

# C프로그래밍

10 다차원 배열

# 공지

- **코딩테스트 방식 변경**

- ✓ 사전에 문제를 공지, 연습 후 수업시간에 구현/제출

- **코딩테스트 일정 변경**

- ✓ 6/9(월) 수업시간에 진행
- ✓ 기말고사 : 6/14(토) 진행

# 9 예제 review

## [예제 01] 입력된 문자열을 반대 순서로 출력

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

문자열을 입력하세요: Hanbit  
tibnaH

1. 입력과 출력에서는 `gets_s( )`와 `puts( )`를 사용
2. 두 개의 문자열 배열을 생성하고, 반복문을 이용하여 글자를 거꾸로 저장하고 출력
3. 문자열의 길이를 파악하기 위해서는 `strlen( )`를 사용

## [예제 01] 입력된 문자열을 반대 순서로 출력

```
.....  
void main( ) {  
    char str1[100], str2[100];  
    int count;                //문자열의 길이를 저장하기 위한 변수  
    printf("문자열을 입력하세요 : ");  
    gets_s(str1, sizeof(str1)); //gets_s( )를 이용하여 문자열 입력 받음  
    count = strlen(str1);  
    for(int i = 0; i < count; i++){ //문자열을 반대로 저장하기 위한 반복문  
        str2[i] = str1[count - (i+1)]; //str2[ ]에 문자열을 반대로 저장  
    }  
    str2[count] = '\0';        //str2[ ]에 저장된 문자들을 문자열로 만들어 줌  
    puts(str2);                //str2[ ]에 저장된 문자열을 출력  
}
```

01

# 다차원 배열



# 2차원 배열의 기본 개념 (1/7)

## • 1차원 배열의 개념

- ✓ 같은 데이터 타입의 요소 여러 개를 연결
- ✓ 'int aa[3];'으로 정의했다면 aa[0], aa[1], aa[2]라는 요소가 생성

int aa[행]

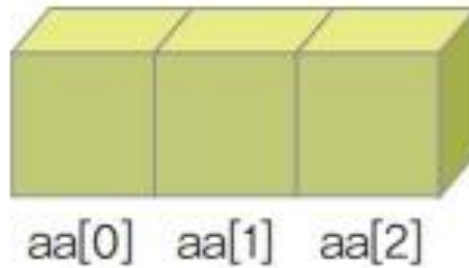


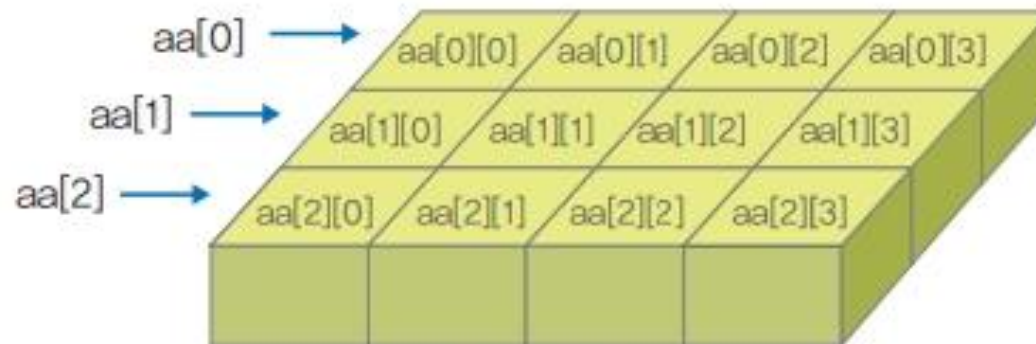
그림 8-15 1차원 배열의 개념

## 2차원 배열의 기본 개념 (2/7)

### • 2차원 배열의 개념

- ✓ 1차원 배열의 각 항목에 또 다른 1차원 배열을 넣는 구조
- ✓ 1차원 배열을 확장해서 2차원 배열 'int aa[3][4]'를 정의  
→ aa[0], aa[1], aa[2]의 요소에 크기가 4인 배열을 넣는 형태
- ✓ 이때 앞의 3은 가로줄 수를, 뒤의 4는 세로줄 수를 의미 : 3행 4열
- ✓ 테이블 형태의 데이터를 저장하는 용도

int aa[행][열]



