

07

배열

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

자료형

배열이름



변수

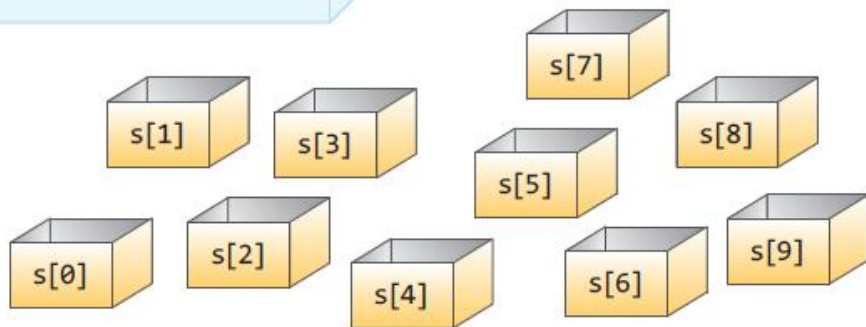


아파트

1. 배열의 이해

- 학생이 10명이 있고 이들의 평균 성적을 계산한다고 가정하자.

```
// 일반 변수 사용  
int s0;  
int s1;  
...  
int s9;
```



```
// 배열 사용  
int s[10];
```

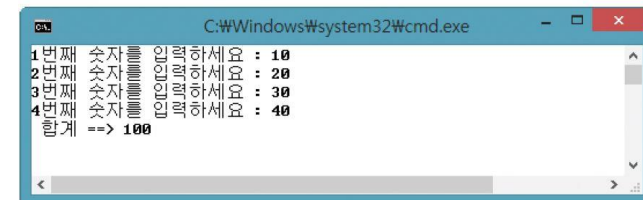


1. 배열의 이해 – 변수값 여러 개를 선언하여 출력하는 예

4 |

```
01 #include <stdio.h>
02
03 int main( )
04 {
05     int a, b, c, d;      ----각 입력 변수와 함께 변수를 선언한다.
06     int hap;
07
08     printf("1번째 숫자를 입력하세요 : ");
09     scanf_s("%d", &a);   ----변수에 숫자를 입력한다.
10     printf("2번째 숫자를 입력하세요 : ");
11     scanf_s("%d", &b);   ----변수에 숫자를 입력한다.
12     printf("3번째 숫자를 입력하세요 : ");
13     scanf_s("%d", &c);   ----변수에 숫자를 입력한다.
14     printf("4번째 숫자를 입력하세요 : ");
15     scanf_s("%d", &d);   ----변수에 숫자를 입력한다.
16
17     hap = a+b+c+d;      ----입력받은 숫자의 합계 결과이다.
18
19     printf(" 합계 == > %d \n", hap);
20 }
```

실행 결과 ▼



```
C:\Windows\system32\cmd.exe
1번째 숫자를 입력하세요 : 10
2번째 숫자를 입력하세요 : 20
3번째 숫자를 입력하세요 : 30
4번째 숫자를 입력하세요 : 40
합계 ==> 100
```

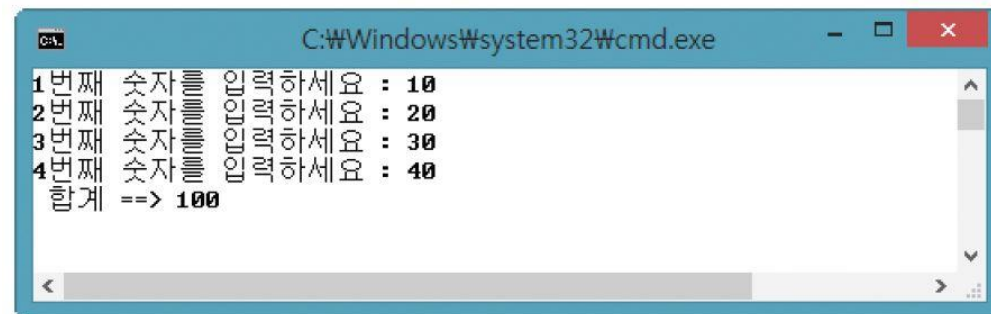
1. 배열의 이해 - 배열에 값을 선언하여 출력하는 예

```

01 #include <stdio.h>
02
03 int main( )
04 {
05     int aa[4];          ---정수형 배열을 선언한다.
06     int hap;
07
08     printf("1번째 숫자를 입력하세요 : ");
09     scanf_s("%d", &aa[0]); ---aa[0]에 숫자를 입력한다.
10     printf("2번째 숫자를 입력하세요 : ");
11     scanf_s("%d", &aa[1]); ---aa[1]에 숫자를 입력한다.
12     printf("3번째 숫자를 입력하세요 : ");
13     scanf_s("%d", &aa[2]); ---aa[2]에 숫자를 입력한다.
14     printf("4번째 숫자를 입력하세요 : ");
15     scanf_s("%d", &aa[3]); ---aa[3]에 숫자를 입력한다.
16
17     hap = aa[0] + aa[1] + aa[2] + aa[3]; ---입력받은 배열에 저장된 숫자 합계 결과이다.
18
19     printf(" 합계 == > %d \n", hap);
20 }

```

실행 결과 ▼



```

C:\Windows\system32\cmd.exe
1번째 숫자를 입력하세요 : 10
2번째 숫자를 입력하세요 : 20
3번째 숫자를 입력하세요 : 30
4번째 숫자를 입력하세요 : 40
합계 ==> 100

```

배열의 선언

형식 자료형 배열이름[배열크기];

예 `int scores[10];`

설명 지정된 자료형과 크기를 가지는 배열을 생성한다.

```
int scores[10];
```

자료형

배열이름

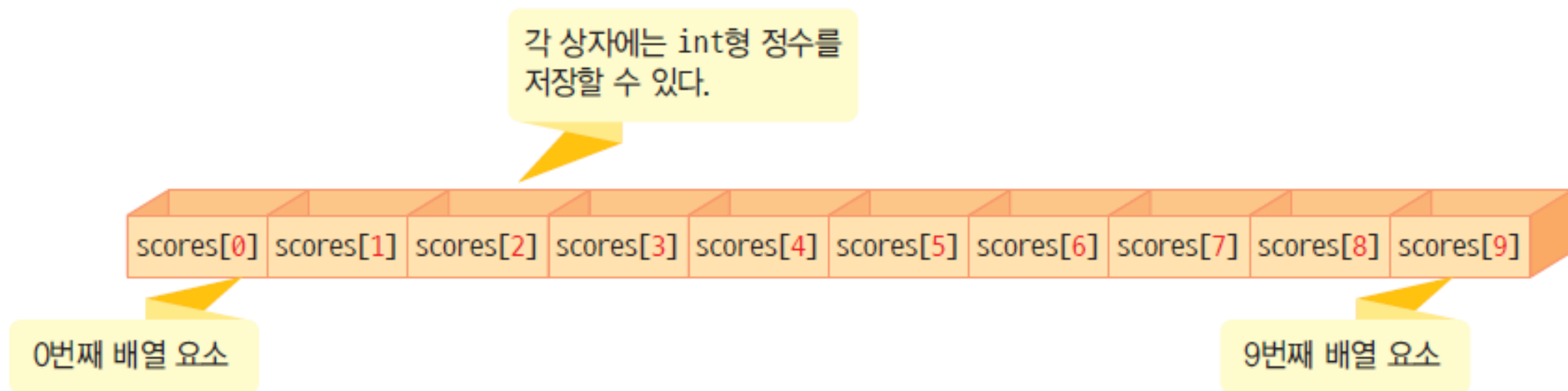
배열크기

- 배열의 모든 요소는 **같은 형태의 자료형** (Homogeneous Data Type)

`char char_set[26];` `double math_score[12];`

- **인덱스(index):** 배열 요소의 번호

```
int scores[10];
```



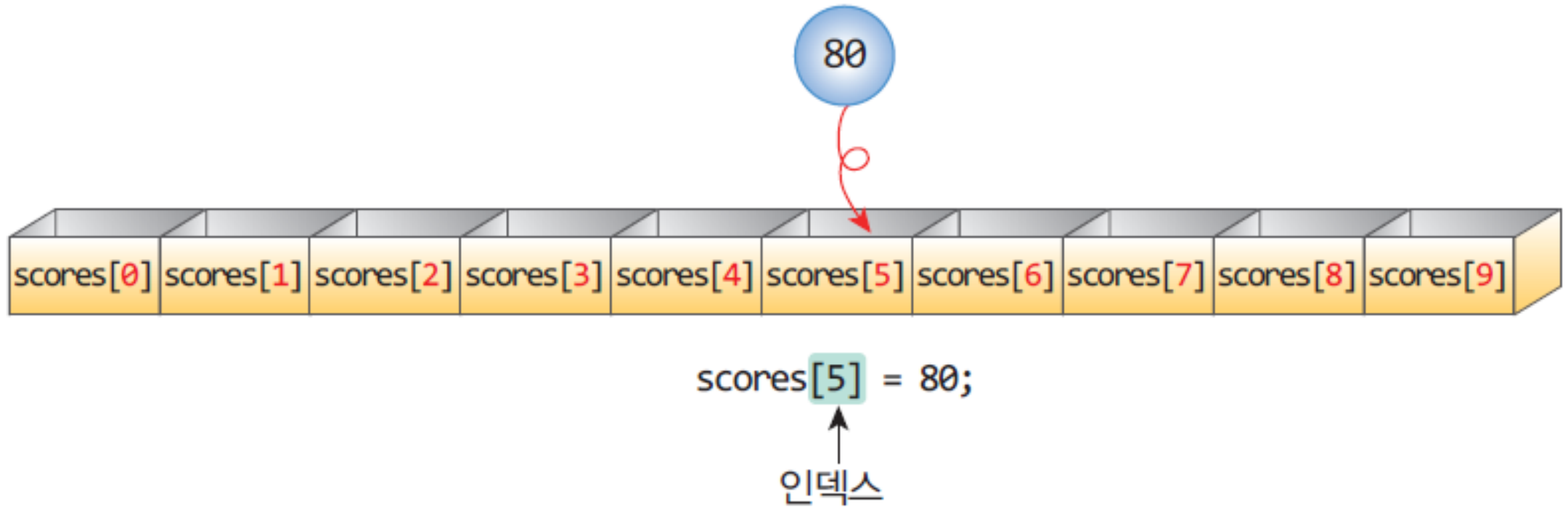
- **C 언어의 배열 인덱스는 0부터 시작**

- 인덱스는 0부터 9
- scores[10]은 존재하지 않음

```
int scores[60]; // 60개의 int형 값을 가지는 배열 scores
```

```
float cost[12]; // 12개의 float형 값을 가지는 배열 cost
```

```
char name[50]; // 50개의 char형 값을 가지는 배열 name
```

```
scores[0] = 80;
```

// 0번째 요소에 80을 저장한다.

```
scores[3] = scores[2];
```

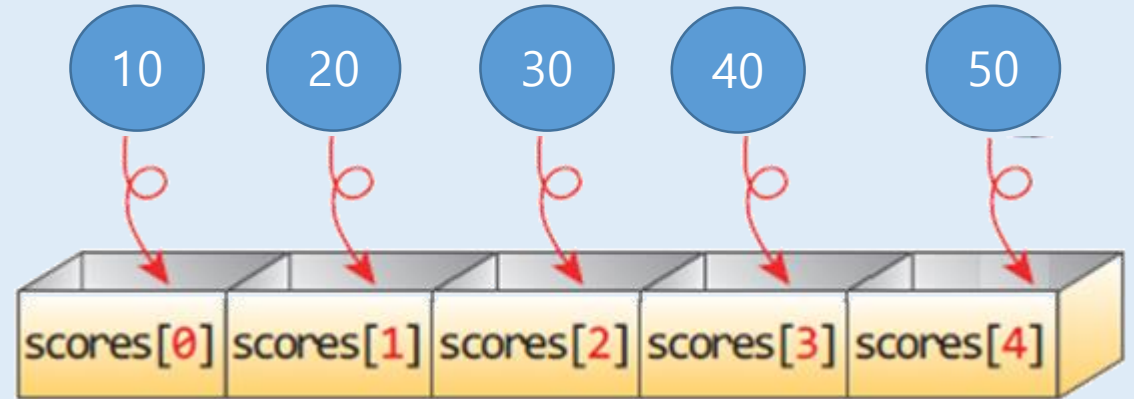
// 2번째 요소를 3번째 요소로 복사한다.

```
scores[k] = 100;
```

// k번째 요소에 100을 저장한다.

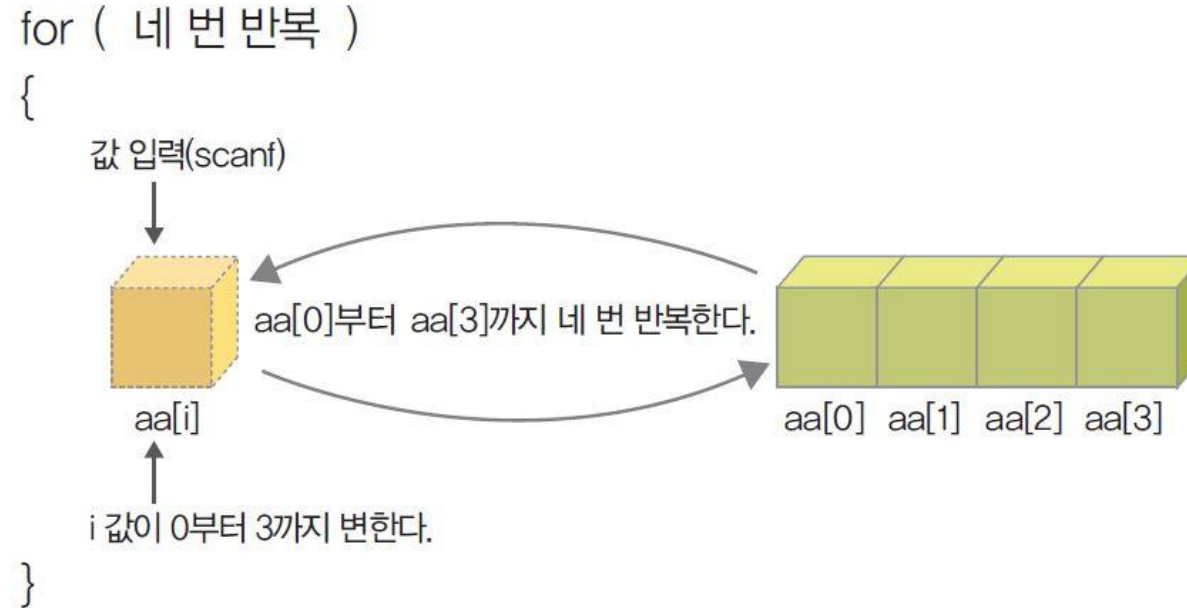
배열 선언 예제

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int i;
06     int scores[5];
07
08     scores[0] = 10;
09     scores[1] = 20;
10     scores[2] = 30;
11     scores[3] = 40;
12     scores[4] = 50;
13
14     for(i=0; i<5; i++)
15         printf("scores[%d]=%d\n",i, scores[i]);
16     return 0;
17 }
```



```
C:\Windows\system32\cmd.exe
scores[0]=10
scores[1]=20
scores[2]=30
scores[3]=40
scores[4]=50
계속하려면 아무 키나 누르십시오 . . .
```

- 배열의 가장 큰 장점은 반복문을 사용하여 배열의 원소를 간편하게 처리할 수 있다



```
aa[0] = 0;  
aa[1] = 0;  
aa[2] = 0;  
aa[3] = 0;
```

```
#define SIZE 4  
...  
for(i=0 ; i<SIZE ; i++)  
    aa[i] = 0;
```

1. 배열의 이해 - for문으로 배열의 첨자를 활용하는 예

12 |

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

```
02
```

```
03 int main( )
```

```
04 {
```

```
05     int aa[4];
```

```
06     int hap=0;
```

```
07     int i;
```

```
08
```

```
09     for (i=0; i<4; i++)
```

```
10     {
```

```
11         printf("%d번째 숫자를 입력하세요 : ", i+1);
```

```
12         scanf_s("%d", &aa[i] );
```

```
13     }
```

```
14
```

```
15     hap = aa[0] + aa[1] + aa[2] + aa[3];
```

```
16
```

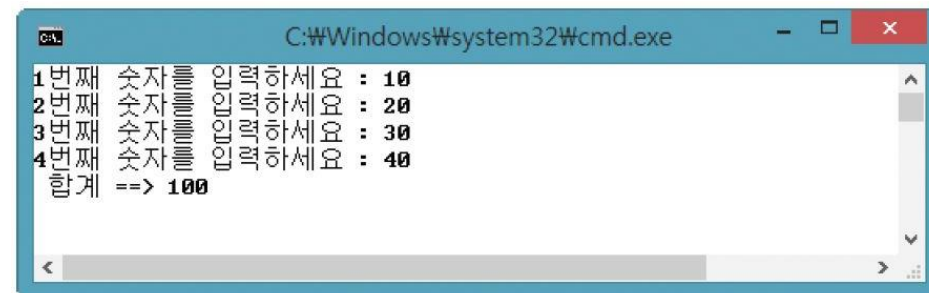
```
17     printf(" 합계 == > %d \n", hap);
```

```
18 }
```

---배열과 함께 변수, 첨자를 선언한다.

---배열 aa[0]~[3]에 숫자 4개를 입력받는다.

---배열에 저장된 숫자 4개를 더한다.



