

자료구조 실습과제 04

- 개발 도구 : visual studio를 이용한 C프로그래밍
- 솔루션 명칭 : Sol_04_이름이니셜(예: Sol_04_hsoh)
- 프로젝트 명칭 : 문제 번호(예: Prob_01, Prob_02, ...)
- 제출방법 : 아래 문제를 하나의 솔루션에 여러 개의 프로젝트로 구성하여 프로그램한 후, 컴파일 및 실행해보고 오류가 없으면, 메뉴에서 솔루션 정리를 수행한 후 윈도우 탐색기에서 솔루션 폴더를 찾아 압축하여 E-class에 올림

문제 1) 연결리스트로 표현되는 두 다항식 연산 덧셈과 곱셈을 수행하는 프로그램을 아래 작성 방법을 고려하여 작성하시오.

- 다항식은 고차원부터 내림차순으로 입력된다.
- 한 다항식의 각 항을 저장할 수 있는 구조체를 정의한다. 각 항은 정수형 계수 coef와 정수형 차수 exp를 가진다.

구현 및 실행 예 :

함수의 프로토타입은 다음과 같이 정의하고 해당 함수를 구현하시오.

```
typedef struct PolyNode {
    int coef;
    int exp;
    struct PolyNode* link;
}node;

node* addPoly(node* a, node* b);
node* mulPoly(node* a, node* b);
```

위 함수를 이용하여 main()함수에서

```
// a(x) = 3x5 + 4x3 - 2x2 + 1x + 7
```

```
// b(x) = 4x3 + 2x2 - 6x - 3
```

의 예와 같은 두 다항식이 연결리스트로 구성되었다고 가정할 경우 실행 결과는 다음과 같다.

Microsoft Visual Studio 디버그 콘솔

```
a(x) = 3x^5    4x^3    -2x^2    1x^1    7x^0
b(x) = 4x^3    2x^2    -6x^1    -3x^0
c(x) = 3x^5    8x^3    -5x^1    4x^0
d(x) = 12x^8    6x^7    -2x^6    -9x^5    -24x^4    30x^3    14x^2    -45x^1    -21x^0
```

여기서 c(x)는 덧셈 결과이고, d(x)는 곱셈 결과이다.

문제 2) 명함을 관리하는 구조체 배열을 이용하여 작성하시오. 명함은 이름, 회사, 전화번호의 3가지 정보를 가지며, 프로그램을 통하여 다음과 같은 작업을 수행하고자 한다.

프로그램의 기능은

1. 새로운 명함을 입력받아 저장함
2. 명함자료에 저장된 모든 명함을 출력
3. 명함 이름으로 검색하여 해당 정보를 모두 출력함. 단, 찾고자하는 단어가 이름의 일부인 경우도 검색하여 출력
4. 명함을 이름으로 검색하여 삭제함. 단, 명함 이름과 일치하는 경우만 삭제
5. 프로그램을 종료

프로그램이 시작되면 파일에서 namecard.txt에 저장되어 있는 모든 명함자료를 읽어 구조체 배열에 저장하고, 프로그램이 종료될 때 구조체에 저장되어 있는 모든 명함정보를 namecard.txt에 저장한다. 처음 시작할 때는 해당 파일이 없으며, 프로그램 수행 과정에 입력된 명함을 저장할 때 파일을 생성한다.

main 함수의 구성과 실행 예는 다음과 같다.

```
int main(void)
{
    ...
    count = init(NC);

    while (1) {
        printf_menu();
        scanf("%d", &menu_item); gets(answer);
        ...
        switch (menu_item) {
            case 1:
                printf("명함 입력\n");
                printf("-----\n");
                ...
                break;
            case 2 :
                printf("명함 출력\n");
                printf("-----\n");
                ...
                break;
            case 3 :
                printf("명함 검색\n");
                printf("-----\n");
                ...
                break;
            case 4 :
                printf("명함 삭제\n");
                printf("-----\n");
                ...
                break;
            case 5 :
                printf("프로그램 종료\n");
                finalize(NC, count);
                ...
        }
    }
    ...
}
```

}

```
===== Name Card DBMS =====
명령어를 선택하세요.
1. 명함 입력
2. 모든명함 출력
3. 명함 검색
4. 명함 삭제
5. 종료
명령 입력 --> 2

명함 출력
-----
모든 명함 출력
0. 이름:3, 회사:3, 전화:3
1. 이름:11, 회사:11, 전화:11
2. 이름:1, 회사:1, 전화:1
3. 이름:11, 회사:11, 전화:11
4. 이름:222, 회사:222, 전화:222
5개의 명함이 존재함

===== Name Card DBMS =====
명령어를 선택하세요.
1. 명함 입력
2. 모든명함 출력
3. 명함 검색
4. 명함 삭제
5. 종료
명령 입력 --> 1

명함 입력
-----
이름 : 2222
회사 : 2222
전화 : 2222
저장할까요<yes or no>? : yes
저장하였습니다.

===== Name Card DBMS =====
명령어를 선택하세요.
1. 명함 입력
2. 모든명함 출력
3. 명함 검색
4. 명함 삭제
5. 종료
명령 입력 --> 2

명함 출력
-----
모든 명함 출력
0. 이름:3, 회사:3, 전화:3
1. 이름:11, 회사:11, 전화:11
2. 이름:1, 회사:1, 전화:1
3. 이름:11, 회사:11, 전화:11
4. 이름:222, 회사:222, 전화:222
5. 이름:2222, 회사:2222, 전화:2222
6개의 명함이 존재함
```

```
===== Name Card DBMS =====
명령어를 선택하세요.
1. 명함 입력
2. 모든명함 출력
3. 명함 검색
4. 명함 삭제
5. 종료
명령 입력 --> 3

명함 검색
-----
검색할 이름을 입력하세요. : 22
검색한 명함 이름:222, 회사:222, 전화:222
검색한 명함 이름:2222, 회사:2222, 전화:2222
2개의 명함 검색함

===== Name Card DBMS =====
명령어를 선택하세요.
1. 명함 입력
2. 모든명함 출력
3. 명함 검색
4. 명함 삭제
5. 종료
명령 입력 --> 4

명함 삭제
-----
삭제할 이름을 입력하세요.: 222
삭제 명함 이름:222, 회사:222, 전화:222
1개의 명함 삭제함

===== Name Card DBMS =====
명령어를 선택하세요.
1. 명함 입력
2. 모든명함 출력
3. 명함 검색
4. 명함 삭제
5. 종료
명령 입력 --> 2

명함 출력
-----
모든 명함 출력
0. 이름:3, 회사:3, 전화:3
1. 이름:11, 회사:11, 전화:11
2. 이름:1, 회사:1, 전화:1
3. 이름:11, 회사:11, 전화:11
4. 이름:2222, 회사:2222, 전화:2222
5개의 명함이 존재함

===== Name Card DBMS =====
명령어를 선택하세요.
1. 명함 입력
2. 모든명함 출력
3. 명함 검색
4. 명함 삭제
5. 종료
명령 입력 --> 5

프로그램 종료
```