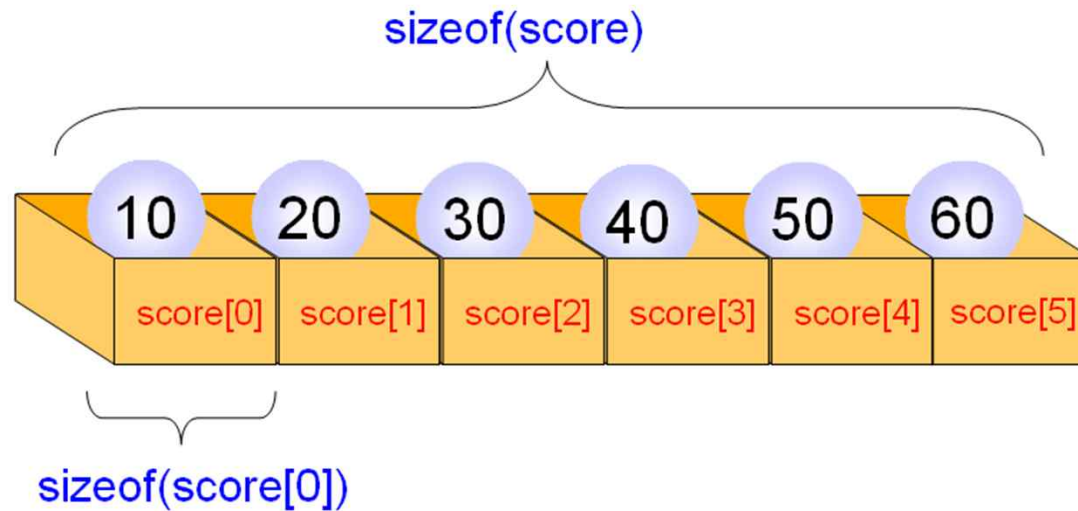
    return 0;
}
```



```
grade[0]=4206620
grade[1]=0
grade[2]=4206636
grade[3]=2018779649
grade[4]=1
```



배열 원소의 개수 계산



```
int grade[] = { 1, 2, 3, 4, 5, 6 };
```

```
int i, size;
```

```
size = sizeof(grade) / sizeof(grade[0]);
```

배열 원소 개수 자동 계산

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

```
    printf("%d ", grade[i]);
```



중간 점검

1. 배열 $a[6]$ 의 원소를 1, 2, 3, 4, 5, 6으로 초기화하는 문장을 작성하라.
2. 배열의 초기화에서 초기값이 개수가 배열 원소의 개수보다 적은 경우에는 어떻게 되는가? 또 반대로 많은 경우에는 어떻게 되는가?
3. 배열의 크기를 주지 않고 초기값의 개수로 배열의 크기를 결정할 수 있는가?





예제

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

int main(void)
{
    int grade[STUDENTS] = { 30, 20, 10, 40, 50 };
    int i, s;

    for(i = 0; i < STUDENTS; i++)
    {
        printf("번호 %d: ", i);
        for(s = 0; s < grade[i]; s++)
            printf("*");
        printf("\n");
    }

    return 0;
}
```



```
번호 0: *****
번호 1: *****
번호 2: *****
번호 3: *****
번호 4: *****
```



최소값 탐색



```
#include <stdio.h>
#define SIZE 5
```

```
int main(void)
{
    int grade[SIZE];
    int i, min;

    for(i = 0; i < SIZE; i++)
    {
        printf("성적을 입력하시오: ");
        scanf("%d", &grade[i]);
    }

    min = grade[0];

    for(i = 1; i < SIZE; i++)
    {
        if( grade[i] < min )
            min = grade[i];
    }

    printf("최소값은 %d입니다.\n", min);
    return 0;
}
```

성적을 입력하시오: 50
성적을 입력하시오: 40
성적을 입력하시오: 30
성적을 입력하시오: 20
성적을 입력하시오: 10
최소값은 10입니다.