```
C:\Windows\system32\cmd.exe
1번째 숫자를 입력하세요 : 10
2번째 숫자를 입력하세요 : 20
3번째 숫자를 입력하세요 : 30
4번째 숫자를 입력하세요 : 40
합계 ==> 100
```

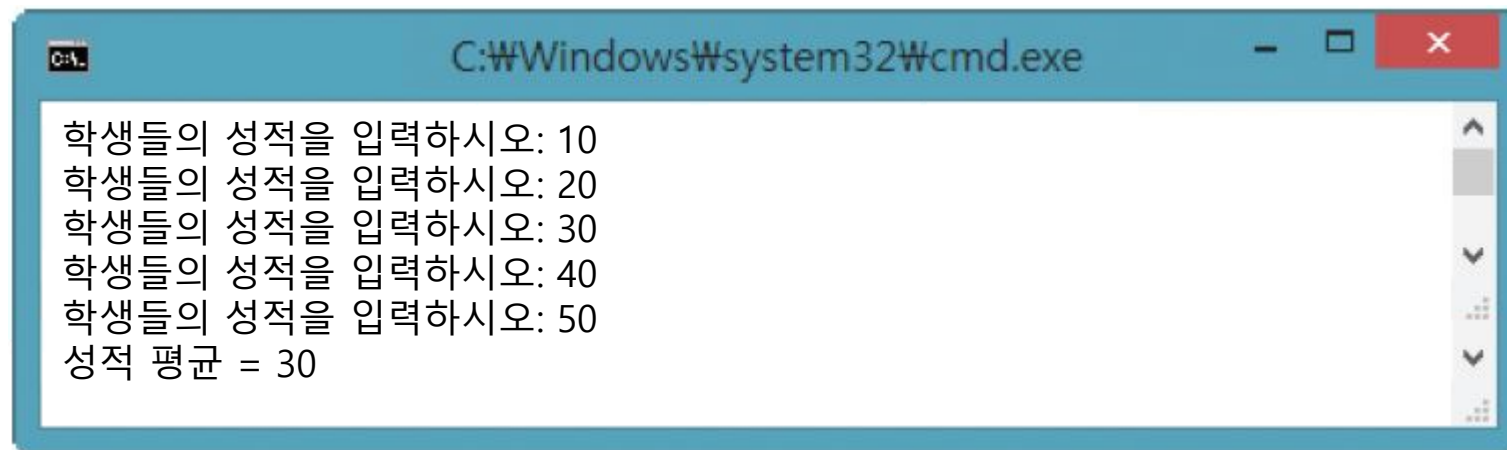
1. 배열의 이해 - for문으로 배열의 첨자를 활용하는 예

```
01 #include <stdio.h>
02
03 int main( )
04 {
05     int aa[4];
06     int hap=0;
07     int i;
08
09     for (i=0; i<4; i++)
10     {
11         printf("%d번째 숫자를 입력하세요 : ", i+1);
12         scanf_s("%d", &aa[i] );
13     }
14
15     hap = aa[0] + aa[1] + aa[2] + aa[3];
16
17     printf(" 합계 == > %d \n", hap);
18 }
```

```
01 #include <stdio.h>
02
03 int main( )
04 {
05     int aa[4];
06     int hap=0;
07     int i;
08
09     for (i=0; i<4; i++)
10     {
11         printf("%d번째 숫자를 입력하세요 : ", i+1);
12         scanf_s("%d", &aa[i] );
13         hap += aa[i];
14     }
15
16     printf(" 합계 == > %d \n", hap);
17 }
```

예제 설명 학생들의 성적을 입력받아 평균을 구하는 프로그램을 작성하시오

실행 결과



```
C:\Windows\system32\cmd.exe

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

[예제] 성적 구하기

```
01 #include <stdio.h>
02 #define STUDENTS 5
03
04 int main(void)
05 {
06     int scores[STUDENTS];
07     int sum = 0;
08     int i, average;
09     for(i = 0; i < STUDENTS; i++)
10     {
11         printf("학생들의 성적을 입력하시오: ");
12         scanf_s("%d", &scores[i]);
13     }
14     for(i = 0; i < STUDENTS; i++)
15         sum += scores[i];
16     average = sum / STUDENTS;
17     printf("성적 평균= %d\n", average);
18
19     return 0;
20 }
```

- 인덱스가 배열의 크기를 벗어나게 되면 프로그램에 치명적인 오류를 발생시킨다.

```
int score[5];
```

```
...
```

```
score[5] = 60; // 치명적인 오류!
```



[예제] 잘못된 인덱스 문제



시스템에 심각한 오류가 발생할 수도 있다.

```
01 #include <stdio.h>
02 int main(void)
03 {
04     int scores[5];
05     int i;
06
07     scores[0]=10;
08     scores[1]=20;
09     scores[2]=30;
10     scores[3]=40;
11     scores[4]=50;
12     scores[5]=60;
13
14     for(i=0; i<=5; i++)
15         printf("scores[%d]=%d\n", i, scores[i]);
16     return 0;
17 }
```

1. 배열의 초기화

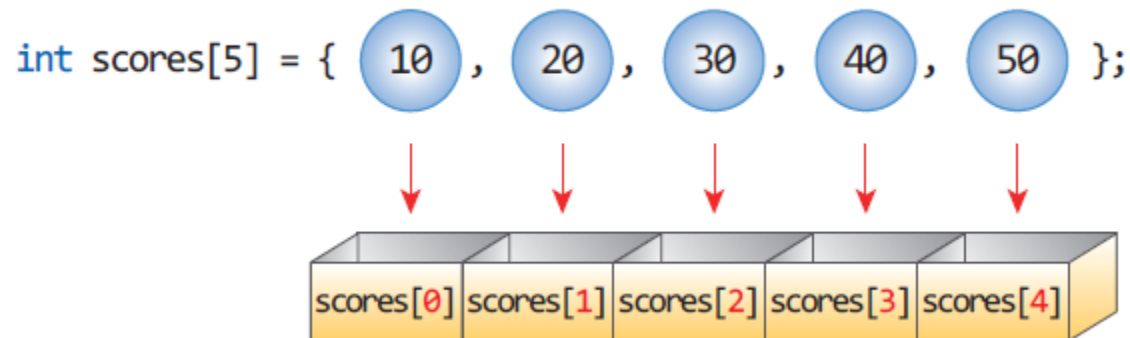
- 배열의 초기화

- 배열을 정의하는 동시에 값을 대입하는 것
- 5개의 값을 담은 배열 scores의 초기화(블록({ })과 콤마(,)를 사용)

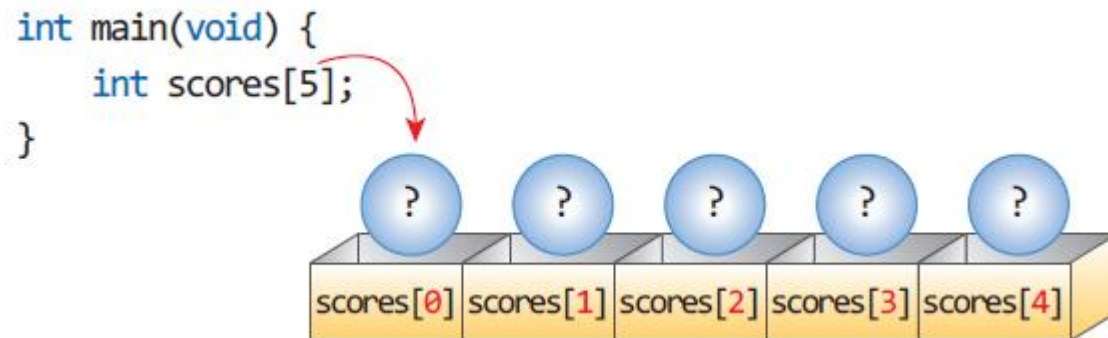
```
int a=10, b=20, c=30, d=40, e=50;
```

||

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

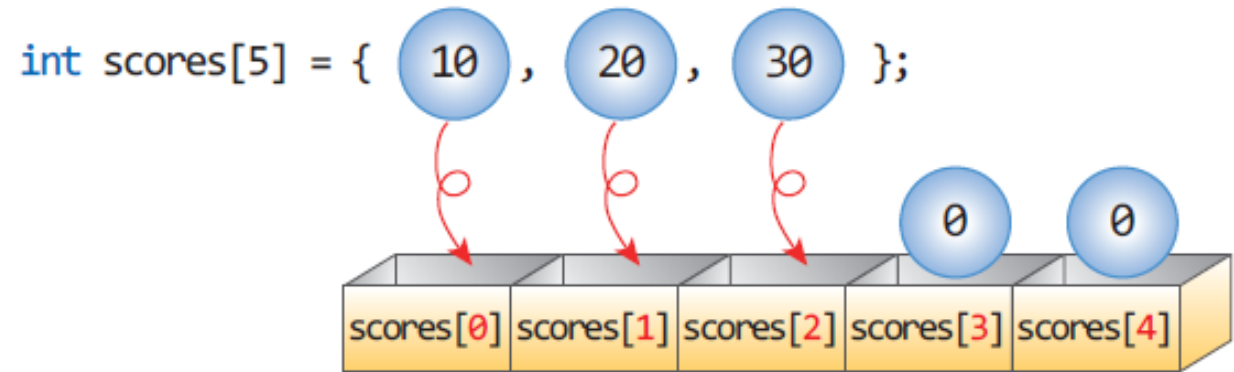


- 초기화 하지 않고 선언만 한 경우
 - 일반변수와 마찬가지로 쓰레기 값이 들어감

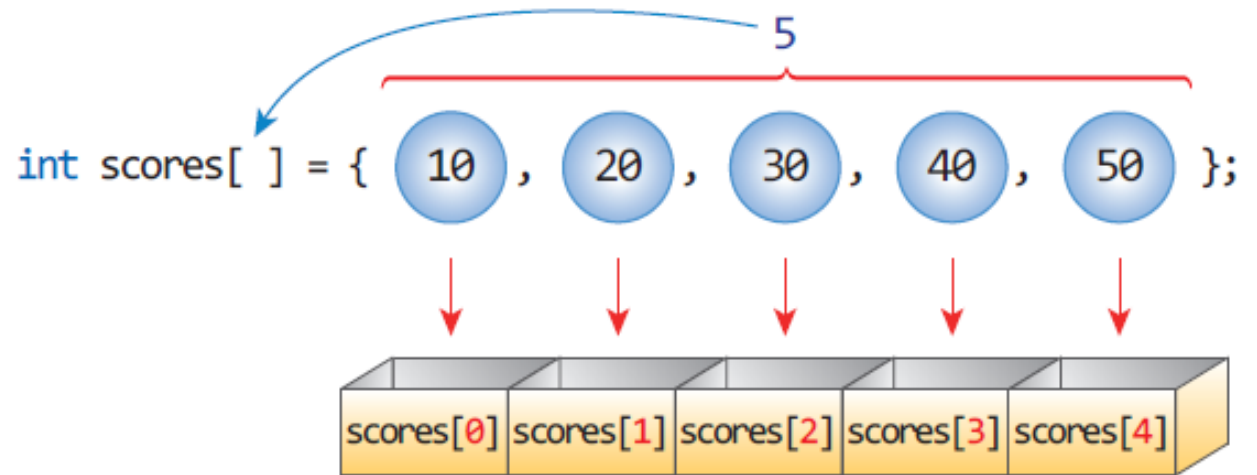


1. 배열의 이해

- 배열의 개수보다 초기화 값의 개수가 적은 경우



- 배열의 크기가 주어지지 않을 경우
 - 자동으로 초기값의 개수만큼 배열 크기가 잡힘



- 1,000개를 모두 0으로 초기화

```
int scores[1000]={0};
```

```
01 #include <stdio.h>
02 int main(void)
03 {
04     int scores[5] = { 31, 63, 62, 87, 14 };
05
06     int i;
07
08     for(i = 0; i < 5; i++)
09         printf("scores[%d] = %d\n", i, scores[i]);
10
11     return 0;
12 }
```

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

[예제] 배열 초기화 예제

21 |

```
01 #include <stdio.h>
02
03 int main( )
04 {
05     int scores[10]={95,100,98,78,65,55,78,52,97,98};
06
07     int i;
08     for (i=0; i<=10; i++){
09         printf("%d ", scores[i]);
10     }
11     printf("\n");
12     return 0;
13 }
```



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The command prompt displays the output of the C program: '95 100 98 78 65 55 78 52 97 98'.

```
01 #include <stdio.h>
02 int main(void)
03 {
04     int scores[] = { 31, 63, 62, 87, 14 };
05
06     int i;
07
08     for(i = 0; i < 5; i++)
09         printf("scores[%d] = %d\n", i, scores[i]);
10
11     return 0;
12 }
```

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

```
01 #include <stdio.h>
02 int main(void)
03 {
04     int scores[5] = { 31, 63 };
05
06     int i;
07
08     for(i = 0; i < 5; i++)
09         printf("scores[%d] = %d\n", i, scores[i]);
10
11     return 0;
12 }
```

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

```
01 #include <stdio.h>
02 int main(void)
03 {
04     int scores[5] = {0};
05
06     int i;
07
08     for(i = 0; i < 5; i++)
09         printf("scores[%d] = %d\n", i, scores[i]);
10
11     return 0;
12 }
```

```
scores[0] = 0
scores[1] = 0
scores[2] = 0
scores[3] = 0
scores[4] = 0
```

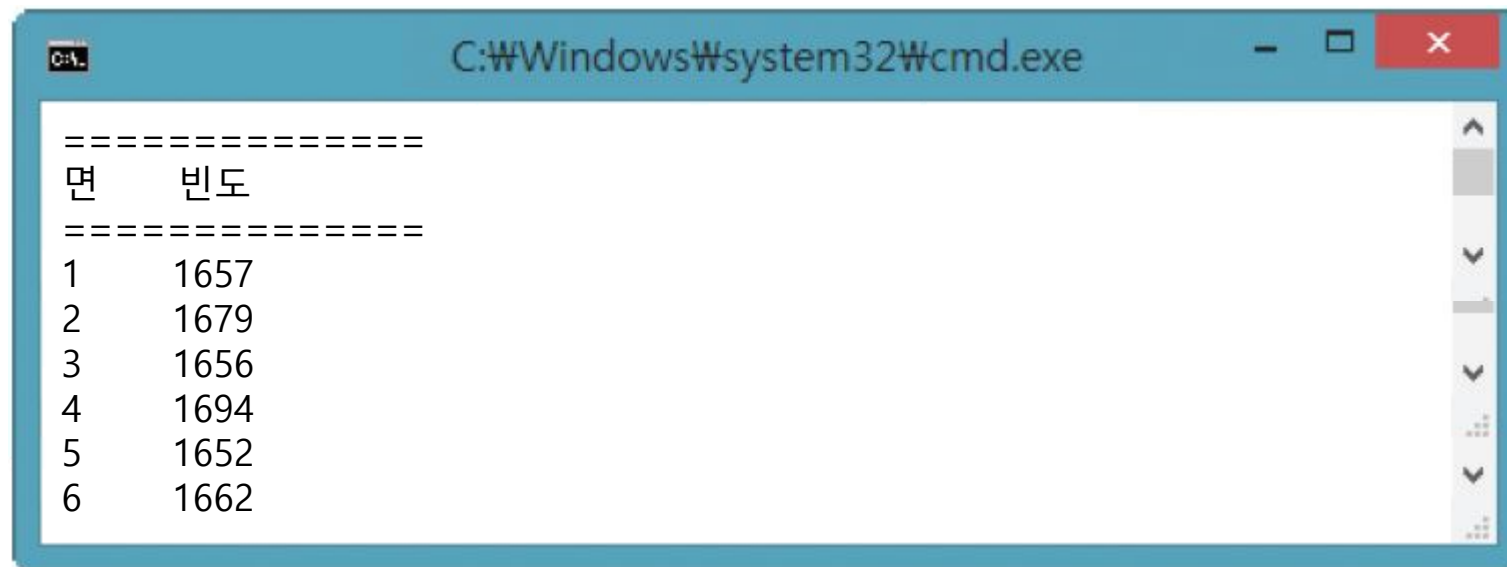


```
01 #include <stdio.h>
02 int main(void)
03 {
04     int scores[5];
05
06     int i;
07
08     for(i = 0; i < 5; i++)
09         printf("scores[%d] = %d\n", i, scores[i]);
10
11     return 0;
12 }
```

```
scores[0] = 4206620
scores[1] = 0
scores[2] = 4206636
scores[3] = 201877964
scores[4] = 0
```

예제 설명 주사위를 10000번 던져서 각 면이 나오는 횟수를 출력하여 보자.

실행 결과



```
C:\Windows\system32\cmd.exe

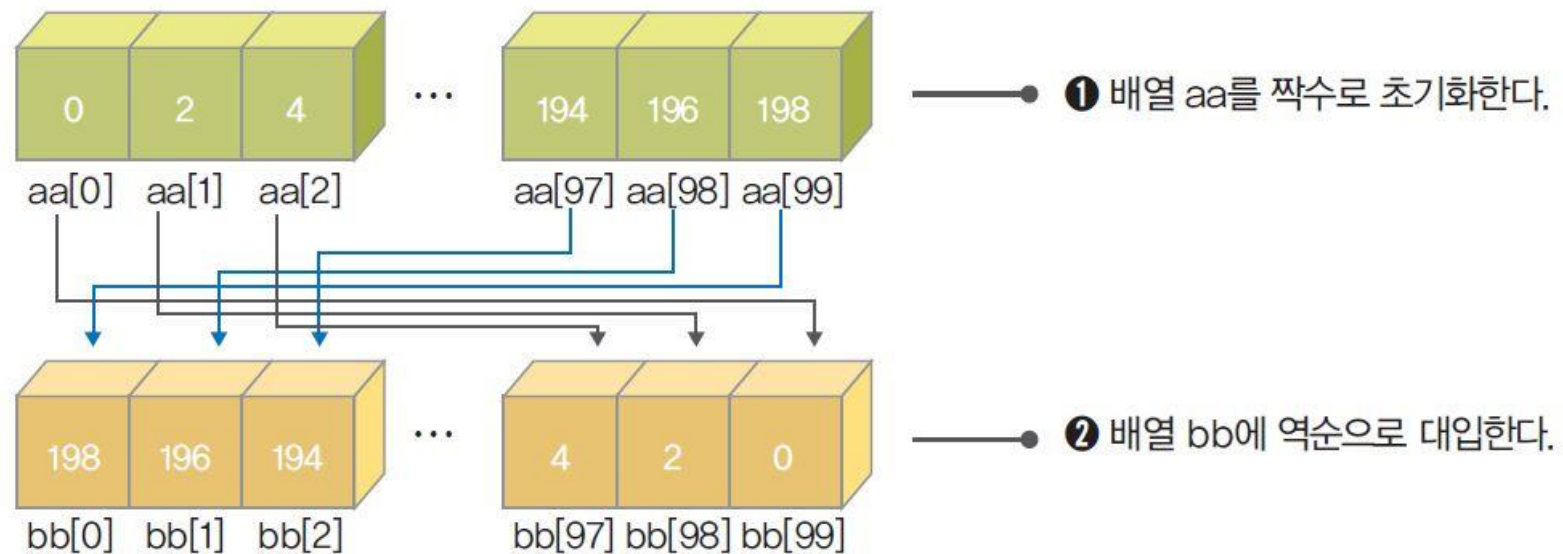
=====
면    빈도
=====
1      1657
2      1679
3      1656
4      1694
5      1652
6      1662
```

[예제] 주사위 던지기

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 #define SIZE 6
04
05 int main(void)
06 {
07     int freq[SIZE] = {0}; //배열을 반드시 초기화 해야한다.
08     int i;
09     srand((unsigned)time(NULL));
10
11     for(i=0; i<10000; i++)
12         freq[rand()%6]++;
13
14     printf("=====\n");
15     printf(" 면      빈도\n");
16     printf("=====\n");
17
18     for(i=0; i<SIZE; i++)
19         printf("%4d      %4d\n", i+1, freq[i]);
20
21     return 0;
22 }
```

● for문을 사용한 배열 초기화

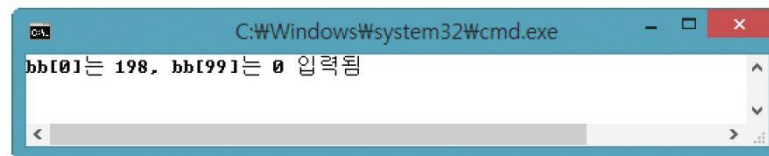
- aa[100]에 2의 배수로 초기화하기
- bb[100]에 역순으로 넣기



1. 배열의 이해 - 배열의 초기화 예 ②

