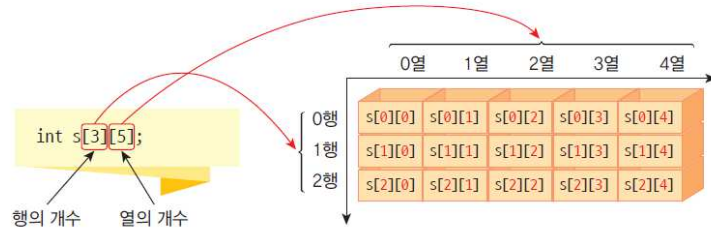




2차원 배열

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

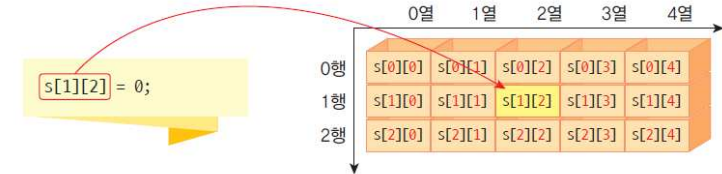


© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



2차원 배열에서 인덱스



© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



2차원 배열의 활용

```
#include <stdio.h>
#include <stdlib.h>
#define ROWS 3
#define COLS 5
int main(void)
{
    int s[ROWS][COLS]; // 2차원 배열 선언
    int i, j; // 2개의 인덱스 변수
    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(" %02d ", s[i][j]);
        printf("\n");
    }
    return 0;
}
```



© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express 1/12(컴프 3주차)



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
};
```



© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



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
};
```



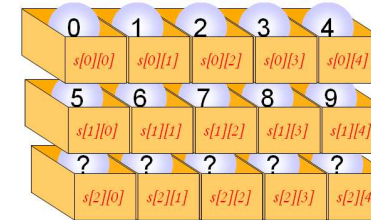
© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



2차원 배열의 초기화

```
int s[ ][5] = {
    0, 1, 2, 3, 4, // 첫 번째 행의 원소들의 초기값
    5, 6, 7, 8, 9, // 두 번째 행의 원소들의 초기값
};
```



© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



예제

- 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

© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express 2/12(컴프 3주차)



2차원 배열의 초기화

```
#include <stdio.h>
#define ROWS 3
#define COLS 5

int main(void) {

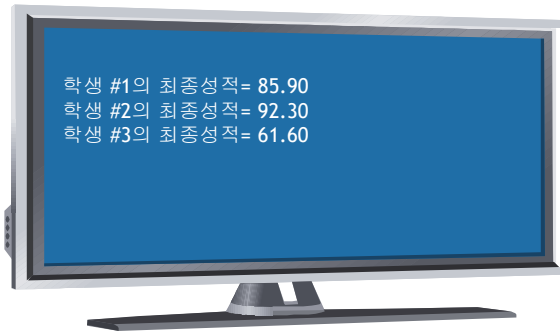
    int a[ROWS][COLS] = { { 87, 98, 80, 76, 3 },
                          { 99, 89, 90, 90, 0 },
                          { 65, 68, 50, 49, 0 }
    };

    int i;
    for (i = 0; i < ROWS; i++) {
        double final_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 %i의 최종성적 = %10.2f \n", i + 1, final_scores);
    }
    return 0;
}
```

© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express

실행 결과



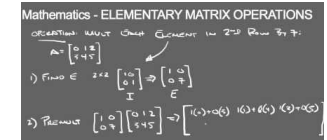
© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express

행렬

- 행렬(matrix)은 자연과학에서 많은 문제를 해결하는데 사용

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



© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express

다차원 배열을 이용한 행렬의 표현

```
#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];
```

© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express 3/12(컴프 3주차)

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;
```

중첩 for 루프를 이용하여 행렬 A의 각 원소들과 행렬 B의 각 원소들을 서로 더하여 행렬 C에 대입한다.



© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



2차원 배열을 함수로 전달하기