전체 배열 이름: aa

그림 8-16 2차원 배열의 개념



# 2차원 배열의 기본 개념 (3/7)

## • 2차원 배열의 간단한 사용 예

기본 8-14 2차원 배열 사용 예 1

8-14.c

```
01 #include <stdio.h>
02
03 void 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; —— 각 요소에 값을 대입한다.
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
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

aa[0][0]부터 aa[2][3]까지 출력

1	2	3	4
5	6	7	8
9	10	11	12

## 2차원 배열의 기본 개념 (4/7)

### • 반복문을 이용한 2차원 배열 사용 예

응용 8-15 2차원 배열 사용 예 2

8-15.c

```
01 #include <stdio.h>
02
03 void main( )
04 {
05     int aa[3][4];           —— 2차원 배열과 첨자 변수를 선언한다.
06     int i, k;
07
08     int val=1;              —— 배열에 들어갈 값을 초기화한다.
09
10     for( i=0; i < 3; i++ )  —— 바깥 for문을 세 번 반복한다.
11     {                      —— 즉 앞 첨자가 행 단위로 변경된다.
12         for( k=0; k < 4; k++ ) —— 안쪽 for문을 네 번 반복한다.
13         {                  —— 즉 뒤 첨자가 열 단위로 변경된다.
14             aa[i][k] = val; —— 배열에 val 값을 입력한 후 1 증가시킨다.
15             val++;
16         }
17     }
```

## 2차원 배열의 기본 개념 (5/7)

### • 반복문을 이용한 2차원 배열 사용 예 (cont'd)

```
18
19  printf("aa[0][0]부터 aa[2][3]까지 출력 \n");
20
21  for( i=0; i < 3; i++ )      — 입력과 동일한 개념으로 12회 출력한다.
22  {
23      for( k=0; k < 4; k++ )
24      {
25          printf("%3d ", aa[i][k] );
26      }
27      printf("\n");          — 한 행을 출력한 후 줄을 넘긴다.
28  }
29 }
```

aa[i][k] ++k :k<4; k=0; i=0; aa[0][k]

#### 실행 결과

aa[0][0]부터 aa[2][3]까지 출력

```
1  2  3  4
5  6  7  8
9 10 11 12
```

# 2차원 배열의 기본 개념 (6/7)

## • 2차원 배열 초기화 예

### 기본 8-16 2차원 배열의 초기화 예

```
01 #include <stdio.h>
02
03 void 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
```

int aa[3][4] =  
{  
aa[0] → { 1, 2, 3, 4 } , ← 1행(aa[0])과 2행(aa[1])을 구분한다.  
aa[1] → { 5, 6, 7, 8 } , ← 2행(aa[1])과 3행(aa[2])을 구분한다.  
aa[2] → { 9, 10, 11, 12 } ← 마지막 행이므로逗가 없다.  
};

그림 8-17 2차원 배열의 초기화

—— 2차원 배열을 초기화한다.

2차원 배열은 1차원 배열의 각 항목에 각각 1차원 배열을 저장하는 구조라는 것을 기억

# 2차원 배열의 기본 개념 (7/7)

## • 2차원 배열 초기화 예 (cont'd)

```
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 }
```

—— 2차원 배열에 저장된 값을 출력한다.

### 실행 결과

aa[0][0]부터 aa[2][3]까지 출력