```
01 #include <stdio.h>
02
03 int main( )
04 {
05     int aa[100], bb[100]; ----배열 aa와 bb를 선언한다.
06     int i;
07
08     for (i=0; i<100; i++) ----배열 aa[0], aa[1], aa[2], ..., aa[99]에
09     {                                0, 2, 4, ..., 198이 대입된다.
10         aa[i] = i * 2;
11     }
12
13     for (i=0; i<100; i++) ----배열 bb[0]에 aa[99]를 대입하는 방식으로
14     {                                값이 역순으로 대입된다.
15         bb[i] = aa[99-i];
16     }
17
18     printf("bb[0]는 %d, bb[99]는 %d 입력됨\n", bb[0], bb[99]);
19 }
```

실행 결과 ▼



1. 배열의 이해

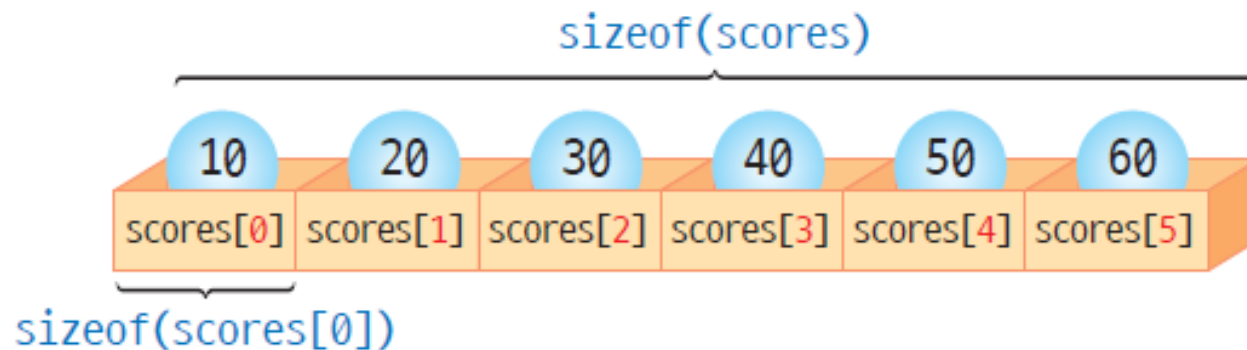
- 배열의 크기 알아내기
 - `sizeof()` 함수 사용

배열의 크기(요소의 개수) = `sizeof(전체 배열 이름) / sizeof(배열의 데이터 형식);`

- `int scores[6];` 배열의 크기 알아내기

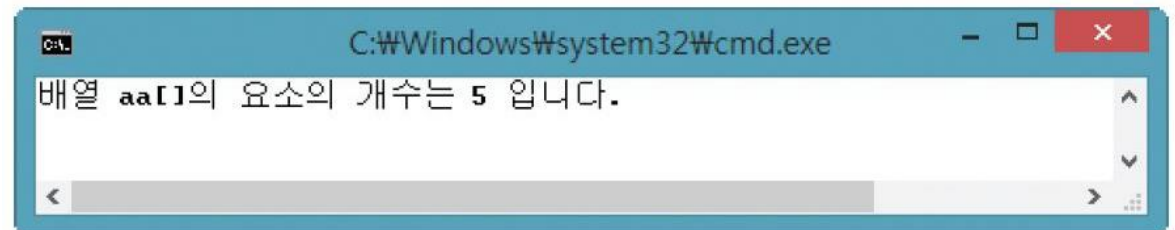
배열의 크기(요소의 개수) = `sizeof(scores) / sizeof(int);`

- ① `scores` 배열이 메모리에서 차지하는 크기(4바이트×6개=24바이트)를 알아낸다.
- ② 선언한 배열의 데이터 형식 크기(4바이트)로 나눈다.



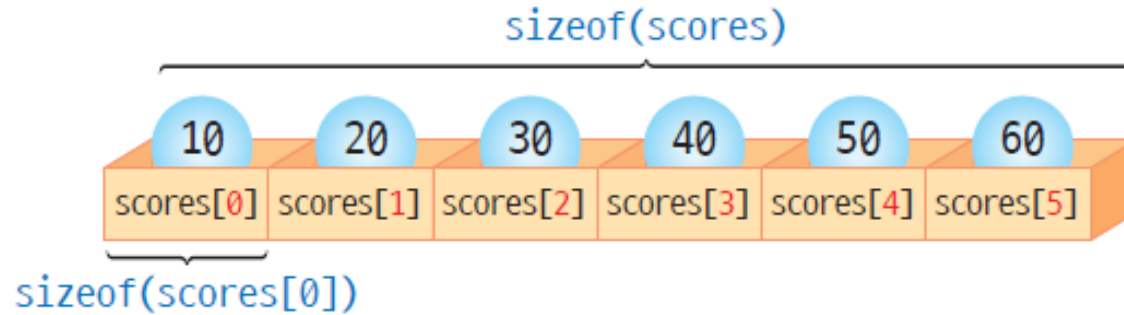
1. 배열의 이해 - 배열의 크기를 계산하는 예

```
01 #include <stdio.h>
02
03 int main( )
04 {
05     int scores[] = {10, 20, 30, 40, 50};    ---배열을 선언한다. 변수의 개수를 지정하지 않았다.
06     int count;                               ---배열 크기를 저장할 변수이다.
07
08     count = sizeof(scores) / sizeof(int);    ---배열 크기를 계산한다.
09
10     printf("배열 scores[]의 요소의 개수는 %d 입니다.\n", count);
11 }
```



A screenshot of a Windows Command Prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The command prompt displays the output of the program: '배열 aa[]의 요소의 개수는 5 입니다.' (The number of elements in array aa[] is 5).

배열 원소의 개수 계산



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

```
int main( )
```

```
{
```

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

```
    int i, size;
```

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

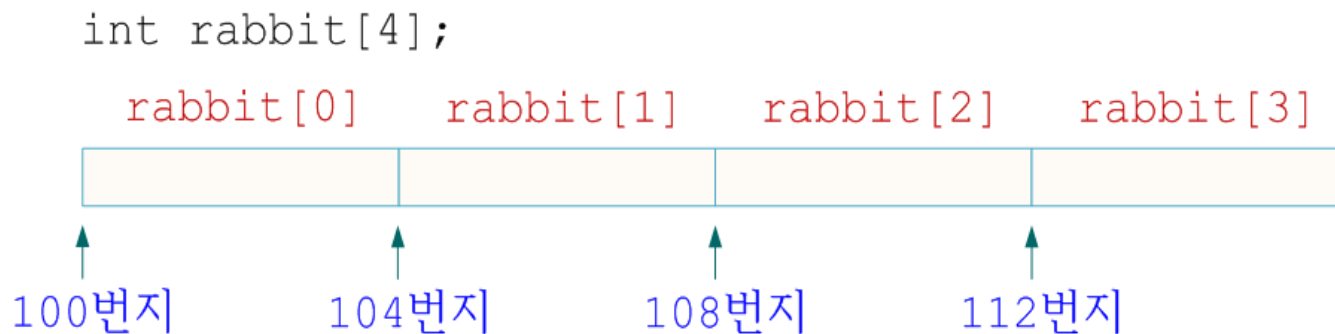
```
    for(i=0; i<size; i++)
```

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

```
    return 0;
```

```
}
```

배열 원소 개수 자동 계산



$$\&\text{rabbit}[i] = \&\text{rabbit}[0] + \text{sizeof}(\text{int}) * i$$

- 배열은 연속된 메모리 공간(Contiguous Memory Space)에 존재
 - 요소끼리 서로 붙어 있기 때문에 위 공식이 성립
 - cf. 만약 배열 인덱스를 1부터 시작하면 이는
 - $\&\text{rabbit}[i] = \&\text{rabbit}[1] + \text{sizeof}(\text{int}) * (i - 1)$ 로 바뀜
 - $(i - 1)$ 계산으로 인한 시간적 부담 초래

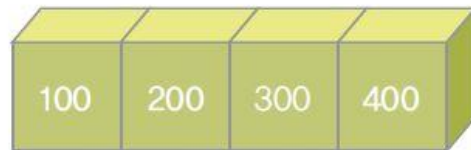
배열 명 \equiv 배열 시작 주소

$\text{rabbit} \equiv \&\text{rabbit}[0]$

$\&\text{rabbit}[i] = \text{rabbit} + \text{sizeof}(\text{int}) * i$

■ 정수형 배열과 문자형 배열

정수형 배열



aa[0] aa[1] aa[2] aa[3]

정수형(int): 각 4바이트

```
int aa[4] = {100, 200, 300, 400};
```

문자형 배열



ss[0] ss[1] ss[2] ss[3]

문자형(char): 각 1바이트

```
char ss[4] = {'X', 'Y', 'Z', '\0'};
```

또는

```
char ss[4];
```

```
ss[0] = 'X';
```

```
ss[1] = 'Y';
```

```
ss[2] = 'Z';
```

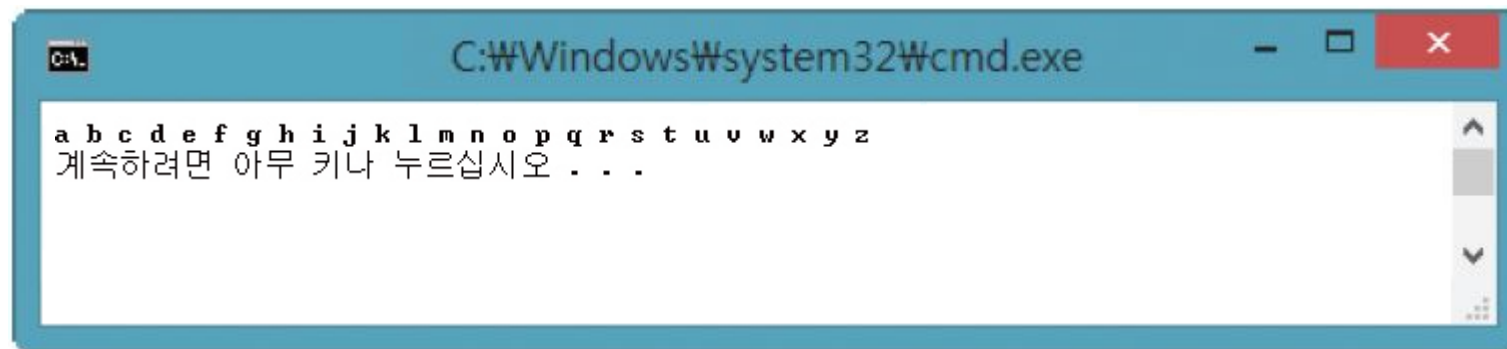
```
ss[3] = '\0';
```

또는

```
char ss[4] = "XYZ"
```

예제 설명 배열을 사용(`codes[26]`)하여 a부터 z까지 출력하여 보자

실행 결과

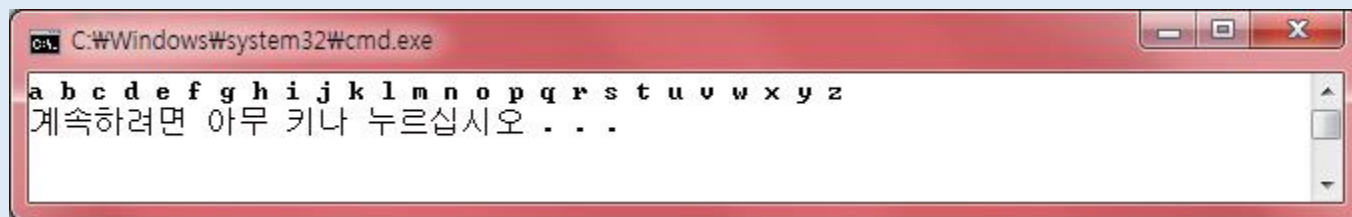


```
C:\Windows\system32\cmd.exe
a b c d e f g h i j k l m n o p q r s t u v w x y z
계속하려면 아무 키나 누르십시오 . . .
```

[예제] 문자형 배열

```
01 #include <stdio.h>
02 #define SIZE 26
03
04 int main(void)
05 {
06     int i;
07     char codes[SIZE];
08
09     for (i = 0; i < SIZE; i++)
10         codes[i] = 'a' + i;
11
12     for (i = 0; i < SIZE; i++)
13         printf("%c ", codes[i]);
14     printf("\n");
15
16     return 0;
17 }
```

// 'a'에 1을 더하면 'b'가 된다.



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed on two lines: the first line contains the lowercase alphabet 'a' through 'z' separated by spaces, and the second line contains the Korean text '계속하려면 아무 키나 누르십시오...' followed by three dots.

```
C:\Windows\system32\cmd.exe
a b c d e f g h i j k l m n o p q r s t u v w x y z
계속하려면 아무 키나 누르십시오...
```

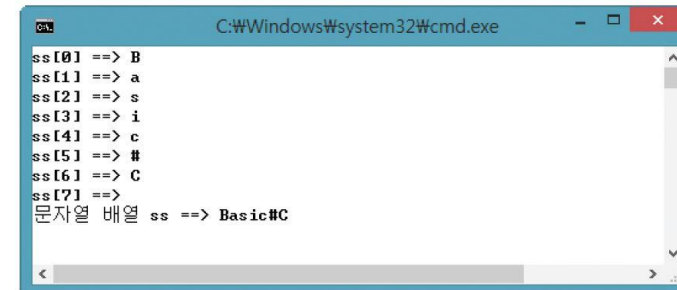
2. 배열과 문자열 - 문자열을 선언하고 출력하는 예

```

01 #include <stdio.h>
02
03 int main( )
04 {
05     char ss[8] = "Basic-C";      ---크기가 8인 문자형 배열을 선언하고 초기화한다.
06     int i;
07
08     ss[5] = '#';                ---여섯 번째 문자를 바꾼다.
09
10     for(i=0; i<8; i++)          ---여덟 번 반복하면서 배열 ss의 각 문자를 출력한다.
11     {
12         printf("ss[%d] == > %c \n", i, ss[i]);
13     }
14
15     printf("문자열 배열 ss == > %s \n", ss);  ---배열 ss의 전체 문자열을 출력한다.
16 }

```

실행 결과 ▼



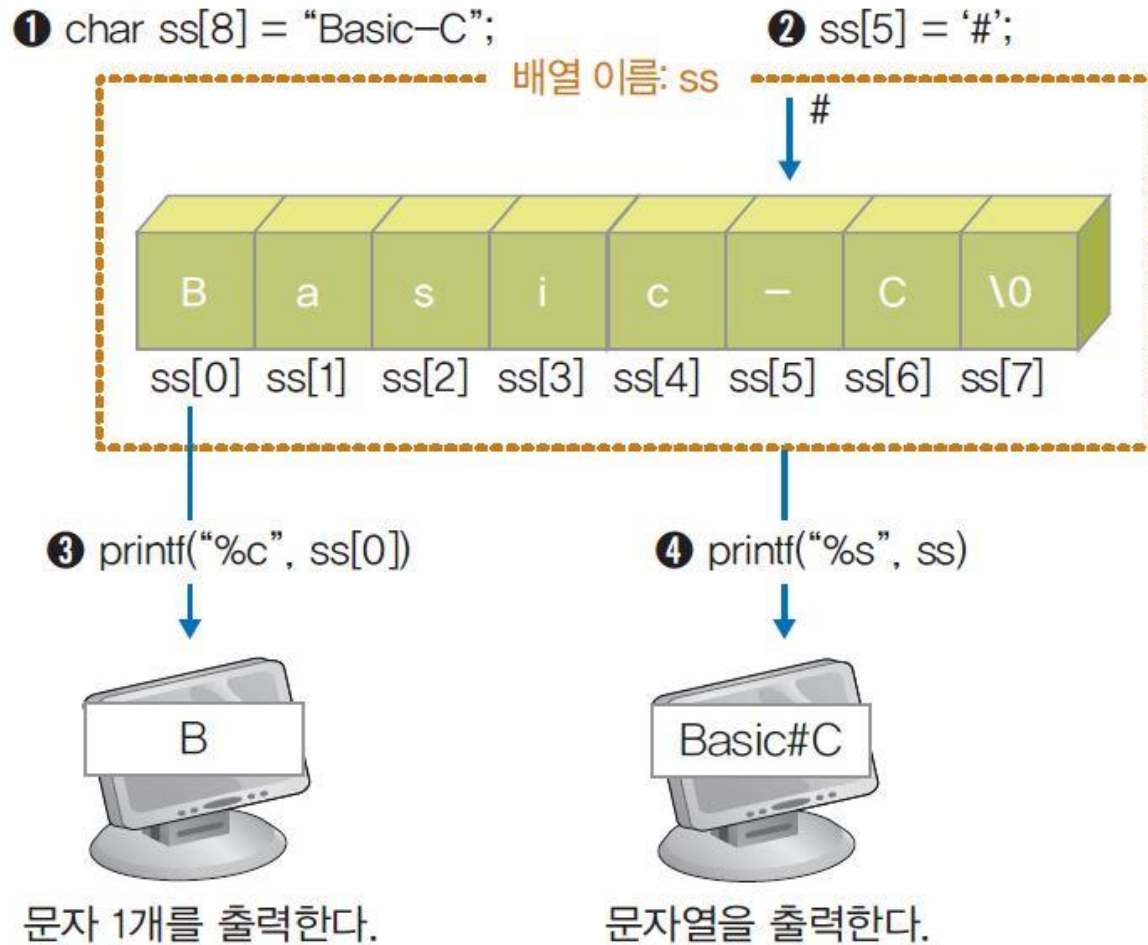
```

C:\Windows\system32\cmd.exe
ss[0] ==> B
ss[1] ==> a
ss[2] ==> s
ss[3] ==> i
ss[4] ==> c
ss[5] ==> #
ss[6] ==> C
ss[7] ==>
문자열 배열 ss ==> Basic#C

```

2. 배열과 문자열

- 문자 출력과 문자열 출력 비교



2. 배열과 문자열 - 문자열을 반대 순서로 출력하는 예

```

01 #include <stdio.h>
02
03 int main( )
04 {
05     char ss[5] = "abcd";
06     char tt[5];
07     int i;
08
09     for(i=0; i<4; i++)
10     {
11         tt[i] = ss[3-i];
12     }
13     tt[4] = '\0';
14
15     printf("거꾸로 출력한 결과== > %s \n", tt);
16 }