```
#include <stdio.h>

#define YEARS      3
#define PRODUCTS   5

int sum(int grade[][PRODUCTS]);

int main(void)
{
    int sales[YEARS][PRODUCTS] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };
    int total_sale;

    total_sale = sum(sales);
    printf("총매출은 %d입니다.\n", total_sale);

    return 0;
}
```

© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



2차원 배열을 함수로 전달하기

```
int sum(int grade[][PRODUCTS])
{
    int y, p;
    int total = 0;

    for(y = 0; y < YEARS; y++)
        for(p = 0; p < PRODUCTS; p++)
            total += grade[y][p];

    return total;
}
```

© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



중간 점검

- 다차원 배열 int a[3][2][10]에는 몇 개의 원소가 존재하는가?
- 다차원 배열 int a[3][2][10]의 모든 요소를 0으로 초기화하는 문장을 작성하시오.



© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express 4/12(컴프 3주차)



lab: 영상 처리

- 디지털 영상은 픽셀들의 2차원 배열이라 할 수 있다.

```
int image[8][16] = {
    { 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 },
    { 1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1 },
    { 1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1 },
    { 1,1,1,0,0,0,1,1,0,0,1,1,1,1,1,1 },
    { 1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1 },
    { 1,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1 },
    { 1,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1 },
    { 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 };
```



© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



실행 결과

```
C:\WINDOWS\system32\cmd.exe
변환전 이미지
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****

변환후 이미지
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C 언어 Express



```
#include <stdio.h>

void display(int image[8][16])
{
    for (int r = 0; r < 8; r++) {
        for (int c = 0; c < 16; c++) {
            if (image[r][c] == 0)
                printf("*");
            else
                printf("_");
        }
        printf("\n");
    }
}

void inverse(int img[8][16])
{
    for (int r = 0; r < 8; r++) {
        for (int c = 0; c < 16; c++) {
            if (img[r][c] == 0)
                img[r][c] = 1;
            else
                img[r][c] = 0;
        }
    }
}
```

© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C 언어 Express



```
int main(void)
{
    int image[8][16] = {
        { 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 },
        { 1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1 },
        { 1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1 },
        { 1,1,1,0,0,0,1,1,0,0,1,1,1,1,1,1 },
        { 1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1 },
        { 1,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1 },
        { 1,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1 },
        { 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 } };

    printf("변환전 이미지\n");
    display(image);
    inverse(image);
    printf("\n\n변환후 이미지\n");
    display(image);
    return 0;
}
```

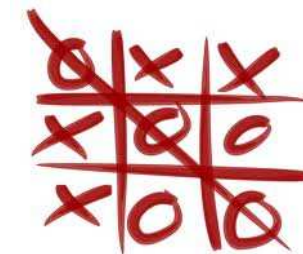
© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C 언어 Express 5/12(컴포 3주차)



mini project: tic-tac-toe

- tic-tac-toe 게임은 2명의 경기자가 오른쪽과 같은 보드를 이용하여서 번갈아가며 O와 X를 놓는 게임이다.
- 같은 글자가 가로, 세로, 혹은 대각선 상에 놓이면 이기게 된다.



© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C 언어 Express

실행 결과

```
C:\WINDOWS\system32\cmd.exe
(x, y) 좌표: 0 0
X
|
|
|
|
|
(x, y) 좌표: 1 1
X
|
|
0
|
|
(x, y) 좌표:
```

© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express

알고리즘

```

보드를 초기화한다.
while(1)
    보드를 화면에 출력한다.
    사용자로부터 좌표 x, y를 받는다.
    if (board[x][y]가 비어 있으면)
        if( 현재 경기자가 'X'이면 )
            board[x][y] = 'X'
        else
            board[x][y] = 'O'
    else
        오류 메시지를 출력한다
    
```

© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



^^

```
#include <stdio.h>
int main(void)
{
    char board[3][3];
    int x, y, k, i;
    // 보드를 초기화한다.
    for (x = 0; x < 3; x++)
        for (y = 0; y < 3; y++) board[x][y] = ' ';

    // 사용자로부터 위치를 받아서 보드에 표시한다.
    for (k = 0; k < 9; k++) {
        printf("(x, y) 좌표: ");
        scanf("%d %d", &x, &y);
        board[x][y] = (k % 2 == 0) ? 'X' : 'O'; // 순번에 따라 'X', 'O'중
    }
}
```

© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express 6/12(컴프 3주차)



^^

```

// 보드를 화면에 그린다.
for (i = 0; i < 3; i++) {
    printf("---|---|---\n");
    printf("%c | %c | %c\n", board[i][0], board[i][1],
board[i][2]);
}
printf("---|---|---\n");
return 0;
}
```

© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



도전문제

- (1) 위의 코드를 실행하면 상대방이 놓은 곳에 다시 놓을 수 있다. 이것을 방지하는 코드를 추가하라.
- (2) 보드를 분석하여서 게임이 종료되었는지를 검사하는 함수를 추가하라.
- (3) 컴퓨터가 자동으로 다음 수를 결정하도록 프로그램을 변경하라. 가장 간단한 알고리즘을 사용한다. 예를 들면 비어 있는 첫 번째 좌표에 놓는다.

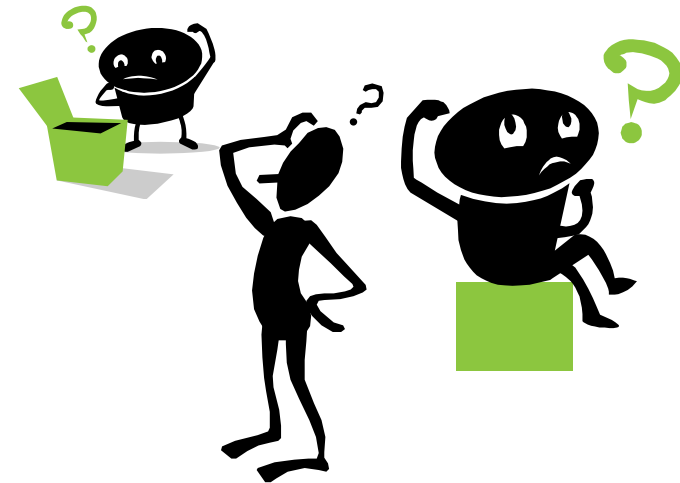


© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express



Q & A



© 2012 생능출판사 All rights reserved

쉽게 풀어쓴 C언어 Express

프론 수업 #14. 배열 revisited : 2차원 배열

1차원 배열: int list[5];
2차원 배열: int table[5][3];

table

■ 2차원 배열의 접근 방법:

table[0][0] 첫 원소
table[0][1] ?
...
table[i][j]
...
table[4][2] 마지막 원소

■ 2차원 배열 원소의 초기화

1. 방법1: 원소별로 대입문 사용	2. 방법2: for문 안에서 대입문 사용	3. 방법3: 선언과 동시에 초기화
<pre>int table[5][3]; table[0][0] = 0; table[0][1] = 0; table[0][2] = 0; table[1][0] = 0; . . . table[4][2] = 0;</pre>	<pre>int table[5][3];</pre>	<pre>int table[5][3] = {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}, {0, 0, 0}, {0, 0, 0}}; 혹은 int table[5][3] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}; 혹은 int table[5][3] = {0}; // 나머지는 다 0으로 채움</pre>

1차원 배열의 경우
int list[] = {1, 2, 3, 4, 5}; // 배열 list의 크기는 5

2차원 배열의 경우
int table[][3] = {1, 2, 3, 4, 5, 6}; // 행의 개수는 ?, 열의 개수는 생략할 수 없음

■ 배열 원소의 사용: 1차원 배열과 같음

■ sizeof 연산자로 배열의 크기를 알 수 있음

```
// 1차원 배열
sizeof(list) / sizeof(list[0])

// 2차원 배열
sizeof(table) / sizeof(table[0]) // sizeof(table[1]), sizeof(table{2})도 모두 같은 값
```

■ 2차원 배열의 연습: int table[5][3]; 으로 선언되었다고 가정하고 아래의 연습문제를 풀자.

