

```

    for (int t = 0; t < 3; t++) {

```

프로그램 구성 4 단계인 다항식을 바꾸어 가며 과정을 3 번 반복하는 것을 위한 코드입니다.

```

int A[10] = { 0, };
int B[10] = { 0, };
int a, b;

```

다항식을 계수 배열 표현 방법으로 저장하기 위한 배열 선언입니다. 배열은 모든 항을 0 으로 초기화하며 선언했습니다. 변수 a, b 는 계수 배열로 scanf 하는 데에 이용하기 위해 미리 선언했습니다.

```

    typedef struct {
        int coef;
        int exp;
    } Term;
    Term D[8] = { 0 };

```

D(x) 배열을 항 배열 방식으로 저장하기 위해 미리 선언해둔 구조체와 배열입니다. 변수 coef 는 계수, exp 는 지수를 나타냅니다.

```

printf("A(x)를 입력해주세요.");
for (int i = 0; i < 3; i++) { //A(x)를 계수 배열로 저장
    scanf_s("%dx%d", &a, &b);
    A[b] = a;
}

```

```

printf("B(x)를 입력해주세요.");
for (int i = 0; i < 3; i++) { //B(x)를 계수 배열로 저장
    scanf_s("%dx%d", &a, &b);
    B[b] = a;
}

```

$A(x)$ ,  $B(x)$ 를 계수 배열로 저장하기 위한 반복문입니다. 모든 항이 0으로 초기화된 배열에 세 항의 값만 업데이트해서 저장해줍니다.

```
int j = 0; //D(x)의 항 표현 위한 변수
for (int k = 18; k > 9; k--) { //D(x) 안의 항의 지수가 9 초과일 경우
    for (int i = 9; i >= 0; i--) {
        if (k - i < 10 && A[i] != 0 && B[k - i] != 0) {
            D[j].coef += A[i] * B[k - i];
            D[j].exp = k;
        }
    }
    if (D[j].exp == k) {
        j++;
    }
}

for (int k = 9; k >= 0; k--) { //D(x) 안의 항의 지수가 9 이하일 경우
    for (int i = k; i >= 0; i--) {
        if (A[i] != 0 && B[k - i] != 0) {
            D[j].coef += A[i] * B[k - i];
            D[j].exp = k;
        }
    }
    if (D[j].exp == k) {
        j++;
    }
}
```

$A(x)$ 와  $B(x)$ 를 곱해서  $D(x)$ 를 얻는 과정의 코드입니다. 지수가 높은 항부터 지수 내림순으로 정렬하기 위해 가능한 가장 큰 지수인 18부터 1씩 내려가며 해당 지수 항이 있는지 찾는 알고리즘입니다. 18일 경우 ( $A[9]$ ,  $B[9]$ ), 17일 경우 ( $A[9]$ ,  $B[8]$ ), ( $A[8]$ ,  $B[9]$ ) 이런식으로 짝지어지는 항 두개가 모두 0이 아니라면 두 항의 계수를 곱한 다음  $D(x)$  배열에 저장합니다. 이때, 지수의 합  $k$ 가 같은 항이 두 쌍 이상이라면, 지수의 합이 같은 항의 계수를 더해서 저장됩니다. 그리고 지수를 표현하는 `exp`에는  $k$ 를 저장해줍니다. 지수의 합  $k$ 가 9보다 작아질 경우,  $i$ 는  $k$ 부터 1씩 작아져야 하므로  $k$ 가 9 초과일때와 9 이하일때로 나누어 코딩했습니다.

```

printf("\nA(x)=");
for (int i = 0; i < 10; i++) {
    printf("%dx%d", A[i], i);
    if (A[i + 1] >= 0) {
        printf("+");
    }
}
printf("\nB(x)=");
for (int i = 0; i < 10; i++) {
    printf("%dx%d", B[i], i);
    if (B[i + 1] >= 0) {
        printf("+");
    }
}

printf("\nD(x)=");
for (int i = 0; D[i].coef != 0; i++) { //D(x) 출력
    printf("%dx%d", D[i].coef, D[i].exp);
    if (D[i + 1].coef > 0) {
        printf("+");
    }
}
}

```

반복문을 활용해 A(x)와 B(x), D(x)에 저장된 값을 다시 출력하는 코드입니다. A(x)와 B(x)는 지수 오름순으로 계수가 0 인 항도 모두 표시해서 출력합니다. 만약 다음 항이 마이너스가 아니고, 이번 항이 마지막 항이 아니라면 문자 +를 출력해 다항식으로써 이해하기 편하도록 했습니다. D(x)의 경우 배열의 크기는 9 이지만 계수가 0 인 항은 저장하지 않는 항 배열 특성상 coef 가 0 인 항이 시작되면 반복문이 진행되지 않도록 작성했습니다. D(x)는 지수 내림순으로 계수가 0 이 아닌 항만 출력됩니다.

```

A(x) == 입력해주세요 .2x5+7x2-4x0
B(x) == 입력해주세요 .6x3+2x1+9x0

A(x)=-4x0+0x1+7x2+0x3+0x4+2x5+0x6+0x7+0x8+0x9
B(x)=9x0+2x1+0x2+6x3+0x4+0x5+0x6+0x7+0x8+0x9
D(x)=12x8+4x6+60x5-10x3+63x2-8x1-36x0
A(x) == 입력해주세요 .4x8-2x4+4x0
B(x) == 입력해주세요 .5x9-1x6+9x2

A(x)=4x0+0x1+0x2+0x3-2x4+0x5+0x6+0x7+4x8+0x9
B(x)=0x0+0x1+9x2+0x3+0x4+0x5-1x6+0x7+0x8+5x9
D(x)=20x17-4x14-10x13+38x10+20x9-22x6+36x2
A(x) == 입력해주세요 .4x3-2x2-1x0
B(x) == 입력해주세요 .-7x2+8x1+4x0

A(x)=-1x0+0x1-2x2+4x3+0x4+0x5+0x6+0x7+0x8+0x9
B(x)=4x0+8x1-7x2+0x3+0x4+0x5+0x6+0x7+0x8+0x9
D(x)=-28x5+46x4
D:문버리#2학년 2학기#자료구조#자료구조 1차과제#Debug#자료구조 1차과제.exe(프로세스 21188개)이(가) 종료되었습니다(코드:
02#)
이 창을 닫으려면 아무 키나 누르세요...

```

A(x)와 B(x)의 항을 입력받은 뒤, 계수가 0 인 항까지 포함한 A(x)와 B(x)를 출력하고, 두 다항식의 곱인 D(x)도 출력합니다. A(x)의 최고차항이 더 큰 경우, B(x)의 최고차항이 더 큰 경우 두가지 모두 잘 출력되는 것을 확인했습니다.