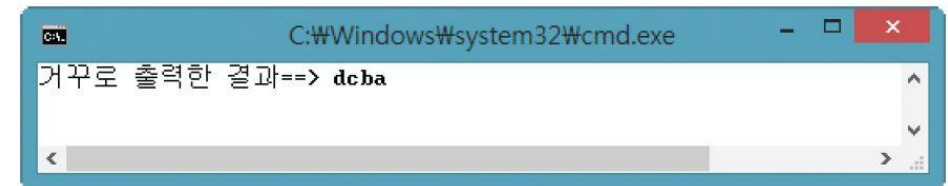
```

----크기 5의 문자형 배열 ss와
변환해서 저장할 배열 tt이다.

----4회 반복해서 각 배열에 문자를
반대 순서로 대입한다.

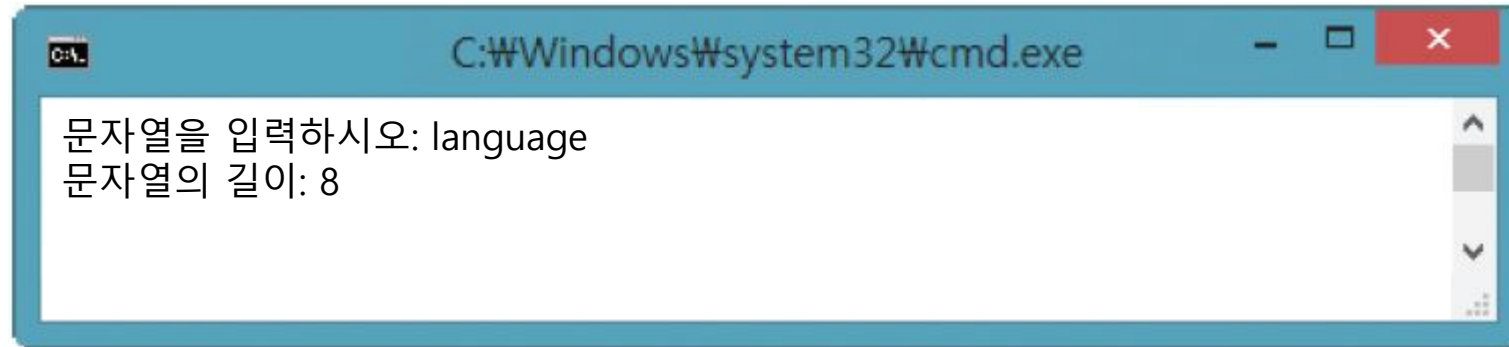
----마지막에 널 문자를 삽입한다.

실행 결과 ▼



예제 설명 문자열을 입력받고 입력받은 문자열의 길이를 출력해 보자

실행 결과



[예제] 문자열 길이

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int len = 0;
06     char s[10];
07
08     printf("문자열을 입력하시오: ");
09     scanf_s("%s", s, sizeof(s));
10
11     while(s[len] != '\0')
12         len++;
13
14     printf("문자열 \"%s\"의 길이 : %d\n", s, len);
15     return 0;
16 }
17
```

2. 배열과 문자열

■ 문자열 함수로 문자열 다루기

- 문자열 처리 함수

```
#include <string.h>
```

- 문자열의 길이를 알려주는 함수 : `strlen()`



↓ `strlen(ss)`: 널 문자를 제외한 문자의 개수만 센다.

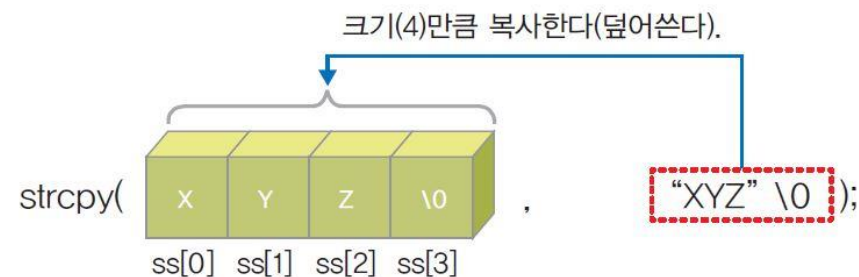
3

[예제] 문자열 길이

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int len = 0;
06     char s[10];
07
08     printf("문자열을 입력하시오: ");
09     scanf_s("%s", s, sizeof(s));
10
11     len = strlen(s);
12
13     printf("문자열 \"%s\"의 길이 : %d\n", s, len);
14     return 0;
15 }
```

2. 배열과 문자열

- 문자열을 복사하는 함수 : strcpy()
 - strcpy(문자열 배열 A, 문자열 B)는 '문자열 배열 A'에 '문자열 B'를 복사.



```
01 #include <stdio.h>
02 #include <string.h>
03
04 int main( )
05 {
06     char ss[4];          ---문자열 배열을 선언한다.
07
08     strcpy(ss, "XYZ");    ---배열 ss에 문자열 "XYZ"를 복사한다.
09
10     printf("문자열 ss의 내용 == > %s \n", ss);
11 }
```

C:\Windows\system32\cmd.exe

문자열 ss의 내용 ==> XYZ

2. 배열과 문자열

- strcpy() 함수의 사용

- 오류

```
ss = "XYZ"
```

실행 결과 ▶

오류

- 올바른 사용

```
char ss[4] = "XYZ"
```

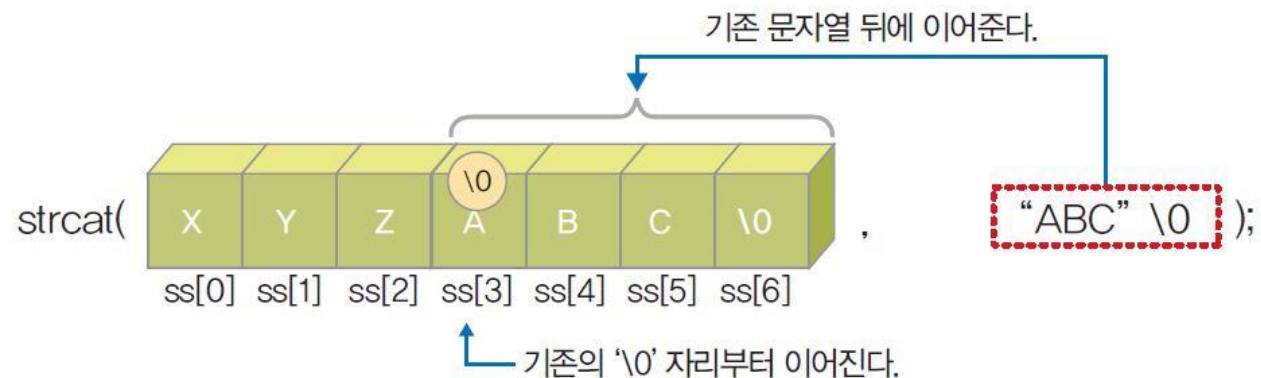
실행 결과 ▶

가능

```
char ss[4];  
ss[0] = 'X';  
ss[1] = 'Y';  
ss[2] = 'Z';  
ss[3] = '\0';
```

2. 배열과 문자열

- 두 문자열을 이어주는 함수 : `strcat()`
 - `strcat(문자열 배열 A, 문자열 B)`는 '문자열 배열 A'와 '문자열 B'를 이어 다시 '문자열 배열 A'에 넣음.
(문자열 배열 A의 길이는 문자열 배열 A와 문자열 B를 합친 길이 +1 이상이어야 함)



`strcat(A, B)`의 경우에 A는 꼭 문자형 배열이어야 한다. 이는 A와 B를 이은 결과를 다시 A(배열)에 넣어야 하기 때문이다. B는 문자형 배열이든 문자열 상수든 상관없다. `strcpy(A, B)`도 마찬가지이다.

```
char ss[10] = "XYZ";  
char tt[4] = "ABC";
```

`strcat(ss, tt)` \Rightarrow (○)

`strcat(ss, "ABC")` \Rightarrow (○)

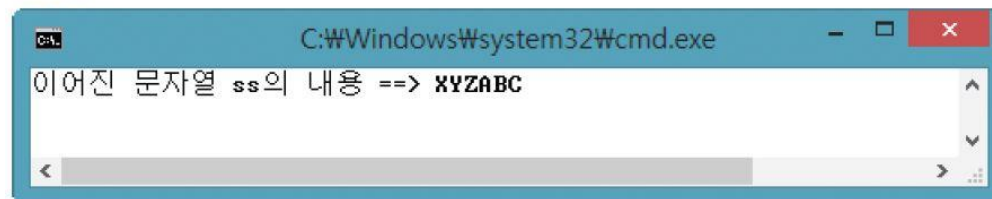
`strcat("ABC", "XYZ")` \Rightarrow (×)

`strcat("ABC", ss)` \Rightarrow (×)

2. 배열과 문자열 - 문자열 처리 함수 strcat() 사용 예

```
01 #include <stdio.h>
02 #include <string.h>
03
04 int main( )
05 {
06     char ss[7] = "XYZ";      ----문자열 배열을 선언하고 초기화한다.
07
08     strcat(ss, "ABC");      ----배열 ss의 내용("XYZ")에 문자열 "ABC"를 이어서
                                다시 ss에 대입한다.
09
10     printf("이어진 문자열 ss의 내용 == > %s \n", ss);
11 }
```

실행 결과 ▼



2. 배열과 문자열

- 두 문자열을 비교하는 함수 : strcmp()
 - strcmp(문자열 A, 문자열 B)는 'A-B'의 결과를 돌려줌.
 - 결과가 0이 나오면 A와 B가 같은 문자열이라는 뜻, 그 외의 값은 두 문자열이 다르다는 뜻임.

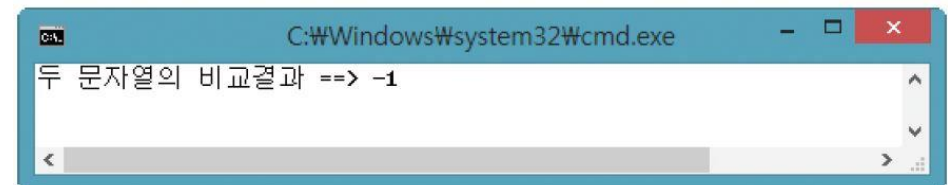
2. 배열과 문자열 - 문자열 처리 함수 strcmp() 사용 예

```
01 #include <stdio.h>
02 #include <string.h>
03
04 int main( )
05 {
06     char ss[] = "XYZ";
07     char tt[] = "xyz";
08     int r;
09
10     r = strcmp(ss, tt);
11
12     printf("두 문자열의 비교결과 == > %d \n", r);
13 }
```

---문자열 배열 ss와 tt를 선언하고 초깃값을 대입한다.

---두 문자열을 비교한 결과를 r에 대입한다.

실행 결과 ▼

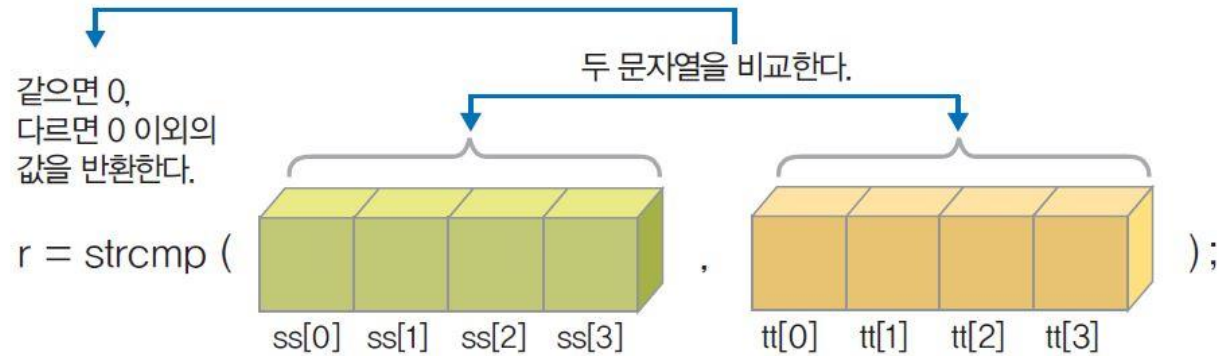


C:\Windows\system32\cmd.exe

```
두 문자열의 비교결과 ==> -1
```

2. 배열과 문자열

- `strcmp()` 함수는 두 문자열을 비교해서 같으면 0, 다르면 그 외의 숫자를 돌려줌. 주로 두 문자열이 같은지 확인할 때 사용함.
- 아래의 `ss`와 `tt`가 서로 다르므로(대문자와 소문자는 완전히 다른 문자) 0 이외의 값이 나옴.



`strcmp` 함수의
의미

`strcmp(ss, tt)` 함수는 `ss`의 아스키코드 값에서 `tt`의 아스키코드 값을 뺀다. 그래서 두 문자열이 같으면 결과값이 0이고, 다르면 0이 아닌 값이 나온다. 0 이외의 값은 두 문자열의 아스키코드 값 차이를 나타내는데, 그다지 활용할 일은 없고 단지 두 문자열이 다르다는 뜻으로만 받아들이면 된다(일부 컴파일러의 경우에는 `ss > tt`일 때 1, `ss == tt`일 때 0, `ss < tt`일 때 -1 값을 돌려주기도 한다).

2. 배열과 문자열

- 문자열 입출력 함수
- 문자열을 입력받는 함수 : `gets()`
 - `scanf_s()`와 기능은 비슷하지만 문자열 입력 시 더 유용함.
 - 형식은 `gets(입력받을 문자 배열)`, 최대 입력 문자는 널 문자 고려해서 '배열 크기 -1'까지 입력 가능
 - [Enter]를 입력할 때까지 `ss`에 문자열을 받아들임.

```
char ss[10];  
gets(ss);
```

- 문자열을 출력하는 함수 : `puts()`
 - `printf()`와 기능은 비슷하지만 문자열 출력 시 더 유용함.
 - '\n'이 없어도 출력한 후 자동으로 줄을 넘김.

```
char ss[10] = "XYZ";  
puts(ss);
```

2. 배열과 문자열 - 문자열 입출력 함수 gets(), puts() 사용 예

```

01 #include <stdio.h>
02 #include <string.h>
03
04 int main( )
05 {
06     char ss[20];          ----문자형 배열 ss와 tt를 선언한다.
07     char tt[20];
08     int r1, r2;
09
10     puts("첫번째 문자열을 입력하세요.");
11     gets(ss);             ----배열 ss와 tt에 문자열을 입력한다.
12
13     puts("두번째 문자열을 입력하세요.");
14     gets(tt);
15
16     r1 = strlen(ss);      ----배열 ss와 tt의 문자열 길이를 저장한다.
17     r2 = strlen(tt);
18
19     printf("첫번째 문자열의 길이 == > %d \n", r1);
20     printf("두번째 문자열의 길이 == > %d \n", r2);
21
22     if ( strcmp(ss,tt) == 0)
23         puts("두 문자열의 내용이 같습니다.\n");
24     else
25         puts("두 문자열의 내용이 다릅니다.\n");
26 }

```

```

첫번째 문자열을 입력하세요.
IT Cookbook
두번째 문자열을 입력하세요.
Hanbit
첫번째 문자열의 길이 ==> 11
두번째 문자열의 길이 ==> 6
두 문자열의 내용이 다릅니다.

```

배열의 복사



```
int score1[SIZE];  
int score2[SIZE];  
score1 = score2;
```

잘못된 방법

// 컴파일 오류!



```
int score1[SIZE];  
int score2[SIZE];  
int i;
```

원소를 일일이
복사한다

```
for(i = 0; i < SIZE; i++)  
    score1[i] = score2[i];
```

올바른 방법

배열의 비교



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

```
#define SIZE 5
```

```
int main( )
```

```
{
```

```
    int i;
```

```
    int a[SIZE] = { 1, 2, 3, 4, 5 };
```

```
    int b[SIZE] = { 1, 2, 3, 4, 5 };
```

```
    if( a == b )
```

```
        printf("잘못된 결과입니다.\n");
```

```
    else
```

```
        printf("잘못된 결과입니다.\n");
```

```
    return 0;
```

```
}
```

// ① 올바르게 않은 배열 비교

배열의 비교

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

int main( )
{
    for(i = 0; i < SIZE ; i++) // ② 올바른 배열 비교
    {
        if ( a[i] != b[i] )
        {
            printf("a[]와 b[]는 같지 않습니다.\n");
            return 0;
        }
    }
    printf("a[]와 b[]는 같습니다.\n");
    return 0;
}
```



원소를 하나씩
비교한다

```
#include <stdio.h>
//배열을 복사하는 array_copy() 함수의 프로토타입
void array_copy(int arr_1[5], int arr_2[5]);

void main()
{
    //복사할 배열의 선언 및 초기화
    int a[5]={1,2,3,4,5};
    //복사될 배열 선언
    int b[5],i;
    //array_copy() 함수호출
    array_copy(a,b);
    //복사할 배열의 값 출력
    printf("a:");
    for(i=0;i<5;i++)
        printf("%d ",a[i]);
    //복사된 배열의 값 출력
    printf("\nb:");
    for(i=0;i<5;i++)
        printf("%d ",b[i]);
}

// array_copy() 함수
void array_copy(int arr_1[5], int arr_2[5])
{
    int i;
    //배열의 복사
    for(i=0;i<5;i++)
        arr_2[i]=arr_1[i];
}
```



```
#include <stdio.h>
//문자열을 복사하는 string_copy()함수의 프로토타입
void string_copy(char* arr_1, char* arr_2);

void main()
{
    int i;
    //복사할 배열 선언 후 초기화
    char country[10] = "Korea";

    //복사될 배열 선언
    char nation[10];

    //문자열을 복사하는 string_copy() 함수호출
    string_copy(country,nation);

    //원본 배열과 복사한 문자열을 출력
    printf("country[]:%s\n",country);
    printf("nation[]:%s",nation);
}

// 문자열을 복사하는 string_copy() 함수
void string_copy(int arr_1[5], int arr_2[5])
{
    int i=0;
    //배열의 값이 NULL 이 될때까지 배열의 값을 복사
    while(arr_1[i] !=NULL){
        arr_2[i] = arr_1[i];
        i++;
    }
    //복사된 배열의 마지막에 NULL 추가
    arr_2[i] = NULL;
}
```

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

void main()
{
    //문자배열을 선언한 후 초기화
    char str1[10] = "abc";
    char str2[10] = "12345";
    char str3[10] = "";
    int temp;

    //문자 배열에 저장되어 있는 문자 길이를 구한다.
    printf("str1:%s, str2:%s, str3:%s\n",str1,str2,str3);
    temp = strlen(str1);
    printf("strlen(str1):%d\n",temp);
    temp = strlen(str2);
    printf("strlen(str2):%d\n",temp);
    temp = strlen(str3);
    printf("strlen(str3):%d\n\n",temp);

    printf("str1:%s, str2:%s, str3:%s\n",str1,str2,str3);
    //str1을 str3에 복사한다.
    strcpy(str3, str1);
    printf("strcpy(str3,str1)\n");
    printf("str1 : %s, str3 : %s\n\n",str1,str3);

    printf("str1:%s, str2:%s, str3:%s\n",str1,str2,str3);
    //str2의 3자리만 str3에 복사한다.
    strcpy(str3, str2, 3);
    printf("strcpy(str3,str2,3)\n");
    printf("str2 : %s, str3 : %s\n\n",str2,str3);
```