```
1 2 3 4
5 6 7 8
9 10 11 12
```

# 3차원 배열의 기본 개념 (1/2)

## • 3차원 배열의 개념

- ✓ 2차원 배열 위에 또 다른 2차원 배열을 쌓은 것
- ➔ 1차원 배열의 각 항목에 2차원 배열을 저장하는 방식
- ➔ 2차원 배열의 각 항목에 1차원 배열을 저장

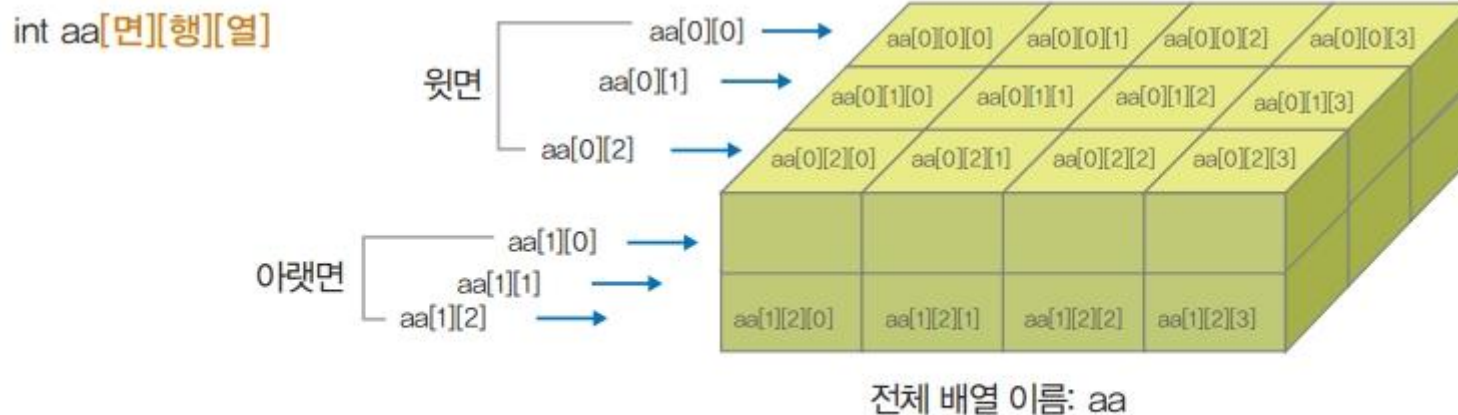


그림 8-18 3차원 배열의 개념



## 3차원 배열의 기본 개념 (2/2)

### • 3차원 배열의 초기화

- ✓ 2차원 배열의 초기화를 한번 더 하는 개념
- ✓ 콤마로 분리하고, 전체를 다시 블록으로 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

그림 8-19 3차원 배열의 초기화

예 제

## [예제 01] 대문자와 소문자의 변환

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

**실행 결과**

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

1. 입력에서는 `gets_s( )`를 사용
2. 두 개의 문자열 배열을 생성하고, 반복문을 이용하여 변환된 결과를 저장하고 출력
3. 문자열의 길이를 파악하기 위해서는 `strlen( )`를 사용
4. 대문자 → 소문자 : 소문자와 대문자와의 아스키코드 값 차이(32) 만큼을 플러스
5. 소문자 → 대문자 : 소문자와 대문자와의 아스키코드 값 차이(32) 만큼을 마이너스

## [예제 02] 문자열 병합 함수 구현

- (1) 두 개의 입력 받은 문자열이 서로 다를 경우 두 문자열을 병합하여 출력하고,
- (2) 같을 경우 병합하지 않고 하나의 문자열만 출력하는 프로그램 작성

첫 번째 : hello  
두 번째 : world  
출력 결과 : helloworld

첫 번째 : hello  
두 번째 : hello  
출력 결과 : hello

1. 입력에서는 `gets_s( )`를 사용
2. 두 개의 문자열 배열을 생성해서 사용
3. 문자열을 비교하기 위해서는 `strcmp( )`를 사용
4. 문자열을 병합하기 위해서는 `strcat( )`를 사용

## [예제 03] 구구단의 결과를 배열에 저장

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

**실행 결과**

```
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
```

**1. 구구단 계산 결과를 2차원 배열에 먼저 저장하고, 다음에 출력**

## [예제 04] 구구단의 결과를 배열에 저장

영희, 철수, 영철, 영수 등 4명의 학생의 국어, 영어, 수학, 과학 점수(1~9점)를 입력 받아 배열에 저장하고, 각 학생의 총점과 과목별 총점을 계산하여 저장하고, 출력하는 프로그램을 작성하시요.

### <1단계> 점수를 입력 받아서 저장

	국어	영어	수학	과학	학생별 총점
영희	2	3	4	5	
철수	5	5	4	7	
영철	3	7	2	4	
영수	8	6	2	9	
과목별 총점					

### <2단계> 총점을 계산/저장하고 출력

	국어	영어	수학	과학	학생별 총점
영희	2	3	4	5	14
철수	5	5	4	7	21
영철	3	7	2	4	16
영수	8	6	2	9	25
과목별 총점	18	21	12	25	

1. 입력 받은 점수와 총점은 5X5 2차원 배열에 저장하여 출력하시요.
2. (옵션) 학생의 이름과 과목명도 배열에 저장하여 관리하자.



Q & A