□ 연습0

위와 같은 값을 갖도록 2차원 배열 table의 값을 초기화하라. 초기화하는데 위의 방법2를 사용하라.

0	1	2
1	2	3
2	3	4
3	4	5
4	5	6

□ 연습1

위와 같은 값을 갖도록 2차원 배열 table의 값을 초기화하라. 초기화하는데 위의 방법2를 사용하라.

0	1	2
10	11	12
20	21	22
30	31	32
40	41	42

for (i = 0; i < 5; i++)
for (j = 0; j < 3; j++)

□ 연습2

table안의 모든 값을 더하여 출력하는 코드 부분을 완성하라.

실행예:
315
for (i = 0; i < 5; i++)
for (j = 0; j < 3; j++)

□ 연습3

첫 행을 더하여 출력하는 코드 부분을 완성하라. 더한 값은 3(= 0 + 1 + 2)이다.

실행예:
3

□ 연습4

모든 행을 각각 더하여 출력하는 코드 부분을 완성하라.

실행예:
3
33
63
93
123

□ 연습5

첫 열을 더하여 출력하는 코드 부분을 완성하라.

실행예:
100

□ 연습6

모든 열을 각각 더하여 출력하는 코드 부분을 완성하라.

실행예:
100
105
110

■ 2차원 배열의 함수 매개변수 전달

```
void printTable(                ) // 정의
{
```

```
}
```

```
int main(void)
{
    int table1[3][5] = {{1, 2, 3, 4, 5}, {10, 20, 30, 40, 50}, {100, 200, 300, 400, 500}};
    int table2[2][5] = ({1, 2, 3}, {10}); // 어떤 값이 들어가나?

    printTable(table1, 3); // 호출: 2차원 배열의 이름, 행의개수
    printTable(table2, 2); // 호출: 2차원 배열의 이름, 행의개수
}
```


LAB16 2차원 배열

- **LAB16_1 (2차원 배열 연습)** 아래의 실행 결과를 갖도록 가)부터 차례로 코드를 추가하여 프로그램하라.

가)아래와 같은 값을 갖도록 2차원 배열 table을 선언과 동시에 초기화한 후 그대로 출력하는 프로그램을 작성하라.

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

나)table 요소들의 값을 아래와 같이 배정하는 코드(for 문 사용)를 작성하라. (힌트 열과 행의 index를 더한 값을 해당 요소가 갖게 된다). 그대로 출력하는 프로그램을 while 문을 사용하여 작성하라.

0	1	2
1	2	3
2	3	4
3	4	5
4	5	6

다)각 요소의 합을 구하여 출력하는 프로그램을 작성하라.

```

G:\ "F:\W++\201001수업W-2010
>
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
>
0 1 2
1 2 3
2 3 4
3 4 5
4 5 6
>
Total is 45
Press any key to continue

```

- **LAB16_2 (1차원 배열 + Visual Studio 디버깅 연습)**아래의 간단한 프로그램을 통해 Visual Studio의 debugging 기능에 대해서 간단히 알아 본다. (실습 시간에 교수님의 설명에 따라 F5, F9, F10 키의 기능을 익힌다).

```

#include <stdio.h>

void main(void)
{
    int num[] = {-2, -1, 0, 1, 2};
    int sum = 0, i = 0

    while (i < 5) {
        sum += num[i]
        i++;
    }
}