```
printf("str1:%s, str2:%s, str3:%s\n",str1,str2,str3);
//str1을 str3에 이어 붙인다.
strcat(str3, str1);
printf("strcat(str3,str1)\n");
printf("str1 : %s, str3 : %s\n\n",str1,str3);

printf("str1:%s, str2:%s, str3:%s\n",str1,str2,str3);
//str2의 1자리만 str3에 복사한다.
strncat(str3, str2, 1);
printf("strncat(str3,str2,1)\n");
printf("str2 : %s, str3 : %s\n\n",str2,str3);

printf("str1:%s, str2:%s, str3:%s\n",str1,str2,str3);
//str1과 str2를 비교한다.
temp = strcmp(str1, str2);
printf("strcmp(str1, str2):%d\n",temp);

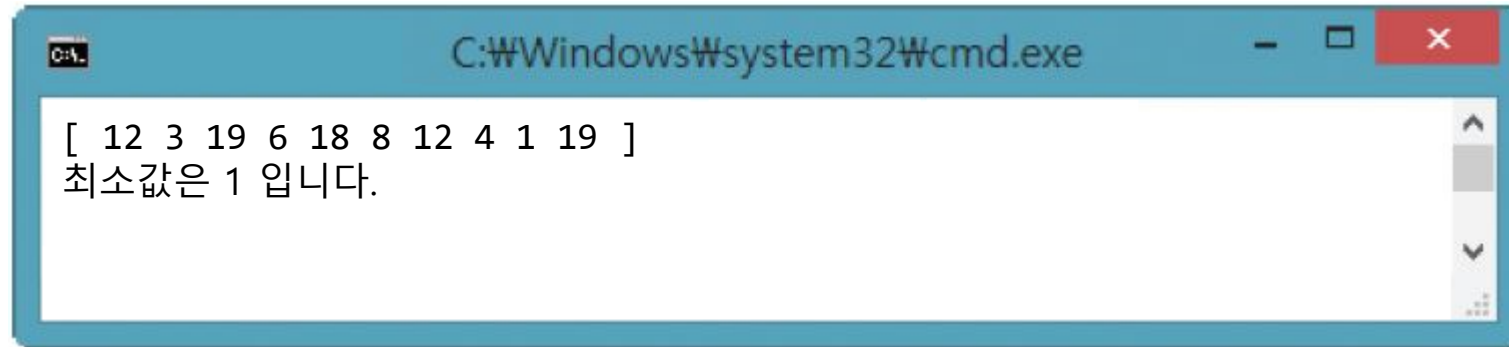
temp = strncmp(str2, str3, 3);
//str2와 str3의 첫번째 문자부터 3자리만 비교한다.
printf("strncmp(str2, str3, 3):%d\n",temp);
```

}

예제 설명

다음 배열(prices)값 가운데 가장 작은값(최소값)을 찾아보자
`prices[10] = { 12, 3, 19, 6, 18, 8, 12, 4, 1, 19 };`

실행 결과



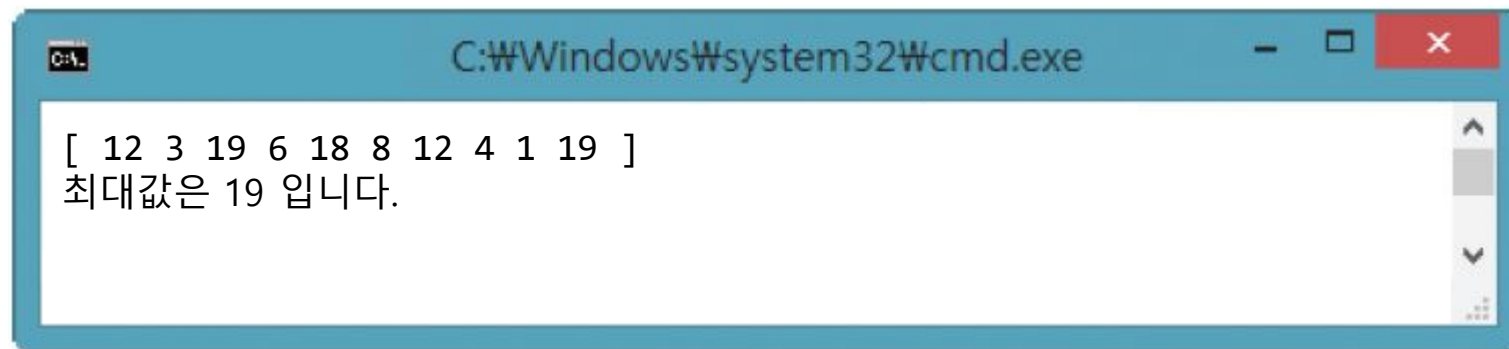
```
C:\Windows\system32\cmd.exe  
[ 12 3 19 6 18 8 12 4 1 19 ]  
최소값은 1 입니다.
```

[예제] 최소값 구하기

```
01 #include <stdio.h>
02 #define SIZE 10
03
04 int main(void)
05 {
06     int prices[SIZE] = { 12, 3, 19, 6, 18, 8, 12, 4, 1, 19 };
07     int i, minimum;
08     printf("[ ");
09     for (i = 0; i < SIZE; i++) {
10         printf("%d ", prices[i]);
11     }
12     printf("]\n");
13
14     minimum = prices[0];
15     for (i = 1; i < SIZE; i++)
16     {
17         if (prices[i] < minimum)
18             minimum = prices[i];
19     }
20     printf("최소값은 %d입니다.\n", minimum);
21     return 0;
22 }
```

예제 설명 앞의 최소값 예제를 활용하여
 다음 배열(prices)값 가운데 가장 큰값(최대값)을 찾아보자
prices[10] = { 12, 3, 19, 6, 18, 8, 12, 4, 1, 19 };

실행 결과



```
C:\Windows\system32\cmd.exe

[ 12 3 19 6 18 8 12 4 1 19 ]
최대값은 19 입니다.
```

[예제] 최대값 구하기

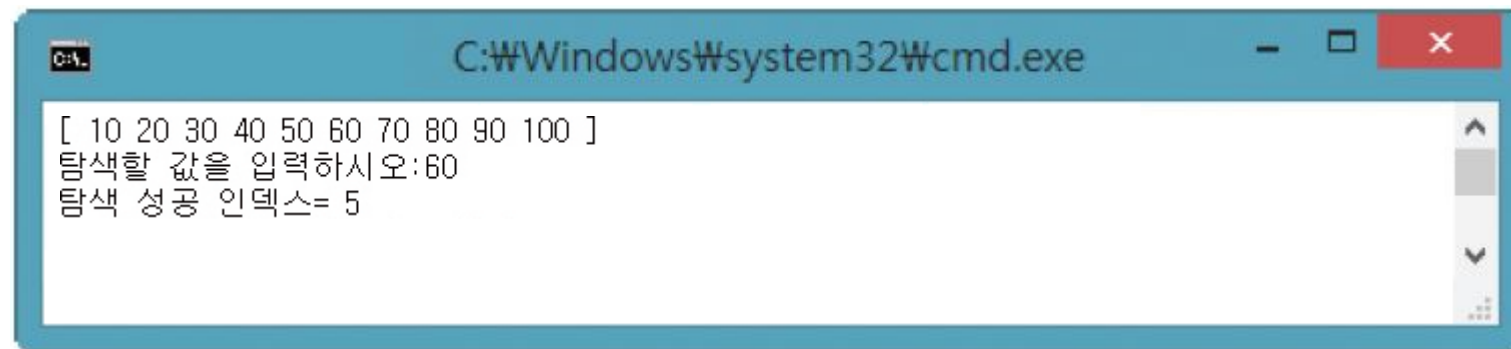
```
01 #include <stdio.h>
02 #define SIZE 10
03
04 int main(void)
05 {
06     int prices[SIZE] = { 12, 3, 19, 6, 18, 8, 12, 4, 1, 19 };
07     int i, maximum;
08     printf("[ ");
09     for (i = 0; i < SIZE; i++) {
10         printf("%d ", prices[i]);
11     }
12     printf("]\n");
13
14     maximum = prices[0];
15     for (i = 1; i < SIZE; i++)
16     {
17         if (prices[i] > maximum)
18             maximum = prices[i];
19     }
20     printf("최대값은 %d입니다.\n", maximum);
21     return 0;
22 }
```

예제 설명

정수들이 배열에 저장되어 있고 여기에서 사용자가 특정한 정수를 찾는다고 가정한다.

```
list[10] = { 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 };
```

실행 결과



```
C:\Windows\system32\cmd.exe  
[ 10 20 30 40 50 60 70 80 90 100 ]  
탐색할 값을 입력하시오:60  
탐색 성공 인덱스= 5
```

[예제] 배열에서 특정한 값 탐색하기

```
01 #include <stdio.h>
02 #define SIZE 10
03
04 int main(void)
05 {
06     int key, i;
07     int list[SIZE] = { 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 };
08     printf("[ ");
09     for (i = 0; i < SIZE; i++) {
10         printf("%d ", list[i]);
11     }
12     printf("]\n");
13     printf("탐색할 값을 입력하시오:");
14     scanf_s("%d", &key);
15
16     for (i = 0; i < SIZE; i++) {
17         if (list[i] == key) {
18             printf("탐색 성공 인덱스= %d\n", i);
19             break;
20         }
21     }
22     return 0;
23 }
```

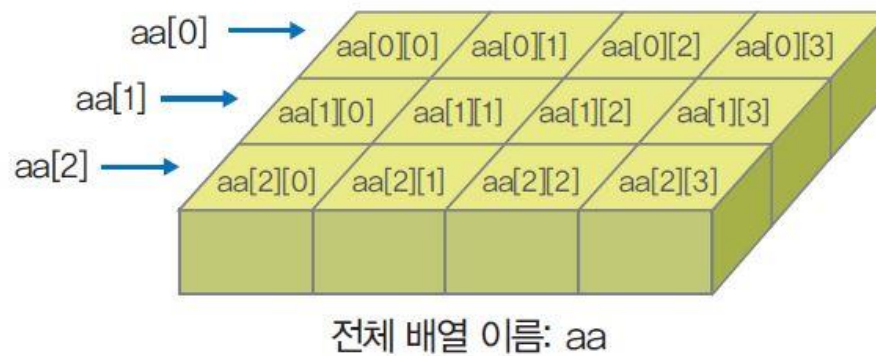

■ 2차원 배열의 기본 개념

- 2차원 배열은 1차원 배열을 여러 개 연결한 것
- 두 개의 첨자 사용

int aa[행]



int aa[행][열]



3. 2차원 배열 - 2차원 배열 사용 예 ①

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

```
02
```

```
03 int main( )
```

```
04 {
```

```
05     int aa[3][4];
```

----2차원 배열을 선언한다.

```
06
```

```
07     aa[0][0] = 1; aa[0][1] = 2; aa[0][2] = 3; aa[0][3] = 4;
```

```
08     aa[1][0] = 5; aa[1][1] = 6; aa[1][2] = 7; aa[1][3] = 8;
```

```
09     aa[2][0] = 9; aa[2][1] = 10; aa[2][2] = 11; aa[2][3] = 12;
```

```
10
```

```
11     printf("aa[0][0]부터 aa[2][3]까지 출력 \n");
```

```
12
```

----각 요소에 값을 대입한다.

```
13     printf("%3d %3d %3d %3d\n", aa[0][0], aa[0][1], aa[0][2], aa[0][3]);
```

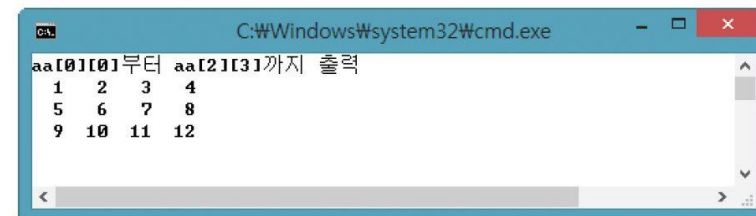
```
14     printf("%3d %3d %3d %3d\n", aa[1][0], aa[1][1], aa[1][2], aa[1][3]);
```

```
15     printf("%3d %3d %3d %3d\n", aa[2][0], aa[2][1], aa[2][2], aa[2][3]);
```

```
16 }
```

----배열의 내용을 출력한다.

실행 결과 ▼



```
C:\Windows\system32\cmd.exe
aa[0][0]부터 aa[2][3]까지 출력
 1  2  3  4
 5  6  7  8
 9 10 11 12
```

3. 2차원 배열 - 2차원 배열 사용 예 ②

67 |

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

```
02
```

```
03 int main( )
```

```
04 {
```

```
05     int aa[3][4];
```

```
06     int i, k;
```

```
07
```

```
08     int val=1;
```

```
09
```

```
10     for( i=0; i<3; i++ )
```

```
11     {
```

```
12         for( k=0; k<4; k++ )
```

```
13         {
```

```
14             aa[i][k] = val;
```

```
15             val++;
```

```
16         }
```

```
17     }
```

```
18
```

```
19     printf("aa[0][0]부터 aa[2][3]까지 출력 \n");
```

```
20
```

```
21     for( i=0; i<3; i++ )
```

```
22     {
```

```
23         for( k=0; k<4; k++ )
```

```
24         {
```

```
25             printf("%3d ", aa[i][k] );
```

```
26         }
```

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

```
28     }
```

```
29 }
```

---2차원 배열과 첨자 변수를 선언한다.

---배열에 들어갈 값을 초기화한다

-----바깥 for문을 세 번 반복한다.
즉, 앞 첨자가 행 단위로 변경된다.

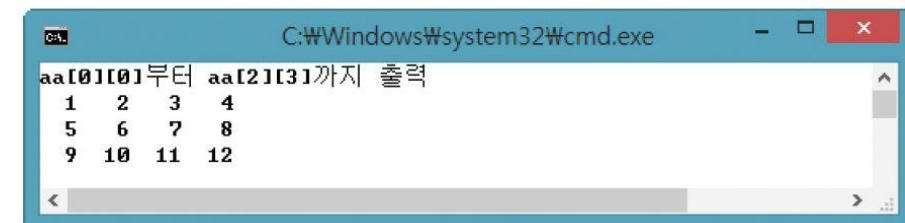
-----안쪽 for문을 네 번 반복한다.
즉, 뒤 첨자가 열 단위로 변경된다

-----배열에 val 값을 입력한 후 1 증가시킨다.

---입력과 동일한 개념으로 12회 출력한다.

-----한 행을 출력한 후 줄을 넘긴다.

실행 결과 ▼



```
C:\Windows\system32\cmd.exe
aa[0][0]부터 aa[2][3]까지 출력
 1  2  3  4
 5  6  7  8
 9 10 11 12
```

■ 2차원 배열의 초기화

- 2차원 배열을 선언하는 동시에 값 초기화하기

```
int aa[3][4] =  
{  
aa[0] → { 1 , 2 , 3 , 4 } ,  
aa[1] → { 5 , 6 , 7 , 8 } ,  
aa[2] → { 9 , 10 , 11 , 12 }  
};
```

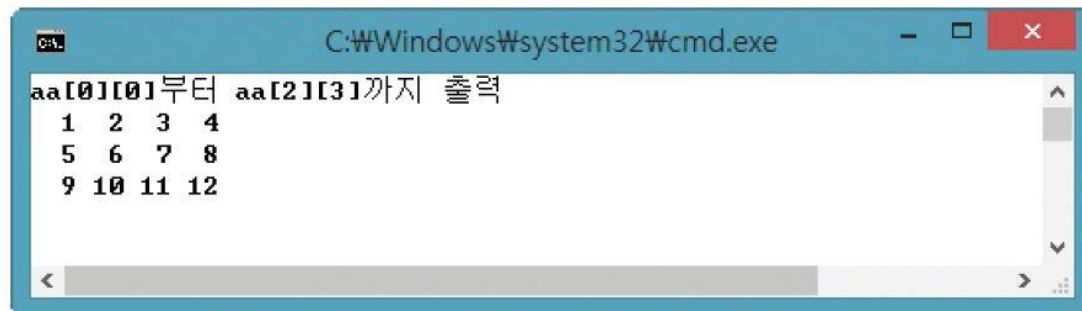
← 1행(aa[0])과 2행(aa[1])을 구분한다.
← 2행(aa[1])과 3행(aa[2])을 구분한다.
← 마지막 행이므로逗마가 없다.

3. 2차원 배열 - 2차원 배열의 초기화 예

```
01 #include <stdio.h>
02
03 int main( )
04 {
05     int aa[3][4] = {
06         { 1, 2, 3, 4 },
07         { 5, 6, 7, 8 },
08         { 9,10,11,12 }
09     };
10
11     int i, k;
12     printf("aa[0][0]부터 aa[2][3]까지 출력 \n");
13
14     for( i=0; i<3; i++ )
15     {
16         for( k=0; k<4; k++ )
17         {
18             printf("%3d", aa[i][k]);
19         }
20         printf("\n");
21     }
22     return 0;
23 }
```

----2차원 배열을 초기화한다.

----2차원 배열에 저장된 값을 출력한다.