빈도 계산

```
#include <stdio.h>
#define SIZE 101

int main(void)
{
    int freq[SIZE];
    int i, score;

    for(i = 0; i < SIZE; i++)
        freq[i] = 0;

    while(1)
    {
        printf("숫자를 입력하시오(종료-1): ");
        scanf("%d", &score);
        if (score < 0) break;
        freq[score]++;
    }

    printf("값 빈도\n");

    for(i = 0; i < SIZE; i++)
        printf("%3d %3d \n", i, freq[i]);

    return 0;
}
```



숫자를 입력하시오(종료-1): 0
숫자를 입력하시오(종료-1): 1
숫자를 입력하시오(종료-1): 99
숫자를 입력하시오(종료-1): 100
숫자를 입력하시오(종료-1): 100
숫자를 입력하시오(종료-1): -1

값 빈도

| | |
|-----|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 0 |
| ... | |
| 98 | 0 |
| 99 | 1 |
| 100 | 2 |



이번 장에서 학습할 내용

- 반복의 개념 이해
- 배열의 개념
- 배열의 선언과 초기화
- 일차원 배열
- 배열의 응용
- 정렬과 탐색
- 다차원 배열

배열을 사용하면
한 번에 여러
개의 값을 저장할
수 있는 공간을
할당받을 수
있다.



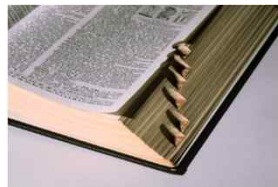


정렬이란?

- 정렬은 물건을 크기순으로 오름차순이나 내림차순으로 나열하는 것
- 정렬은 컴퓨터 공학분야에서 가장 기본적인 알고리즘중의 하나
- 정렬은 자료 탐색에 있어서 필수적이다.



(예) 만약 사전에서 단어들이 정렬이 안되어 있다면?



| 비교 | 제조사 | 모델명 | 요약설명 | 최저가 | 업체수 | 출시 |
|--------------------------|--------|----------------|--|---------|-----|-----|
| <input type="checkbox"/> | ROLLEI | D-41com | 410만화소(0.56")/1.8"LCD/3배줌/연사/CF카드 | 320,000 | 4 | 02년 |
| <input type="checkbox"/> | 카시오 | QV-R40 | 413만화소(0.56")/1.6"LCD/3배줌/동영상/히스토그램/앨범기능/SD,MMC카드 | 344,000 | 73 | 03년 |
| <input type="checkbox"/> | 파나소닉 | DMC-LC43 | 423만화소(0.4")/1.5"LCD/3배줌/동영상+녹음/연사/SD,MMC카드 | 348,000 | 36 | 03년 |
| <input type="checkbox"/> | 현대 | DC-4311 | 400만화소(0.56")/1.6"LCD/3배줌/동영상/SD,MMC카드 | 350,000 | 7 | 03년 |
| <input type="checkbox"/> | 삼성테크윈 | Digimax420 | 410만화소(0.56")/1.5"LCD/3배줌/동영상+녹음/음성메모/한글/SD카드 | 353,000 | 47 | 03년 |
| <input type="checkbox"/> | 니콘 | Coolpix4300 | 413만화소(0.56")/1.5"LCD/3배줌/동영상/연사/CF카드 | 356,800 | 79 | 02년 |
| <input type="checkbox"/> | 올림푸스 | 뮤-20 Digital | 423만화소(0.4")/1.5"LCD/3배줌/동영상/연사/생활방수/xD카드 | 359,000 | 63 | 03년 |
| <input type="checkbox"/> | 코닥 | LS-443(Dock포함) | 420만화소/1.8"LCD/3배줌/동영상+녹음/SD,MMC카드/Dock시스템 | 365,000 | 39 | 02년 |
| <input type="checkbox"/> | 올림푸스 | C-450Z | 423만화소(0.4")/1.8"LCD/3배줌/동영상/연사/xD카드 | 366,000 | 98 | 03년 |
| <input type="checkbox"/> | 올림푸스 | X-1 | 430만화소/1.5"LCD/3배줌/동영상/연사/xD카드 | 367,000 | 19 | 03년 |
| <input type="checkbox"/> | 미놀타 | DIMAGE-F100 | 413만화소(0.56")/1.5"LCD/3배줌/동영상+녹음/음성메모/동체추적AF/연사/SD,MMC카드 | 373,000 | 18 | 02년 |
| <input type="checkbox"/> | 삼성테크윈 | Digimax410 | 410만화소(0.56")/1.6"LCD/3배줌/동영상+녹음/음성메모/한글/CF카드 | 374,000 | 4 | 02년 |