```

- 위의 프로그램을 컴파일 한 후에, F10을 계속 눌러 라인 12에 오도록 해보자. (F10은 프로그램의 한 라인씩 수행해 주는 키이며, 이 키를 수행 할 때 마다 외쪽의 노란색 화살표가 지시하는 라인이 수행된다.).
- 라인 12에 도달 한 후에 다시 F10을 누르면 다소 이상한 윈도우가 뜰 것이다. 이때 cancel을 선택하라. 그러면 다시 이상한 코드 들이 보일 것이다. 당황해 하지 말고 F5키를 눌러 프로그램을 종료시킨다. (F5는 breakpoint가 있는 곳까지 프로그램을 계속해서 수행시키는 키이다. 이 키도 매우 자주 사용되므로 반드시 암기한다)
- 다시 F10을 사용해서 다시 프로그램을 한 라인씩 수행한다(프로그램이 다시 수행된다) 이 때 VS 아래 창에 변수 값이 변하면서 빨간 색으로 표시되는 것을 확인해 보시오. 라인 12까지 노란색 화살표가 이동하도록 한 후에는 F5를 눌러 프로그램을 종료시킨다.
- 자, 이번에는 라인 9에 마우스를 포인트를 위치시킨 후 F9를 누른다. 그러면 왼쪽에 빨간색 동그라미가 생길 것이다. 이것에 breakpoint(이하 BP)이며, 이 지점에서 프로그램이 일시 정지할 것이다 이를 확인하기 위해 이제 F5를 눌러보자. 그러면 라인 9에 노란색 화살표가 위치하여 라인 9에 정지하는 것을 볼 수 있을 것이다.
- 다시 F5를 눌러 라인 9까지 반복 수행되는 것을 확인하자. 이제, 다시 마우스 포인트를 라인 9에 위치시킨 후 F9를 눌러보자. 빨간 색 표시가 꺼진다. 이제 이 라인에 breakpoint가 소멸되었다. 이어서 다시 F5를 눌러보자. 프로그램이 완료될 것이다.
- 바) 현재는 BP가 없을 것이다. 그럼 라인 12에 bp를 생성해 보자. 그리고 나서 F5를 수행해 보자. 라인 12에 노란색 화살표가 있을 것이다. 그럼 다시 한번 라인 8-11의 코드를 수행시켜 보자. (프로그램을 재수행 시키지 말고). 이를 위해 라인 8에 마우스 포인트를 위치시키고 오른쪽 마우스 버튼을 누른다. 그러면 풀다운 메뉴가 보이는데, 여기서 “Set next statement”를 선택하자. 그러면 현재 수행 시킬 라인이 원하는 곳으로 바뀐다. (물론 앞쪽 라인으로도 이동시킬 수 있다).
- 사) 바)에서는 이미 변수 i 값과 sum값이 변했기 때문에, 동일한 결과를 수행하려면 라인 6부터 재수행 하는 것이 옳을 것이다. 라인 6부터 재수행 해보자. (디버깅 시에 자주 원하는 곳으로 실행 라인을 바꿀 필요가 생긴다. 그럴 때를 대비해서 이 기능을 잘 익혀두자)

주의) F9, F10, F5의 기능이 익숙해 질 때까지 반복 수행한다. 학생 여러분은 앞으로 컴파일 한 후에 “!” (빨간 느낌표) 키를 사용하여 프로그램을 한 번에 수행하지 마시오(never!). 반드시 F10을 통해 한 라인씩 수행하여 프로그램의 로직을 정확히 이해하도록 하시오 (특히 교재의 연습문제 프로그램) 수행 결과를 모두 확인한 후에 “!” 키를 사용하시오.