```
C:\Windows\system32\cmd.exe
aa[0][0]부터 aa[2][3]까지 출력
 1  2  3  4
 5  6  7  8
 9 10 11 12
```

```
#include <stdio.h>

void main()
{
    //2행3열로 구성된 배열을 선언하고 초기화한다.
    int temp[2][3]={{10,20,30},{40,50,60}};

    //배열의 요소를 출력한다.
    printf("%4d %4d %4d\n",temp[0][0],temp[0][1],temp[0][2]);
    printf("%4d %4d %4d\n",temp[1][0],temp[1][1],temp[1][2]);

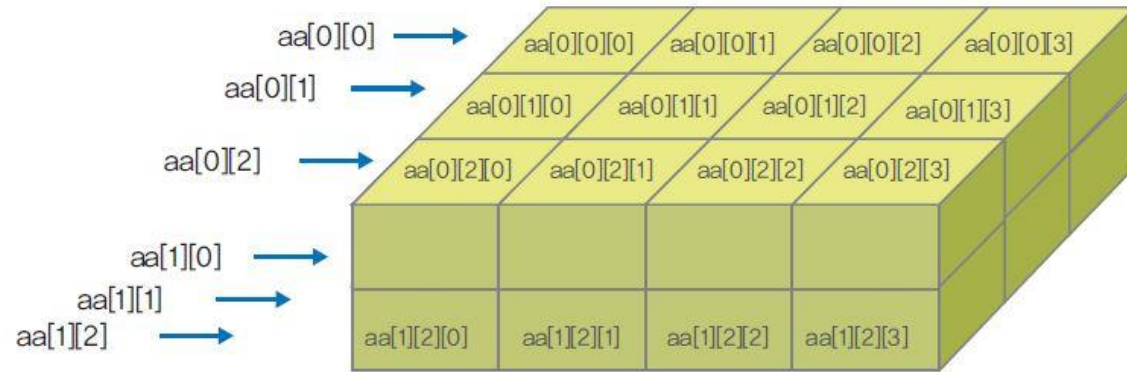
    //배열의 전체의 크기를 출력한다.
    printf("sizeof(배열전체):%d\n",sizeof(temp));

    //한 행의 크기를 출력한다.
    printf("sizeof(행):%d\n",sizeof(temp[0]));
}
```

■ 3차원 이상의 배열

- 2차원 배열 위에 또 다른 2차원 배열을 쌓은 것

int aa[면][행][열]



전체 배열 이름: aa

3. 2차원 배열

- 3차원 배열의 초기화
 - 2차원 배열의 초기화를 한번 더 하는 개념
 - 콤마로 분리하고, 전체를 다시 블록({ })으로 묶음

```
int aa[2][3][4] =  
{  
  {  
    { 1, 2, 3, 4 },  
    { 5, 6, 7, 8 },  
    { 9, 10, 11, 12 }  
  }  
  .  
  {  
    { 13, 14, 15, 16 },  
    { 17, 18, 19, 20 },  
    { 21, 22, 23, 24 }  
  }  
};
```

→ 윗면의 2차원 배열

→ 면 사이 분리

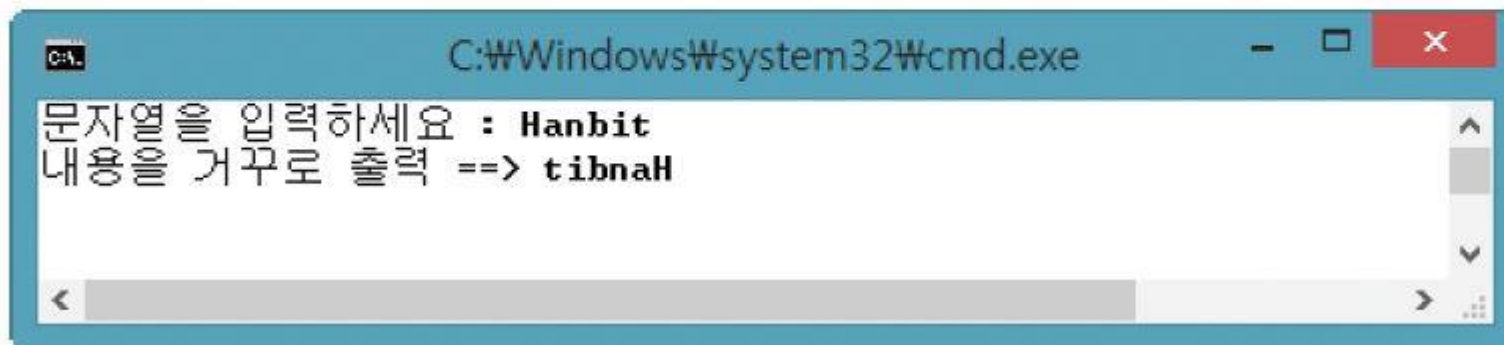
→ 아랫면의 2차원 배열

3차원 배열

```
01 #include <stdio.h>
02 void main()
03 { //3차원 배열을 선언하고 초기화
04     int temp[2][2][2]={{1,2},{3,4}},{5,6},{7,8}};
05
06     //3차원 요소를 출력한다.
07     printf("%4d %4d\n",temp[0][0][0],temp[0][0][1]);
08     printf("%4d %4d\n",temp[0][1][0],temp[0][1][1]);
09     printf("%4d %4d\n",temp[1][0][0],temp[1][0][1]);
10     printf("%4d %4d\n",temp[1][1][0],temp[1][1][1]);
11
12     //3차원 배열의 전체 크기, 면 크기, 행 크기를 출력한다.
13     printf("sizeof(배열전체):%d\n",sizeof(temp));
14     printf("sizeof(면):%d\n",sizeof(temp[0]));
15     printf("sizeof(행):%d\n",sizeof(temp[0][0]));
16 }
```

예제 설명 문자열 배열을 이용해서 입력받은 문자열을 반대 순서로 출력하는 프로그램이다.

실행 결과



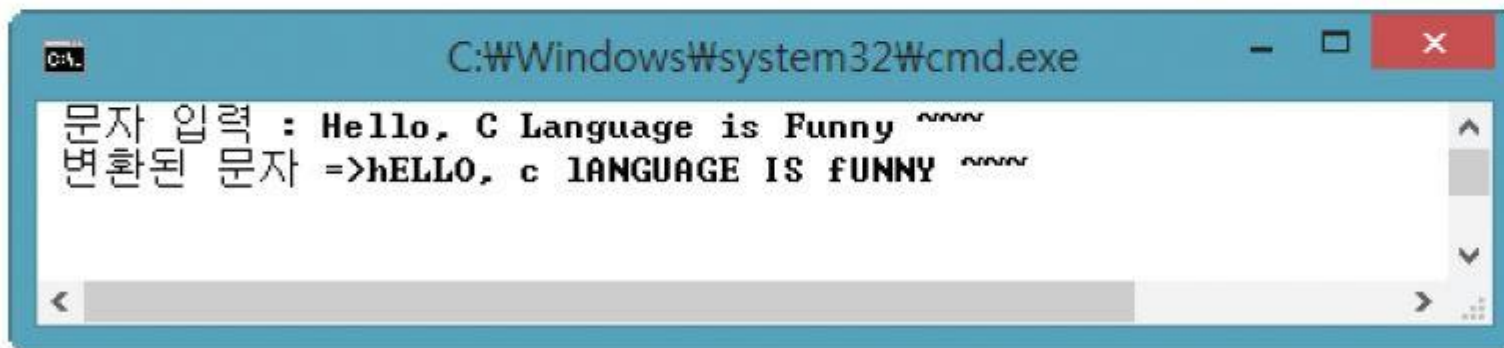
```
C:\Windows\system32\cmd.exe
문자열을 입력하세요 : Hanbit
내용을 거꾸로 출력 ==> tibnaH
```

[예제 1] 입력된 문자열을 반대 순서로 출력

```
01 #include <stdio.h>
02 #include <string.h>
03
04 int main( )
05 {
06     char ss[100];      ---문자형 배열 ss를 선언한다.
07     char tt[100];      ---문자형 배열 tt를 선언한다.
08     int count, i;
09
10     printf("문자열을 입력하세요 : ");
11     scanf_s("%s", ss, sizeof(ss));  ---문자열을 입력받는다.
12
13     count = strlen(ss);  ---입력받은 문자열의 개수를 구한다.
14
15     for(i=0; i<count; i++)  ---문자열의 개수만큼 반복해서 tt 배열에
16     {                      문자열을 반대 순서로 저장한다.
17         tt[i] = ss[count-(i+1)];
18     }
19     tt[count] = '\0';
20     ---tt 배열의 마지막에 널 문자를 입력한다
21     printf("내용을 거꾸로 출력 == > %s \n", tt);
22 }
```

예제 설명 입력된 문자열이 대문자이면 소문자로, 소문자이면 대문자로 변환하고 그 외의 문자는 그대로 출력하는 프로그램이다.

실행 결과



```
C:\Windows\system32\cmd.exe

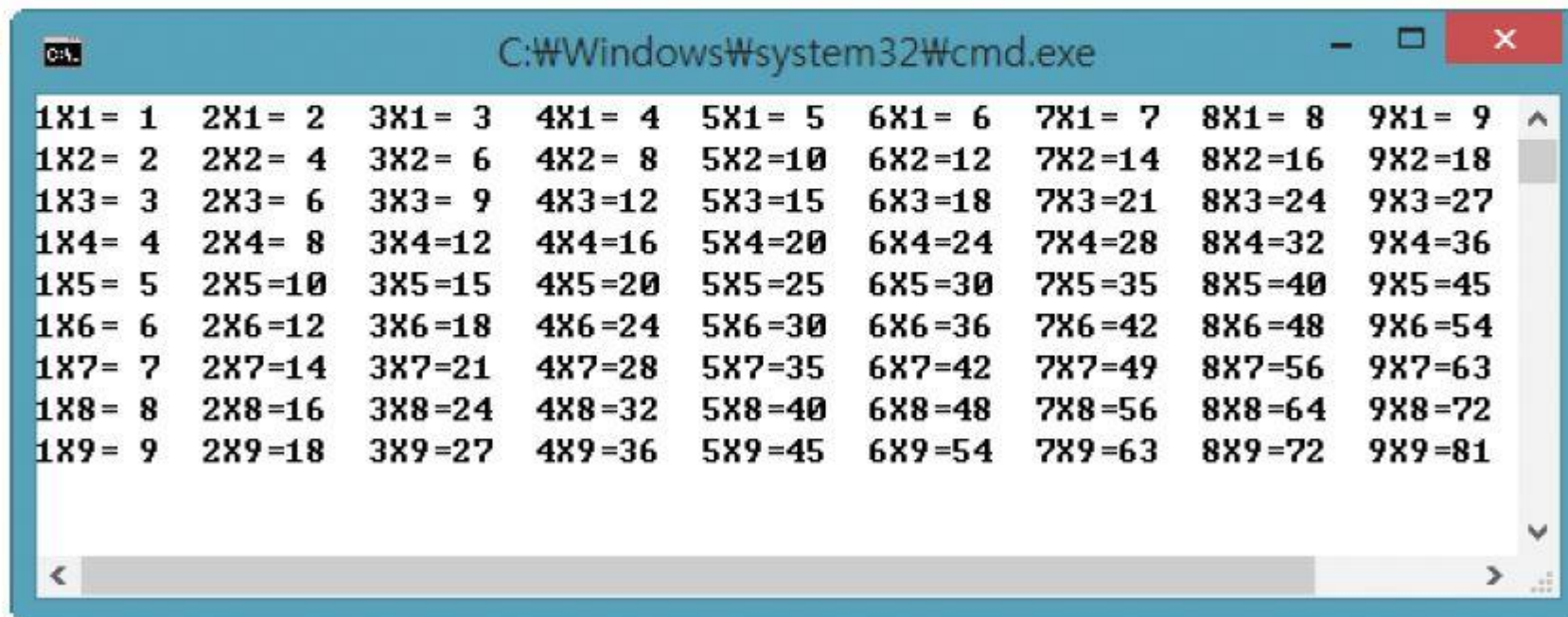
문자 입력 : Hello, C Language is Funny ~~~~
변환된 문자 =>hELLO, c LANGUAGE IS FUNNY ~~~~
```

[예제 2] 대문자와 소문자의 변환

```
01 #include <stdio.h>
02 #include <string.h>
03
04 int main( )
05 {
06     char in[50], out[50];      ---입력 문자형 배열 in과 출력 문자형 배열 out이다.
07     int i, len;
08     int diff = 'a' - 'A';      ---대문자와 소문자의 값 차이를 diff에 저장한다.
09
10     printf(" 문자 입력 : ");
11     gets(in);                  ---문자를 입력받는다. 실제 최대 입력 문자는
12                                ' 배열 크기-1'이다.
13     len = strlen(in);          ---입력한 문자열의 길이를 구한다.
14
15     for(i=0; i<len; i++)
16     {
17         if ( ('A' <= in[i]) && (in[i] <= 'Z') )    ---문자가 대문자이면 대소문자 차이값을 더한다.
18             out[i] = in[i] + diff;
19         else if ( ('a' <= in[i]) && (in[i] <= 'z') ) ---문자가 소문자이면 대소문자 차이값을 뺀다.
20             out[i] = in[i] - diff;
21         else                                          ---영문자가 아닌 기호, 숫자 등은 그대로 둔다.
22             out[i] = in[i];
23     }
24     out[i] = '\0';      ---마지막에 널 문자를 입력한다.
25
26     printf(" 변환된 문자 = >%s \n", out);
27 }
```

예제 설명 구구단의 결과를 2차원 배열에 저장한 후 출력하는 프로그램이다.

실행 결과



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window displays a 9x9 multiplication table. The rows are numbered 1 to 9, and the columns are numbered 1 to 9. Each entry in the table is in the format "row * column = result".

1x1= 1	2x1= 2	3x1= 3	4x1= 4	5x1= 5	6x1= 6	7x1= 7	8x1= 8	9x1= 9
1x2= 2	2x2= 4	3x2= 6	4x2= 8	5x2=10	6x2=12	7x2=14	8x2=16	9x2=18
1x3= 3	2x3= 6	3x3= 9	4x3=12	5x3=15	6x3=18	7x3=21	8x3=24	9x3=27
1x4= 4	2x4= 8	3x4=12	4x4=16	5x4=20	6x4=24	7x4=28	8x4=32	9x4=36
1x5= 5	2x5=10	3x5=15	4x5=20	5x5=25	6x5=30	7x5=35	8x5=40	9x5=45
1x6= 6	2x6=12	3x6=18	4x6=24	5x6=30	6x6=36	7x6=42	8x6=48	9x6=54
1x7= 7	2x7=14	3x7=21	4x7=28	5x7=35	6x7=42	7x7=49	8x7=56	9x7=63
1x8= 8	2x8=16	3x8=24	4x8=32	5x8=40	6x8=48	7x8=56	8x8=64	9x8=72
1x9= 9	2x9=18	3x9=27	4x9=36	5x9=45	6x9=54	7x9=63	8x9=72	9x9=81

[예제 3] 구구단의 결과를 2차원 배열에 저장

```
01 #include <stdio.h>
02
03 int main( )
04 {
05     int gugu[9][9];
06     int i, k;
07
08     for(i=0; i<9; i++)
09         for(k=0; k<9; k++)
10             gugu[i][k] = (i+1) * (k+1);
11
12     for(i=0; i<9; i++)
13     {
14         for(k=0; k<9; k++)
15         {
16             printf("%dX%d= %2d ", k+1, i+1, gugu[i][k]);
17         }
18         printf("\n");
19     }
20 }
```

----문자형 2차원 배열 gugu와 첨자 변수 i, k를 선언한다.

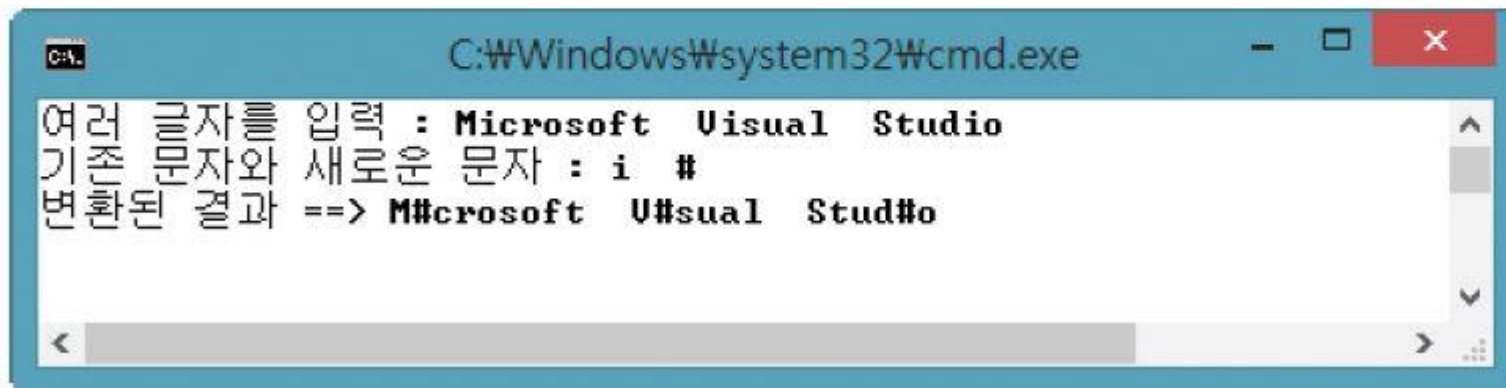
----구구단을 곱한 결과를 2차원 배열에 저장한다.
i, k가 0부터 시작되므로 1을 더해서 곱한다.

----구구단 결과를 출력한다.

-----한 행을 출력한 후 줄을 넘긴다.

예제 설명 문자열을 입력받고 그 문자열에서 변환할 기존 문자와 새로운 문자를 각각 입력받은 뒤 변환된 문자열을 반환하는 프로그램이다.

실행 결과



```
C:\Windows\system32\cmd.exe
여러 글자를 입력 : Microsoft Visual Studio
기존 문자와 새로운 문자 : i #
변환된 결과 ==> M#crosoft U#sual Stud#o
```



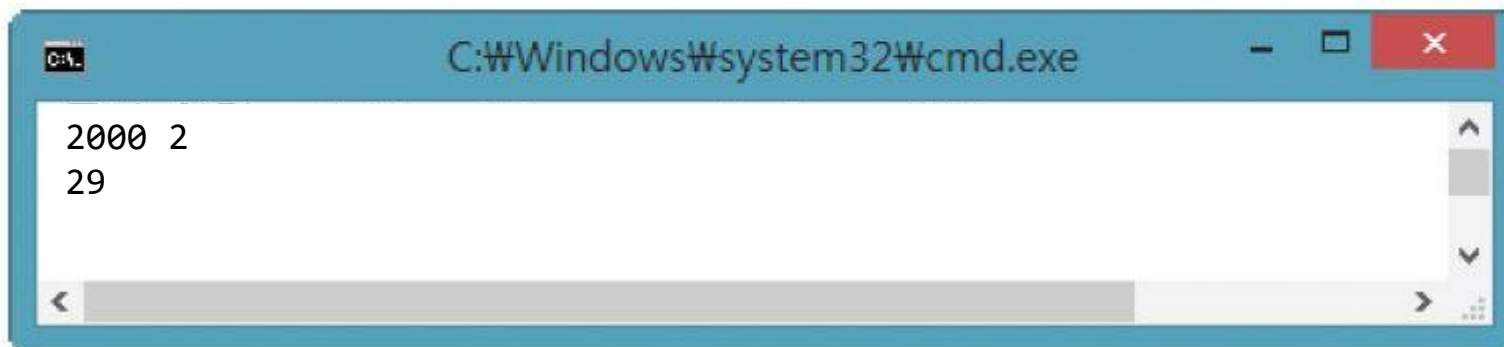
```
01 #include <stdio.h>
02 #include <string.h>
03
04 int main( )
05 {
06     char str[100];          ---문자형 배열 str을 선언한다.
07     char ch1, ch2;          ---기존 문자와 새 문자를 위한 문자형 변수이다.
08     int i;
09
10     printf("여러 글자를 입력 : ");
11     gets(str);              ---최대 99자를 입력받는다.
12
13     printf("기존 문자와 새로운 문자 : ");
14     scanf_s("%c %c", &ch1, &ch2, sizeof(ch1), sizeof(ch2));    ---기존 문자(ch1)와 새 문자(ch2)를 한 글자씩
                                                                    입력받는다(띄어쓰기로 구분).
15
16     for(i=0; i<strlen(str); i++)
17     {
18         if(str[i] == ch1)
19             str[i] = ch2;
20     }
21
22     printf("변환된 결과 == > %s \n", str);
23 }
```

예제 설명 연도와 월을 알고 있을때 그달의 마지막 날을 구하는 프로그램을 작성하시오

조건 1: 400의 배수인 해는 모두 윤년이다.

조건 2: 4의 배수인 해들 중 100의 배수가 아닌 해들은 모두 윤년이다.

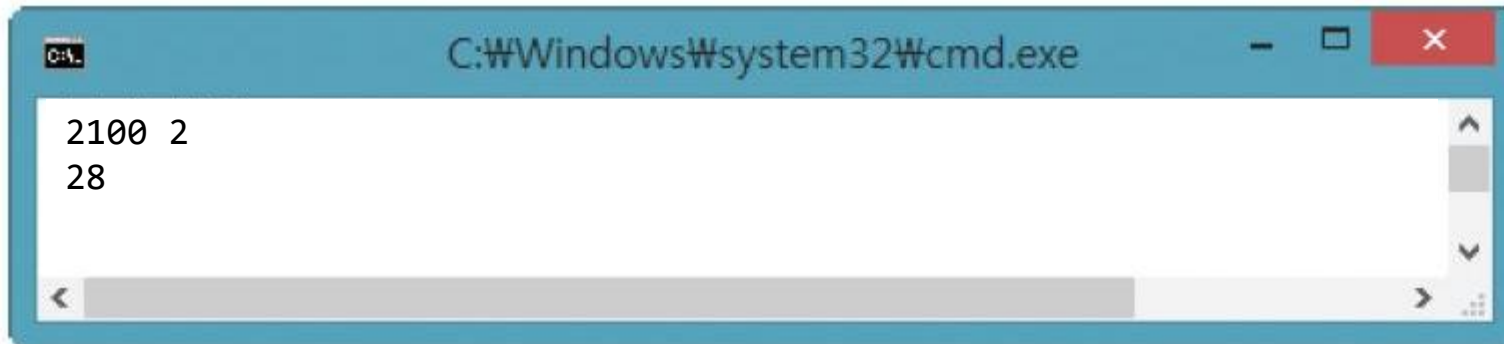
실행 결과



