

자료구조(F012) - 3장 실습 과제

G5_쓴맛볼쫄도독들(송유민, 김성준, 김현민, 김진형, 김다원)

01번 - 정답 : 3번,	02번 - 정답 : 2번,	03번 - 정답 : 2번,	04번 - 정답 : 3번,
05번 - 정답 : 2번,	06번 - 정답 : 1번,	07번 - 정답 : 4번,	08번 - 정답 : 2번,
09번 - 정답 : 3번,	10번 - 정답 : 4번,	11번 - 정답 : 2번,	12번 - 정답 : 4번,
13번 - 정답 : 4번,	14번 - 정답 : 4번,	15번 - 정답 : 4번,	16번 - 정답 : 3번,

17번

<정답>

3번 자리에 새 원소를 삽입하기 위해선 기존 3번 자리에 있던 원소를 4번으로, 4번 자리에 있던 원소를 5번으로, ~~~ 8번 자리에 있던 원소를 9번으로

총 6번 옮겨야 한다.

18번

<정답>

A[0] 3칸, A[1] 3칸, A[2] 3칸, A[3][0] 다음에 A[3][1] 이므로 첫 번째 원소 A[0][0]으로부터 총 $3*3 + 1 = 10$ 만큼 떨어져있다.

따라서 A[3][1]은 11번 째 원소이다.

19번

<정답>

시작 원소의 주소가 100 이므로 아홉 번째 원소의 주소는 $100 + 8 * 8 = 164$ 이다.

20번

20-1(정답) :

	[0]	[1]	[2]	[3]
[0]	1	2	3	4
[1]	5	6	7	8
[2]	9	10	0	0

20-2(정답) : 행 우선 순서 방법은 0번째 행의 요소들을 모두 세고 나서 다음 행으로 넘어가는 방법이다. 따라서 num[0][0]의 주소라고 하면, num[1][3]은 num[0][0]에서 7칸 떨어져 있으니까 num[1][3]의 주소는 1028이다.

20-3(정답) : 열 우선 순서 방법은 하나의 열을 모두 세고 나서 다음 열로 넘어가는 방법이다. num[0][0]의 주소가 1000번이면 num[1][3]의 순서는 1,5,9,2,6,10,3,7,0,4,8으로 11번째 순서이다. 따라서 num[0][0]으로부터 10칸 떨어져 있으므로 num[1][3]의 주소는 1040이다.

21번

<소스코드>

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAX_DEGREE 50

typedef struct {
    int degree;
    float coef[MAX_DEGREE];
} polynomial;

void multPoly(); // 다항식을 입력받고 곱한 후 결과를 출력하는 함수

void main() {
    multPoly();
}

void multPoly() {
    polynomial a, b, c = { 0 };
    int i, j;

    // 첫 번째 다항식 입력
    printf("첫 번째 다항식의 최고 차수를 입력하세요: ");
    scanf("%d", &a.degree);
    for (i = 0; i <= a.degree; i++) {
        printf("x^%d 항의 계수를 입력하세요: ", a.degree - i);
        scanf("%f", &a.coef[i]);
    }

    // 두 번째 다항식 입력
    printf("두 번째 다항식의 최고 차수를 입력하세요: ");
    scanf("%d", &b.degree);
    for (i = 0; i <= b.degree; i++) {
        printf("x^%d 항의 계수를 입력하세요: ", b.degree - i);
        scanf("%f", &b.coef[i]);
    }
}
```

```

// 두 다항식 곱하기
c.degree = a.degree + b.degree; // 결과 다항식의 최고 차수는 두 다항식 차수의 합
for (i = 0; i <= a.degree; i++) {
    for (j = 0; j <= b.degree; j++) {
        c.coef[i + j] += a.coef[i] * b.coef[j];
    }
}

// 첫 번째 다항식 출력
printf("첫 번째 다항식: ");
for (i = 0; i <= a.degree; i++) {
    printf("%2.0fx^%d", a.coef[i], a.degree - i);
    if (i < a.degree)
        printf(" + ");
}
printf("\n");

// 두 번째 다항식 출력
printf("두 번째 다항식: ");
for (i = 0; i <= b.degree; i++) {
    printf("%2.0fx^%d", b.coef[i], b.degree - i);
    if (i < b.degree)
        printf(" + ");
}
printf("\n");

// 곱한 결과 출력
printf("----- 두 식을 곱하면\n");
for (i = 0; i <= c.degree; i++) {
    printf("%2.0fx^%d", c.coef[i], c.degree - i);
    if (i < c.degree)
        printf(" + ");
}
printf("\n");

```

<실행 결과>

```

첫 번째 다항식의 최고 차수를 입력하세요: 4
x^4 항의 계수를 입력하세요: 3
x^3 항의 계수를 입력하세요: 2
x^2 항의 계수를 입력하세요: 1
x^1 항의 계수를 입력하세요: 0
x^0 항의 계수를 입력하세요: 1
두 번째 다항식의 최고 차수를 입력하세요: 4
x^4 항의 계수를 입력하세요: 1
x^3 항의 계수를 입력하세요: 0
x^2 항의 계수를 입력하세요: 2
x^1 항의 계수를 입력하세요: 0
x^0 항의 계수를 입력하세요: 1
첫 번째 다항식: 3x^4 + 2x^3 + 1x^2 + 0x^1 + 1x^0
두 번째 다항식: 1x^4 + 0x^3 + 2x^2 + 0x^1 + 1x^0
----- 두 식을 곱하면
3x^8 + 2x^7 + 7x^6 + 4x^5 + 6x^4 + 2x^3 + 3x^2 + 0x^1 + 1x^0

```

22번

<정답>

7	4	4
0	3	9
1	1	1
3	2	7
5	0	3

23번

<정답>

1	5	4
3	7	9
5	9	5
7	4	9

24번

<소스코드>

```
#include<stdio.h>

int main() {
    int arr[7][4] = { {0, 0, 0, 9}, {0, 1, 0, 0}, {0, 0, 0, 0}, {0, 0, 7, 0}, {0, 0, 0, 0}, {3, 0, 0, 0}, {0, 0, 0, 0} };
    int count = 0;
    for (int i = 0; i < 7; i++) {
        for (int j = 0; j < 4; j++) {
            if (arr[i][j]) {
                count++;
            }
        }
    }

    printf("원래의 행렬\n");
    for (int i = 0; i < 7; i++) {
        printf("%d\t%d\t%d\t%d\n", arr[i][0], arr[i][1], arr[i][2], arr[i][3]);
    }

    printf("변환 후\n");

    int save[4][3];
    count = 0;
    for (int i = 0; i < 7; i++) {
        for (int j = 0; j < 4; j++) {
            if (arr[i][j]) {
                save[count][0] = i;
                save[count][1] = j;
                save[count][2] = arr[i][j];
                count++;
            }
        }
    }

    printf("1행렬을2행2열\n");
    for (int i = 0; i < 4; i++) {
        printf("%d\t%d\t%d\n", save[i][0], save[i][1], save[i][2]);
    }

    return 0;
}
```

<실행 결과>

```
원래의 행렬
|      0      0      0      9      |
|      0      1      0      0      |
|      0      0      0      0      |
|      0      0      7      0      |
|      0      0      0      0      |
|      3      0      0      0      |
|      0      0      0      0      |
|
변환 후
|      행      열      값      |
|      0      3      9      |
|      1      1      1      |
|      3      2      7      |
|      5      0      3      |
```