선택정렬(selection sort)

- 선택정렬(selection sort): 정렬이 안된 숫자들중에서 최소값을 선택하여 배열의 첫번째 요소와 교환





선택 정렬 1/2

```
#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;
```



선택 정렬 2/2

```
for(i = 0; i < SIZE-1; i++)  
{  
    least = i; // i번째 값을 최소값으로 가정  
  
    for(j = i + 1; j < SIZE; j++) // 최소값 탐색  
        if(list[j] < list[least])  
            least = j;  
    // i번째 원소와 least 위치의 원소를 교환  
    temp = list[i];  
    list[i] = list[least];  
    list[least] = temp;  
}  
for(i = 0; i < SIZE; i++)  
    printf("%d ", list[i]);  
printf("\n");  
return 0;  
}
```



0 1 2 3 4 5 6 7 8 9

계속하려면 아무 키나 누르십시오 ...



탐색

- 탐색(search)은 컴퓨터가 가장 많이 하는 작업
- (예) 인터넷 탐색

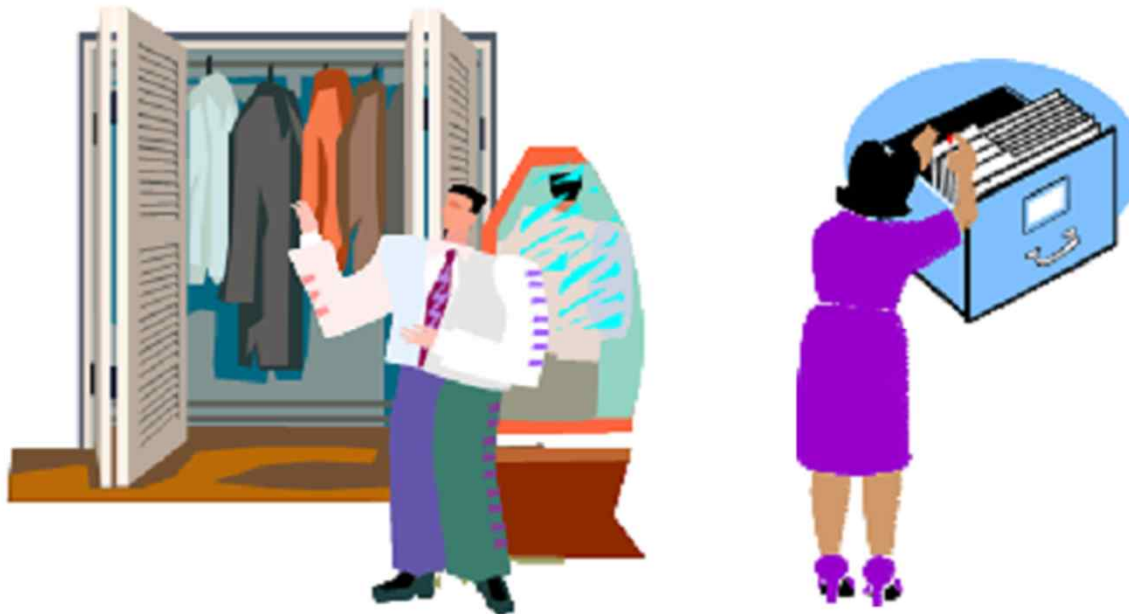


그림 8.12 탐색의 예



순차 탐색

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

```