```
C:\Windows\system32\cmd.exe

2000 2
29
```

실행 결과



```
C:\Windows\system32\cmd.exe

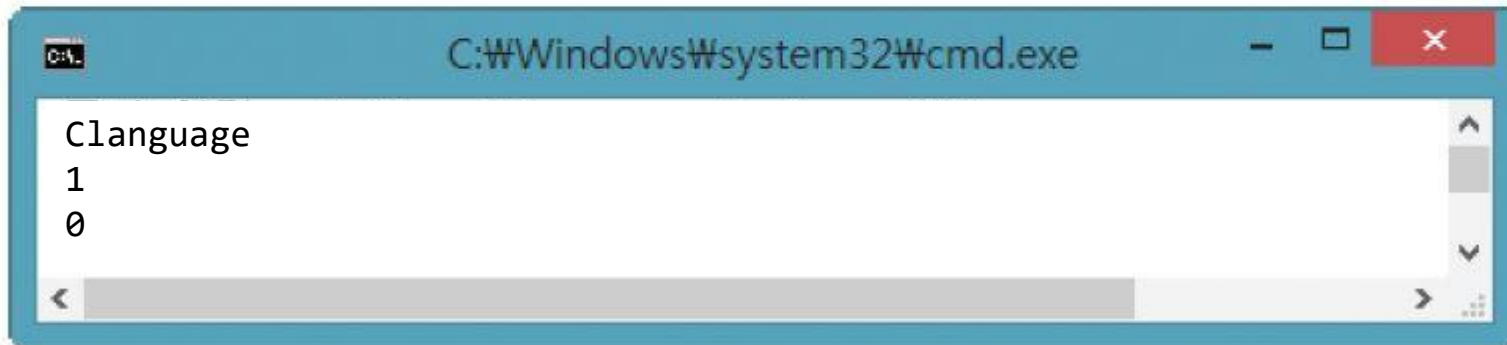
2100 2
28
```

[예제 5] 이달의 마지막 날을 구하라

```
01 #include <stdio.h>
02 void main( )
03 {
04     int year, month, l_month[12]={31,28,31,30,31,30,31,31,30,31,30,31};
05
06     scanf_s("%d %d", &year, &month);
07     if( (month == 2) && ((year%4)==0 && (year%100)!=0 || (year%400)==0))
08     {
09         printf("%d\n",29);
10     }
11     else
12     {
13         printf("%d\n",l_month[month-1]);
14     }
15 }
```

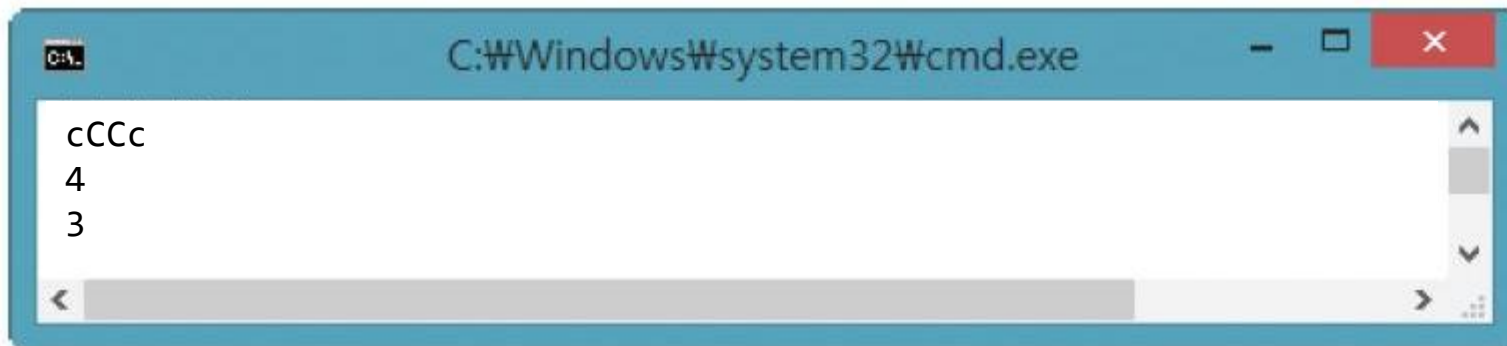
예제 설명 길이가 100 이하의 문자열을 입력받아, 'C'라는 문자와 'CC'라는 문자가 각각 몇 개 존재하는지 알아내는 프로그램을 작성하시오(단, C와 CC는 대소문자를 구분하지 않는다. 즉 'cC'는 'CC'와 같음)

실행 결과



```
C:\Windows\system32\cmd.exe
Clanguage
1
0
```

실행 결과



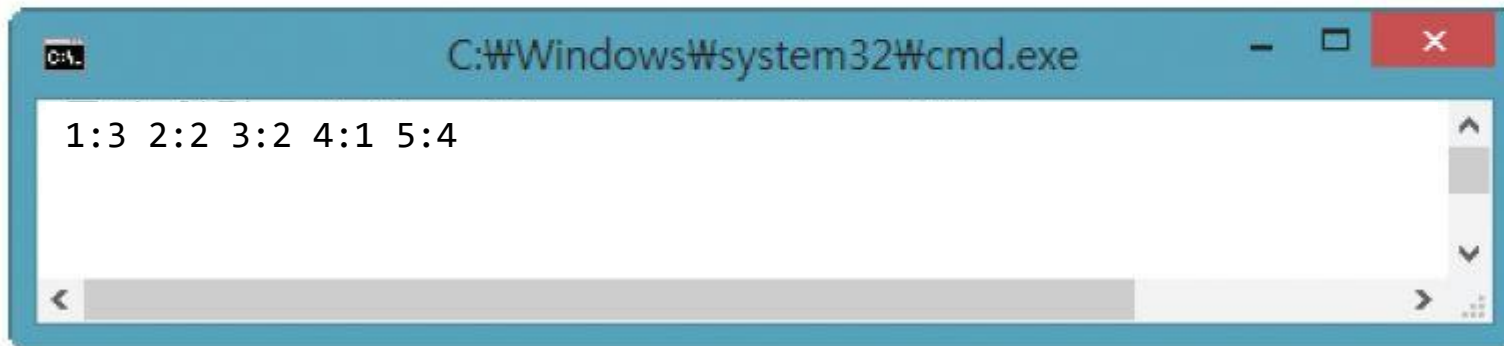
```
C:\Windows\system32\cmd.exe
cCCc
4
3
```

[예제 6] C언어를 찾아라

```
01 #include <stdio.h>
02 void main( )
03 {
04     char S[101];
05     int i, c=0, cc=0;
06     scanf_s("%s",S, sizeof(S));
07     for(i=0;S[i]!=NULL;i++){
08         if(S[i]=='c' || S[i]=='C'){
09             c++;
10             if(S[i+1]=='c' || S[i+1]=='C')
11                 cc++;
12         }
13     }
14     printf("%d\n",c);
15     printf("%d\n",cc);
16 }
```

예제 설명 다음 자료에서 나온 숫자(1~5)의 각 출현횟수를 출력하라
[자료] 1, 2, 5, 2, 4, 5, 1, 5, 5, 3, 3, 1

실행 결과



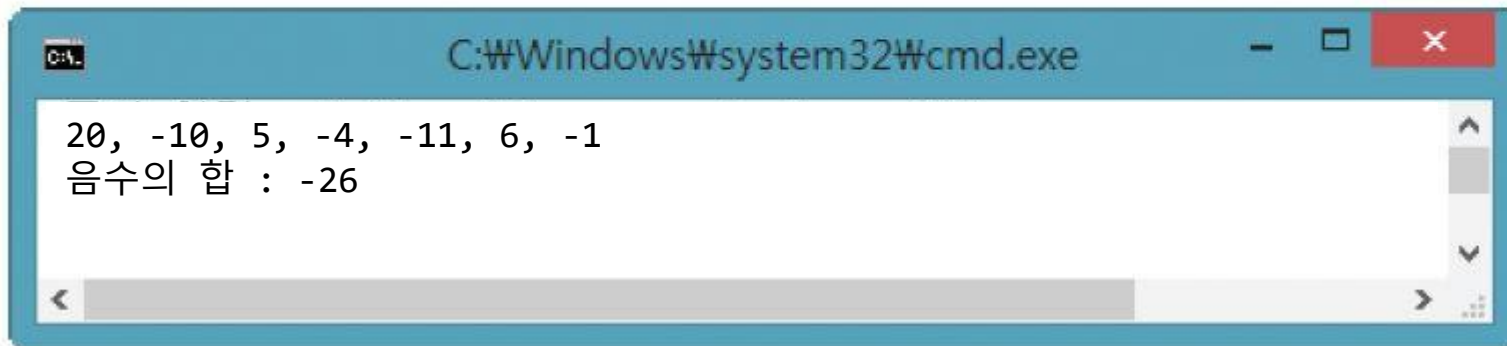
A screenshot of a Windows command prompt window. The title bar shows the path "C:\Windows\system32\cmd.exe". The command prompt icon is visible in the top-left corner. The output text displayed in the window is "1:3 2:2 3:2 4:1 5:4". The window has standard Windows window controls (minimize, maximize, close) in the top-right corner and a scroll bar on the right side.

```
C:\Windows\system32\cmd.exe  
1:3 2:2 3:2 4:1 5:4
```

```
01 #include <stdio.h>
02 void main( )
03 {
04     int i, j, count;
05     int data[] = {1, 2, 5, 2, 4, 5, 1, 5, 5, 3, 3, 1};
06     for(i=1;i<=5;i++){
07         for(j=0, count=0; j<sizeof(data)/sizeof(int); j++) {
08             if (i == data[j])
09                 count++;
10         }
11         printf("%d:%d ",i,count);
12     }
13     printf("\n");
14 }
```

예제 설명 정수배열 na에 20, -10, 5, -4, -11, 6, -1 값이 있을때 배열 요소 값을 모두 출력하고, 음수의 합을 출력하라

실행 결과



```
C:\Windows\system32\cmd.exe

20, -10, 5, -4, -11, 6, -1
음수의 합 : -26
```

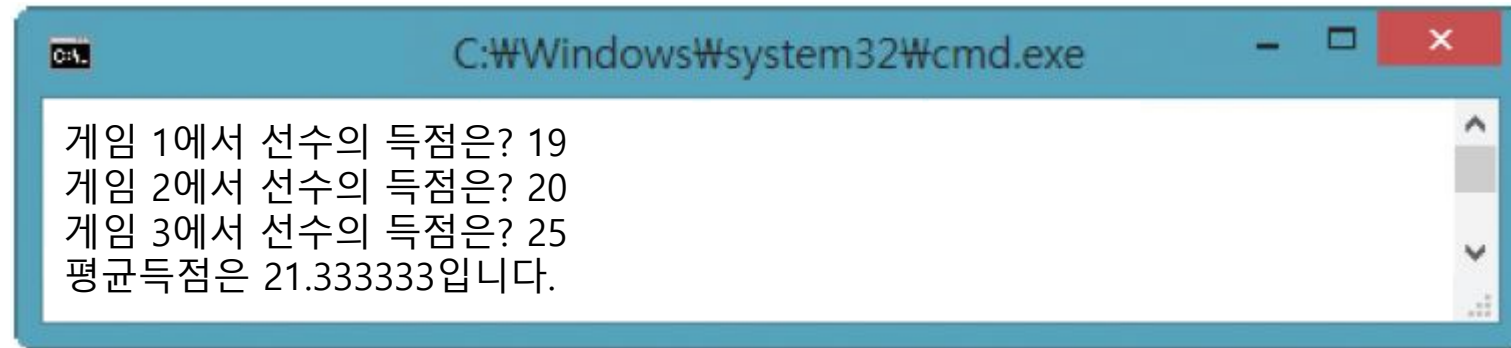
The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window contains two lines of text: "20, -10, 5, -4, -11, 6, -1" and "음수의 합 : -26". The window has a blue title bar and a red close button.


```
01 #include <stdio.h>
02
03 void main( ) {
04     int i, sum=0;
05     int na[] = {20, -10, 5, -4, -11, 6, -1};
06
07     for(i=0;i<sizeof(na)/sizeof(int);i++){
08         printf("%d ",na[i]);
09
10         if(na[i]<0) //음수면 합산 처리
11             sum += na[i];
12     }
13     printf("\n\n음수의 합 : %d\n\n",sum);
14 }
```

예제 설명

농구 게임에서 특정 선수의 평균 득점을 계산하는 프로그램을 작성해보자.
사용자는 최근 3게임에서의 득점을 입력한다. 평균은 실수로 계산하여 출력한다.

실행 결과



```
C:\Windows\system32\cmd.exe

게임 1에서 선수의 득점은? 19
게임 2에서 선수의 득점은? 20
게임 3에서 선수의 득점은? 25
평균득점은 21.333333입니다.
```

[예제 9] 평균득점 계산

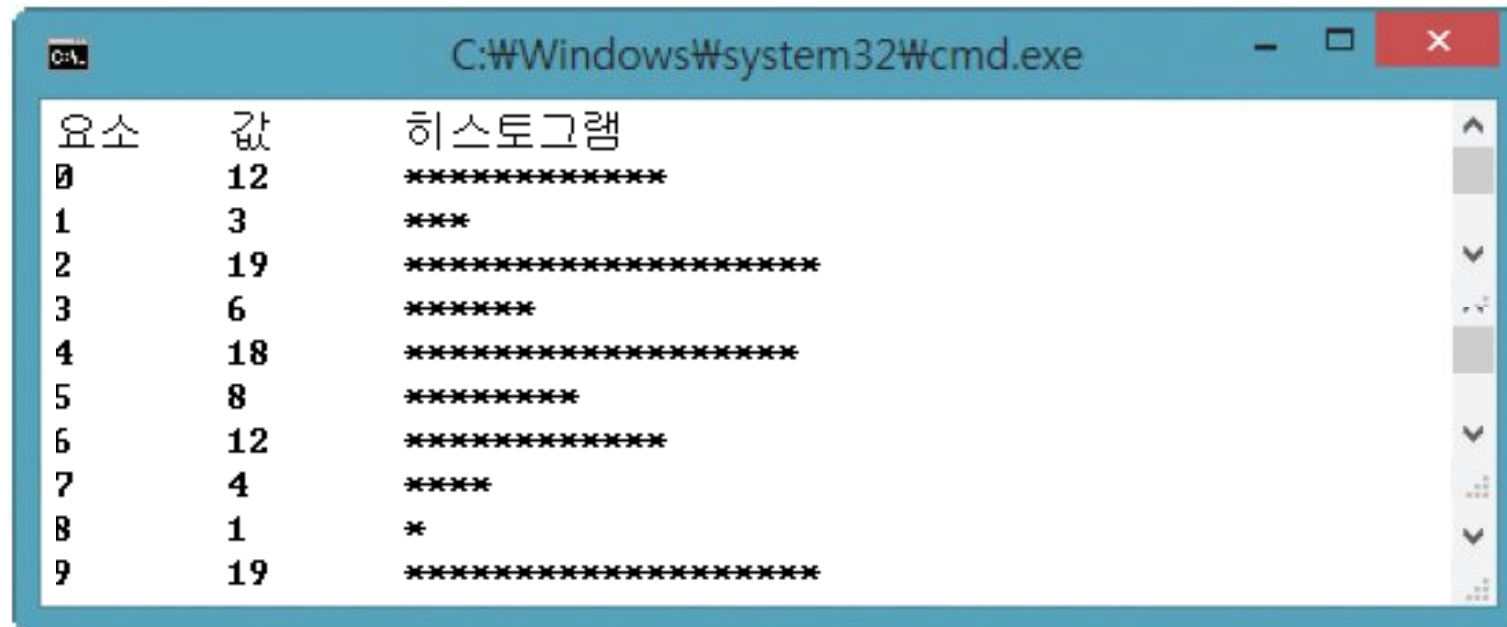
```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int scores[3];
06     int sum = 0;
07     int i;
08     double average;
09
10     for (i = 0; i < 3; i++) {
11         printf("게임 %d에서 선수의 득점은? ", i + 1);
12         scanf_s("%d", &scores[i]);
13     }
14
15     for (i = 0; i < 3; i++)
16         sum += scores[i];
17
18     average = ((double)sum / 3);
19     printf("평균 득점은 %f입니다.\n", average);
20
21     return 0;
22 }
```

예제 설명

다음 배열 `n[10]`에 들어 있는 데이터의 값에 따라서 막대 그래프를 그려보자.