■ LAB16_3 아래와 같이 text 변수가 있다고 하자. 이때 아래의 지시들을 수행해 보시오.

```
char text[][10] = { "ABC", "abcd", "Hello" };
```

- 가) 변수 text의 메모리 크기는 얼마인가 적어보시오. (자신이 예상한 값을 먼저 적고, sizeof 연산자를 이용하여 체크해 본다)
- 나) text 변수에 어떻게 값들이 저장되어 있는지 그려본다.
- 다) 다음과 같은 출력을 내는 프로그램을 작성해 보시오. (힌트: 각 스트링의 끝은 널 문자로 끝나며 널문자는 'W0' 으로 표현하며, 널 문자의 아스키 코드 값은 0이다.)

```
C:\Windows\system32\cmd.exe
text의 메모리 크기는 30이다.
1번째 스트링: ABC, 아스키 코드값: 65 66 67
2번째 스트링: abcd, 아스키 코드값: 97 98 99 100
3번째 스트링: Hello, 아스키 코드값: 72 101 108 108 111
계속하려면 아무 키나 누르십시오 . . .
```

대략의 코드 형태는 아래와 같음.

```
#include <stdio.h>

int main(void)
{
    char text[][10] = {"ABC", "abcd", "Hello"};
    int i, j;

    printf("가) text의 메모리 크기는 %d이다.\n", sizeof(text));
    for (i = 0; i < 3; i++)
    {
        printf("%d번째 스트링: %s, ", i + 1, text[i]);
        printf("아스키 코드값: ");
        // 여기에 코드를 삽입

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

■ LAB16_4 다음과 같은 2차원 배열에 대해 아래의 지시를 수행해 보시오. 가),나),다),라)의 순서로 코딩하시오(

```
int test[4][3] = { {8, 7}, {6, 5}, {1, 2}, {3, 4}};
```

- 가) 앞의 선언 및 초기화 구문에서 초기화되지 않은 엘리먼트는 몇 개이고 이들 엘리먼트들은 어떤 것들인지 말해 보시오. test의 메모리 구조를 2차원 형태로 그림을 그려보시오.

- 나) 아래와 같은 형태의 구문을 사용하여 배열 test를 각 row 별로 출력하는 프로그램을 완성해 보시오.

```
printf("배열 test의 엘리먼트를 출력합니다\n");
for (i = 0; i < 4; i++)
{
    printf("\n%d번째 row: " , i + 1);
    for (j = 0; j < 3; j++)

}

}
```

```
C:\Windows\system32\cmd.exe
배열 test의 엘리먼트를 출력합니다.
1번째 row: 8 7 0
2번째 row: 6 5 0
3번째 row: 1 2 0
4번째 row: 3 4 0
다>
배열 test의 엘리먼트를 출력합니다.
1번째 row: 8 7 -1
2번째 row: 6 5 -1
3번째 row: 1 2 -1
4번째 row: 3 4 -1
라>
Total is 32
Average is 2.67
마>
1번째 row에서의 최대값: 8
2번째 row에서의 최대값: 6
3번째 row에서의 최대값: 2
4번째 row에서의 최대값: 4
최대값의 합: 20
계속하려면 아무 키나 누르십시오 . . .
```

- 다) 아래와 같은 형태의 for 문을 사용하여 test 배열에서 초기화 되지 않은 4개의 엘리먼트들 값을 모두 -1로 수정한후 다시 한번 나)의 코드를 이용하여 출력하시오.

```
for (i = 0; i < 4; i++)
    _____코드 한 줄 추가_____
```

- 라) 앞의 다)를 수행한 후에 test의 모든 엘리먼트(12개) 값의 합과 그 평균을 출력하는 프로그램을 작성해 보시오.. for 반복문과 적당한 변수를 추가하여 프로그램을 완성해 보시오. (주의: 전체 합으로는 32을, 평균 값으로는 2.67을 출력함)

- 마) 앞의 라)를 수행한 후에 다음과 같은 기능을 수행하는 프로그램을 작성해 보시오.

- test 배열의 각 row에 있는 엘리먼트 중 최대값을 출력.
- 출력된 최대값 4개의 합을 출력

HW16 다자원 배열

- HW16.1 아래와 같이 int 형 2 차원 배열을 선언하여 그 안에 구구단의 결과를 9 단까지 저장하자. 다음과 같은 형태로 저장되어야한다. 그리고 나서 제대로 저장되었음을 확인하기 위해서 구구단표를 출력한후 사용자 입력으로 숫자를 읽어들여서 선언한 2 차원 배열을 이용하여 해당 구구단을 출력하는 프로그램을 작성하라.

2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

실행결과