int main(void)
{
```

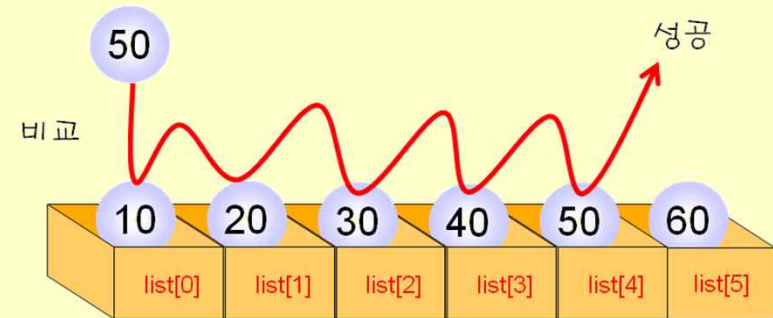
```
    int key, i;
    int list[SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

```
    printf("탐색 할 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;
```


```
}
```



탐색할 값을 입력하시오:7
탐색 성공 인덱스 = 6
탐색 종료



이번 장에서 학습할 내용

- 
- 반복의 개념 이해
 - 배열의 개념
 - 배열의 선언과 초기화
 - 일차원 배열
 - 배열의 응용
 - 정렬과 탐색
 - 다차원 배열

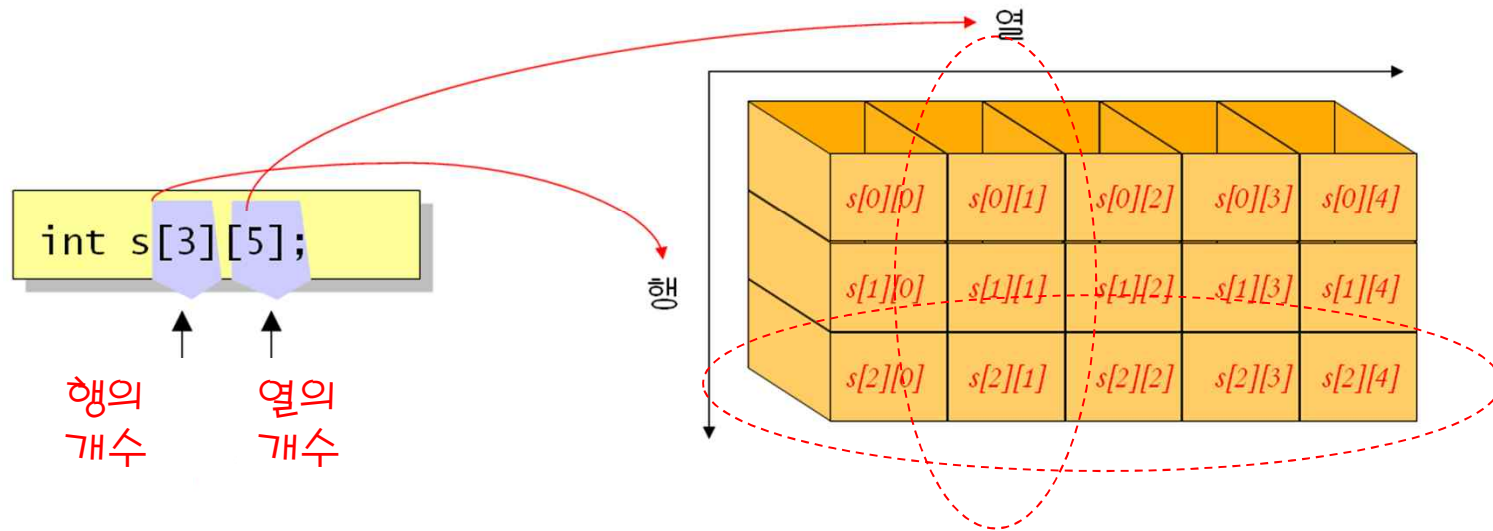
배열을 사용하면
한 번에 여러
개의 값을 저장할
수 있는 공간을
할당받을 수
있다.





2차원 배열

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





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

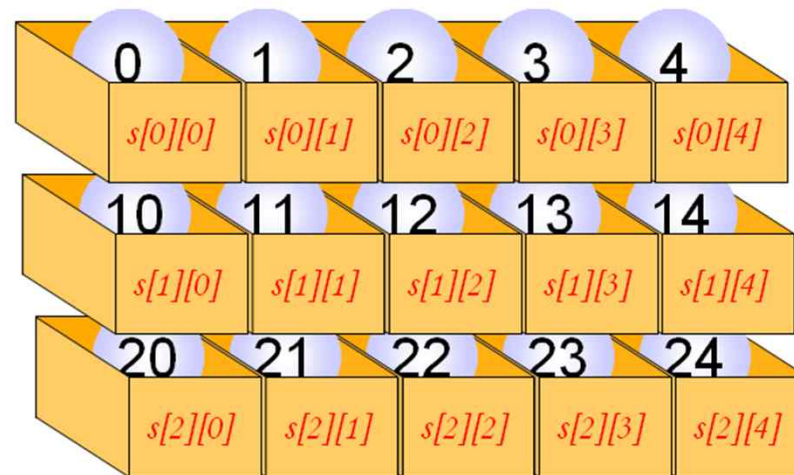
    for(i=0; i<3; i++)
        for(j=0; j<5; j++)
            s[i][j] = value++;

    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  
};
```