```
int n[10] = { 12, 3, 19, 6, 18, 8, 12, 4, 1, 19 };
```

실행 결과



[예제 10] 히스토그램 그리기

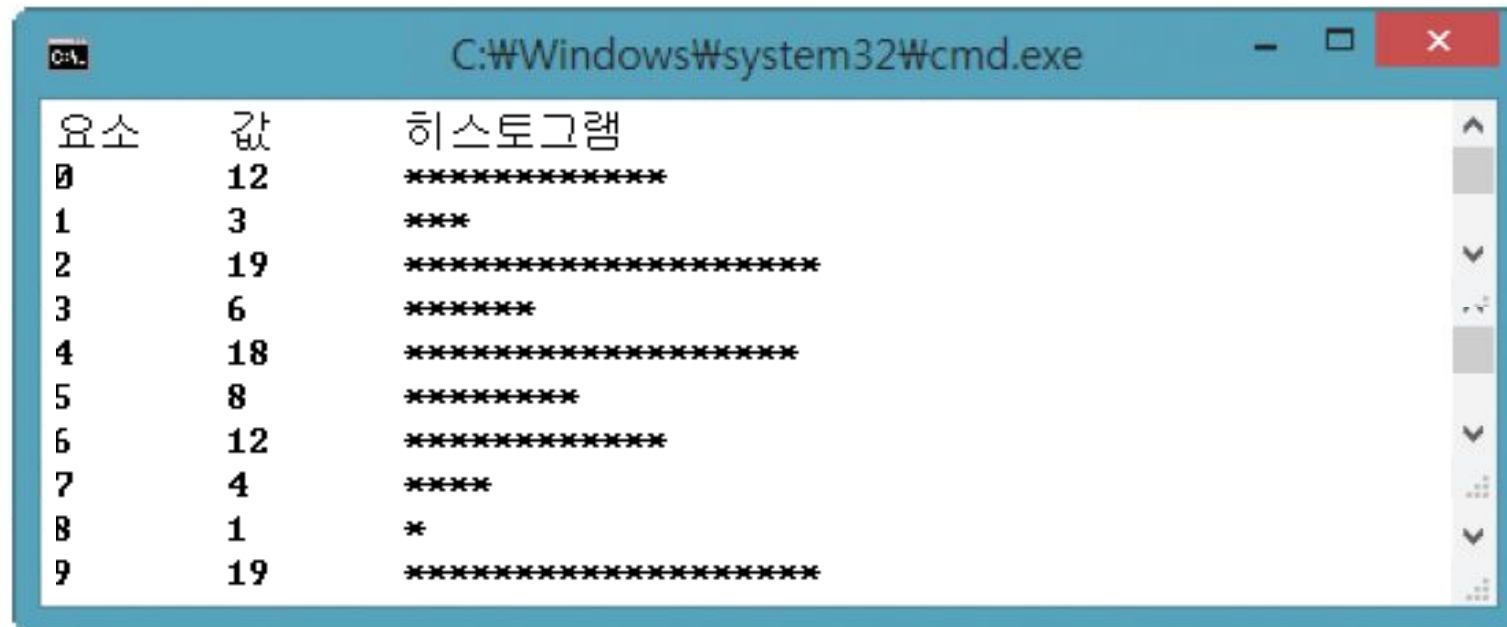
```
01 #include <stdio.h>
02 #define SIZE 10
03
04 int main(void)
05 {
06     int n[SIZE] = { 12, 3, 19, 6, 18, 8, 12, 4, 1, 19 };
07     int i,j;
08
09     printf("요소 값 히스토그램\n");
10     for(i=0; i<SIZE; i++){
11         printf("%2d %2d ", i, n[i]);
12         for(j=1; j<n[i]; j++)
13             printf("*");
14         printf("\n");
15     }
16
17     return 0;
18 }
```

예제 설명

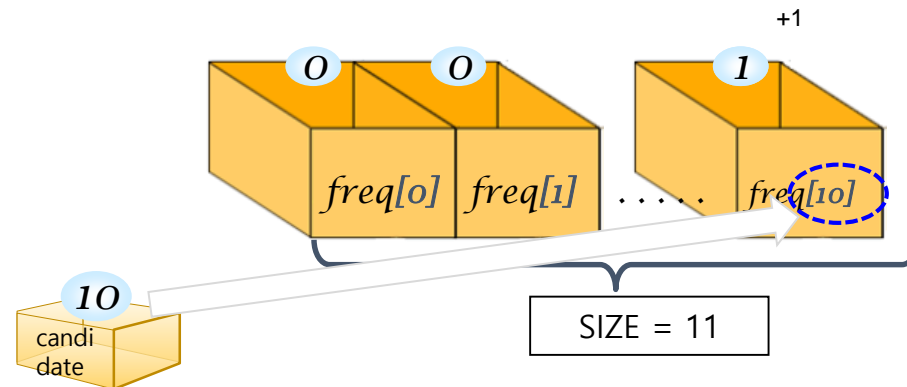
다음 배열 `n[10]`에 들어 있는 데이터의 값에 따라서 막대 그래프를 그려보자.

```
int n[10] = { 12, 3, 19, 6, 18, 8, 12, 4, 1, 19 };
```

실행 결과



- 배열 `freq[]`의 원소를 0으로 초기화한다.
- `while(1)`
 - 사용자로부터 후보자를 입력받는다.
 - `freq[candidate]++;`
- `freq` 배열의 내용을 출력한다.



[예제 11] 투표 집계하기

```
01 #include <stdio.h>
02 #define SIZE 11
03
04 int main(void)
05 {
06     int freq[SIZE] = {0}; //배열을 반드시 초기화 해야한다.
07     int i, candidate;
08
09     while(1)
10     {
11         printf("몇 번 연예인을 선택하시겠습니까?(종료-1): ");
12         scanf_s("%d",&candidate);
13         if(candidate<0) break;
14         freq[candidate]++;
15     }
16     printf("값 득표결과\n");
17     for(i=1; i<SIZE; i++)
18         printf("%2d %3d \n", i, freq[i]);
19     return 0;
20 }
```



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

```
int main( void)
{
    int    ch1 = '1';
    int    ch2 = 'a';
    int    ch3 = 256; // 아스키값 이상

    if ( isdigit(ch1))
        printf( "%c(x%03x)는 숫자 문자입니다.\n", ch1, ch1);
    else
        printf( "%c(x%03x)는 숫자 문자가 아닙니다.\n", ch1, ch1);

    if ( isdigit(ch2))
        printf( "%c(x%03x)는 숫자 문자입니다.\n", ch2, ch2);
    else
        printf( "%c(x%03x)는 숫자 문자가 아닙니다.\n", ch2, ch2);

    if ( isdigit(ch3))
        printf( "%c(x%03x)는 숫자 문자입니다.\n", ch3, ch3);
    else
        printf( "%c(x%03x)는 숫자 문자가 아닙니다.\n", ch3, ch3);

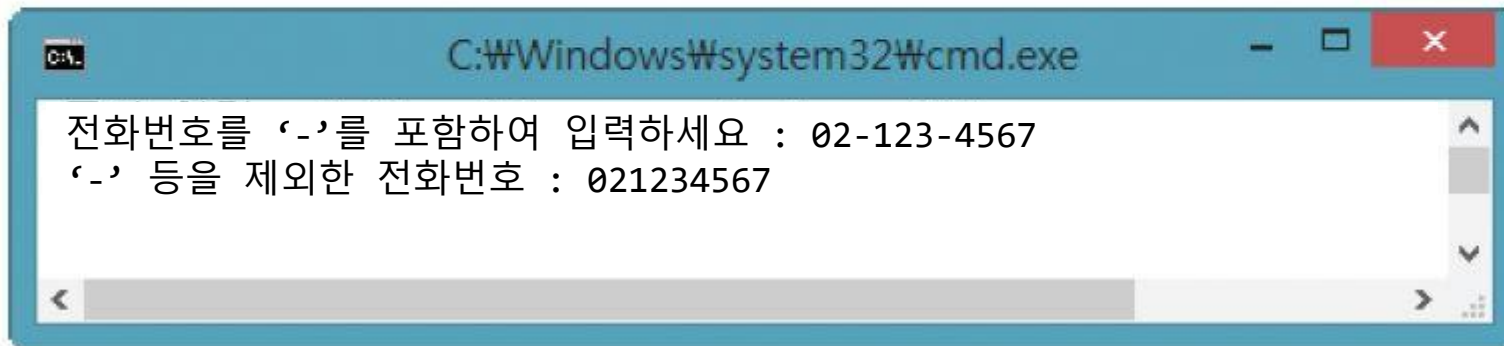
    return 0;
}
```

헤더	ctype.h
형태	int isdigit(int c)
인수	int c 판별할 문자
반환	0 != c가 숫자 문자, '0'~'9' 0 = c는 숫자 문자가 아님

1(x031)는 숫자 문자입니다.
a(x061)는 숫자 문자가 아닙니다.
(x100)는 숫자 문자가 아닙니다.

예제 설명 전화번호 '02-123-4567' 형식으로 입력받고, 그중에서 숫자만 출력하라
isdigit() 함수를 사용하라

실행 결과



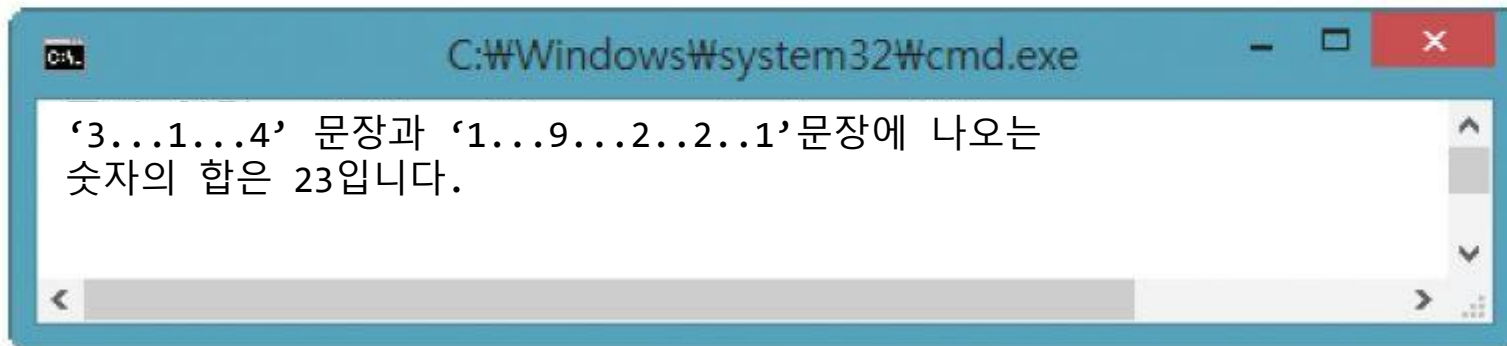
```
C:\Windows\system32\cmd.exe

전화번호를 '-'를 포함하여 입력하세요 : 02-123-4567
'-' 등을 제외한 전화번호 : 021234567
```

```
01 #include <stdio.h>
02 #include <ctype.h>
03 #include <string.h>
04
05 void main( ) {
06     int i,n=0;
07     char p[256], phone[256];
08
09     printf("전화번호를 '-'를 포함하여 입력하세요: ");
10     scanf_s("%s", phone, sizeof(phone));
11
12     for(i=0;i<(int)strlen(phone);i++) {
13         if (isdigit(phone[i])) {
14             p[n] = phone[i];
15             n++;
16         }
17     }
18     p[n] = '\0';
19     printf("\n'-'등을 제외한 전화번호 : %s\n",p);
20 }
```

예제 설명 '3...1...4' 문장과 '1...9...2..2..1'문장에 나오는 숫자의 합을 출력하라

실행 결과



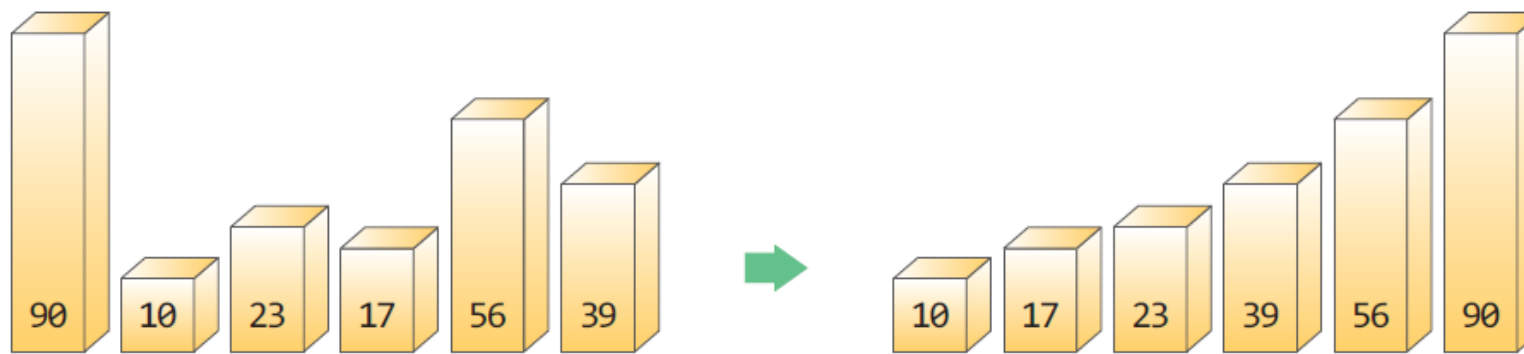
```
C:\Windows\system32\cmd.exe

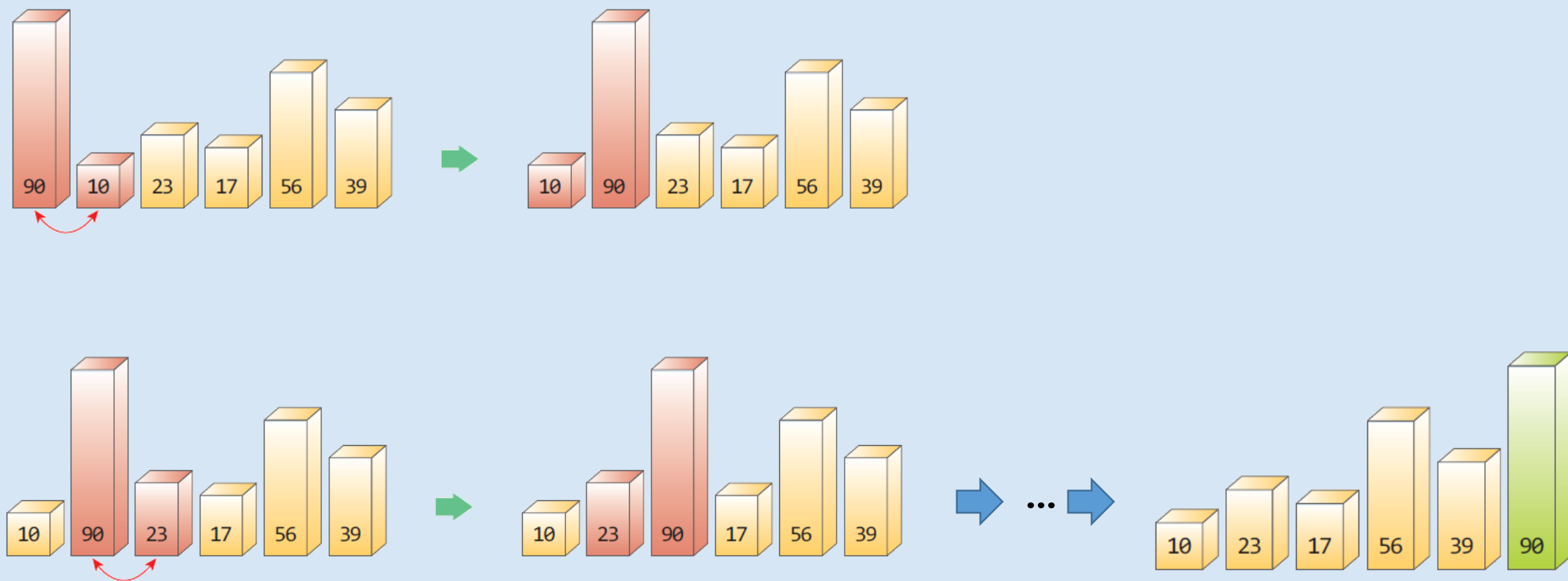
'3...1...4' 문장과 '1...9...2..2..1'문장에 나오는
숫자의 합은 23입니다.
```

The screenshot shows a Windows command prompt window titled 'C:\Windows\system32\cmd.exe'. The window contains the text: '3...1...4' 문장과 '1...9...2..2..1'문장에 나오는 숫자의 합은 23입니다. The text is displayed in a monospaced font, and there is a horizontal scrollbar at the bottom of the window.

```
01 #include <stdio.h>
02 #include <ctype.h>
03 #include <string.h>
04
05 void main( ) {
06     int i, sum = 0;
07     char s1[20] = "3...1...4";
08     char s2[20] = "1...9...2..2..1";
09
10     for(i=0;i<(int)strlen(s1);i++) {
11         if (isdigit((char)s1[i])) {
12             sum += (int)s1[i] - 48;
13             //printf("%d %d\n", (int)s1[i] - 48, sum);
14         }
15     }
16
17     for(i=0;i<(int)strlen(s2);i++) {
18         if (isdigit((char)s2[i])) {
19             sum += (int)s2[i] - 48;
20             //printf("%d %d\n", (int)s2[i] - 48, sum);
21         }
22     }
23     printf("\n'%s' 문장과 '%s' 문장에 나오는\n 숫자의 합은 %d입니다.\n", s1, s2, sum);
24 }
```

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





[예제 13] 버블정렬

```
01 #include <stdio.h>
02 #define SIZE 6
03
04 int main(void)
05 {
06     int i, k;
07     int list[SIZE] = { 90, 10, 23, 17, 56, 39 };
08
09     // 배열의 요소를 정렬한다.
10     for (k = 0; k < SIZE; k++) {
11         for (i = 0; i < SIZE-1; i++) {
12             if (list[i] > list[i + 1]) { // 크기 순이 아니면
13                 // 서로 교환한다.
14                 int tmp = list[i];
15                 list[i] = list[i + 1];
16                 list[i + 1] = tmp;
17             }
18         }
19     }
20     // 배열의 요소를 출력한다.
21     for (i = 0; i < SIZE; i++) {
22         printf("%d ", list[i]);
23     }
24     return 0;
25 }
```


[예제 13] 버블정렬

```
01 #include <stdio.h>
02 #define SIZE 6
03 int main(void)
04 {
05     int i, k, n;
06     int list[SIZE] = { 90, 10, 23, 17, 56, 39 };
07     for (n=0; n < SIZE; n++)
08         printf("%d ", list[n]);
09     printf("\n");
10     for (k = 0; k < SIZE; k++) {
11         for (i = 0; i < SIZE-1; i++) {
12             if (list[i] > list[i + 1]) { // 크기 순이 아니면
13                 int tmp = list[i];
14                 list[i] = list[i + 1];
15                 list[i + 1] = tmp;
16             }
17         }
18         for (n=0; n < SIZE; n++)
19             printf("%d ", list[n]);
20         printf("\n");
21     }
22     for (i = 0; i < SIZE; i++)
23         printf("%d ", list[i]);
24     return 0;
25 }
```