```

C:\> "C:\WDocuments and Settings\Wuser\WMy
--구구단 표는 다음과 같습니다.
  2  4  6  8 10 12 14 16 18
  3  6  9 12 15 18 21 24 27
  4  8 12 16 20 24 28 32 36
  5 10 15 20 25 30 35 40 45
  6 12 18 24 30 36 42 48 54
  7 14 21 28 35 42 49 56 63
  8 16 24 32 40 48 56 64 72
  9 18 27 36 45 54 63 72 81
--원하는 구구단은? 6
  6 12 18 24 30 36 42 48 54
Press any key to continue_

```

- HW16.2 문제 2)의 배열 A(arr1)을 선언과 동시에 초기화하고 이 배열 A의 값을 대입문을 이용하여 배열 B(arr2)와 배열 C(arr3)에 값을 넣어서 출력하는 프로그램을 작성하라.

1	2	3	4					
5	6	7	8					

1	5
2	6
3	7
4	8

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

배열 A 배열 B 배열 C

실행결과

```

C:\> "C:\WDocuments and Settings\Wuser\WMy
--배열 A
  1  2  3  4
  5  6  7  8

--배열 B
  1  5
  2  6
  3  7
  4  8

--배열 C
  1  2  3  4  5  6  7  8
Press any key to continue_

#include <stdio.h>
int main(void)
{
    int arr1[2][4] = {1, 2, 3, 4, 5, 6, 7, 8};
    int arr2[4][2];
    int arr3[8];
    int i, j; // 더 이상의 인덱스 변수는 사용 말라.

    /* 이동(복사) A → B*/

    /* 이동(복사) A → C*/

    /* A, B, C 출력 */

    return 0;
}

```

HW16.3

4명의 학생의 중간고사, 기말고사, 숙제의 성적이 아래와 같이 배열에 저장되어 있습니다. 이를 이용하여 각 학생의 총점과 고사별 총점을 구하여 화면에 출력하는 프로그램을 작성하세요. 교재의 답처럼 함수를 여러개 쓰지 말고 main 함수 하나로 프로그래밍하라. 이것은 조금 어려운 문제입니다.

'이 문제를 해결할 수 있다면, 여러분은 2 차원배열을 맘대로 충분히 주무를수 있는 능력을 지니고 있는 것이다'

- a) 선언과 같이 아래와 같은 배열로 초기화한후, 이를 출력하고

10	15	20	0
30	35	20	0
60	65	20	0
90	95	20	0
0	0	0	0

- b) 아래와 같이 각 학생의 총점과 고사별 총점을 구한후 이를 출력합니다.

10	15	20	45
30	35	20	85
60	65	20	145
90	95	20	205
190	210	80	0

실행결과:

```

C:\> "C:\WDocuments and Settings\W
a>
 10 15 20 0
 30 35 20 0
 60 65 20 0
 90 95 20 0
 0 0 0 0
b>
 10 15 20 45
 30 35 20 85
 60 65 20 145
 90 95 20 205
190 210 80 0
Press any key to continue

```

```

#include <stdio.h>
int main(void)
{
    int gradeTable[5][4] = {{10, 15, 20, 0}, {30, 35, 20, 0},
                             {60, 65, 20, 0}, {90, 95, 20, 0}, {0, 0, 0, 0}};

    int i, j;

    printf("a)\n");
    for (i = 0; i < 5; i++)
    {
        for (j = 0; j < 4; j++)
            printf("%4d", gradeTable[i][j]);
        printf("\n");
    }