2차원 배열의 초기화

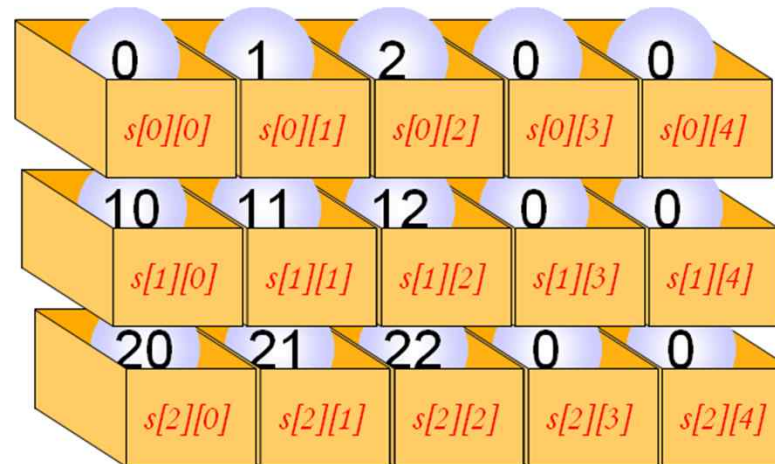
```
int s[ ][5] = {  
    { 0, 1, 2, 3, 4}, // 첫 번째 행의 원소들의 초기값  
    { 10, 11, 12, 13, 14}, // 두 번째 행의 원소들의 초기값  
    { 20, 21, 22, 23, 24}, // 세 번째 행의 원소들의 초기값  
};
```





2차원 배열의 초기화

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





다차원 배열 예제

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

```
int main(void)
{
```

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

```
    int clas, student, total, subtotal;
```

```
// 첫번째 행의 원소들의 초기값
// 두번째 행의 원소들의 초기값
// 세번째 행의 원소들의 초기값
```



다차원 배열 예제

```
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;
}
```



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



다차원 배열을 이용한 행렬의 표현

$$A = \begin{bmatrix} 2 & 3 & 0 \\ 8 & 9 & 1 \\ 7 & 0 & 5 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 7 & 0 & 0 \\ 9 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 \\ 6 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \end{bmatrix}$$



```
#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];
```




다차원 배열을 이용한 행렬의 표현

```
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
```



중간 점검

1. 다차원 배열 `int a[3][2][10]`에는 몇개의 원소가 존재하는가?
2. 다차원 배열 `int a[3][2][10]`의 모든 요소를 0으로 초기화하는 문장을 작성하시오.





Q & A