    // b)추가
}

```

LABHW16 2차원 배열 - 추가

LAB16_5(2차원 배열)

- 세로 3, 가로 5인 테이블에 난수(0에서 99까지)를 채우고 전체의 평균을 계산해서 출력한 후 테이블을 출력하는 프로그램을 작성하라.

실행예(실행시마다 다른 값이 나옴)

```
C:\Windows\system32\cmd.exe
발생된 난수의 평균은 50입니다
96 30 64 33 8
31 16 58 39 65
59 65 99 2 93
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\Windows\system32\cmd.exe
발생된 난수의 평균은 52입니다
13 56 82 76 16
53 92 80 32 10
35 50 67 46 78
계속하려면 아무 키나 누르십시오 . . .
```

- 위의 프로그램을 아래와 같이 함수로 나누어 다시 작성하라. 실행에는 위와 같다.
 - initTable: 난수를 배열에 넣는다.
 - totalTable: 총 합 계산하여 반환
 - printTable: 배열의 내용 출력

main 함수는 아래와 같다.

```
int main(void)
{
    int table[3][5];
    int i, j;
    int total = 0;

    srand(time(NULL));

    initTable(table, 3);
    printf("발생된 난수의 평균은 %d입니다\n", totalTable(table, 3) / 15);

    printTable(table, 3); // 호출

    return 0;
}
```

- (→ HW16_4)각 열의 합계를 구하여 출력하는 프로그램을 작성하라.

```
C:\Windows\system32\cmd.exe
발생된 난수의 평균은 48입니다
48 62 65 4 92
70 73 39 14 38
26 22 96 24 52
-----각 열의 합-----
144 157 200 42 182
계속하려면 아무 키나 누르십시오 . . .
```

Challenge16.1

1차원 배열로 구현한 [자리예약 시스템]을 조금 더 확대한 문제입니다. 열과 행을 갖는 좌석을 갖는 자리예약 시스템을 구현하는 프로그램을 작성하라. 기본 기능은 앞의 문제와 같다. 아래의 실행예를 갖는 프로그램을 작성하라.

```
C:\Windows\system32\cmd.exe
좌석을 예약하시겠습니까?(y/n) y

      1  2  3  4  5  6  7  8  9 10
-----
1      0  0  0  0  0  0  0  0  0  0
2      0  0  0  0  0  0  0  0  0  0
3      0  0  0  0  0  0  0  0  0  0

몇번째 좌석을 예약하시겠습니까? <열 행>의 형태로 입력: 3 5
예약되었습니다.

      1  2  3  4  5  6  7  8  9 10
-----
1      0  0  0  0  0  0  0  0  0  0
2      0  0  0  0  0  0  0  0  0  0
3      0  0  0  0  1  0  0  0  0  0

좌석을 예약하시겠습니까?(y/n) y

      1  2  3  4  5  6  7  8  9 10
-----
1      0  0  0  0  0  0  0  0  0  0
2      0  0  0  0  0  0  0  0  0  0
3      0  0  0  0  1  0  0  0  0  0

몇번째 좌석을 예약하시겠습니까? <열 행>의 형태로 입력: 3 5
이미 예약된 자리입니다.
좌석을 예약하시겠습니까?(y/n) y

      1  2  3  4  5  6  7  8  9 10
-----
1      0  0  0  0  0  0  0  0  0  0
2      0  0  0  0  0  0  0  0  0  0
3      0  0  0  0  1  0  0  0  0  0

몇번째 좌석을 예약하시겠습니까? <열 행>의 형태로 입력: 1 3
예약되었습니다.

      1  2  3  4  5  6  7  8  9 10
-----
1      0  0  1  0  0  0  0  0  0  0
2      0  0  0  0  0  0  0  0  0  0
3      0  0  0  0  1  0  0  0  0  0

좌석을 예약하시겠습니까?(y/n) n
계속하려면 아무 키나 누르십시오 . . .
```