

프로젝트 최종 보고서

-전화번호부 프로그램

팀원 명: 장은희, 유대균, 감달현

전화번호부 프로그램

1. 프로젝트 주제 및 계획	3
1.1 개발 시간표	4
1.2 플로우차트를 이용한 코드 설계	4
2 프로젝트 기획 및 기능	5
2.1 세가지 다른 권한 부여	5
2.2 사용자 등록	5
2.3 전화번호부 입력	5
2.4 전화번호부 수정	5
2.5 전화번호부 삭제	5
2.6 전화번호부 전체보기	6
2.7 전화번호부 찾기	6
3. 사용자 인터페이스 및 매뉴얼	7
3.1 초기화면	7
3.2 사용자 등록	8
3.3 로그인	9
3.4 전화번호부 메뉴 화면	9
3.5 전화번호부 입력 화면	10
3.6 전화번호부 수정 화면	11
3.7 전화번호부 삭제 화면	12
3.8 전화번호부 출력 화면	13
3.9 전화번호부 찾기 화면	14
4. 코드 설명	15
5. 어려웠던 점 및 느낀 점	20

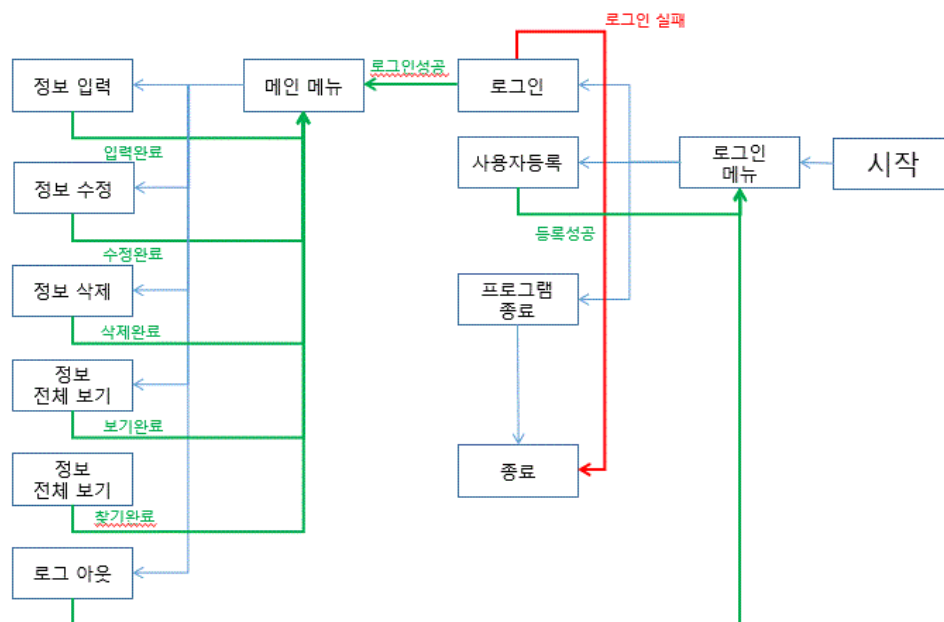
1. 프로젝트 주제 및 계획

개발 기간	2021.01.05 ~ 2021.01.13 (총 9일)
개발 운영체제	Window
개발 도구	Dev c++
개발 언어	C언어
개발 인원	3명
프로젝트 이름	전화번호부 프로그램
프로젝트 설명	관리자, 보조관리자와 일반사용자로 나누어져 있으며, 각 권한마다 제한된 기능과 사용할 수 있는 기능이 있다. 전화번호를 입력하고 입력한 전화번호를 수정, 삭제 및 출력을 할 수 있는 프로그램을 구현했다.
주제 선정 의도	배운 내용을 적극 활용하며, 실생활에서 쓰일 수 있는 프로그램을 만들기 위해 전화번호부 등록 프로그램을 구현하였다.
기능	<ul style="list-style-type: none"> 1) 세가지 다른 권한 부여 2) 사용자 등록 3) 전화번호부 입력 4) 전화번호부 수정 5) 전화번호부 삭제 6) 전화번호부 전체보기 7) 전화번호부 찾기

1.1 개발 시간표

	1.05	1.06	1.07	1.08	1.09	1.10	1.11	1.12	1.13
설계									
구현할 기능 기획									
플로우차트									
구현									
전체 코드 틀 구상									
기능 구현									
콘솔 화면 디자인									
버그 수정 및 점검									

1.2 플로우차트를 이용한 코드 설계



2 프로젝트 기획 및 기능

2.1 세가지 다른 권한 부여

사용자 권한, 보조관리자 권한, 관리자 권한 세가지 권한으로 나눌 수 있다.

- 사용자 권한: 입력, 수정, 삭제, 전체보기 기능을 지원하지만 전화번호부 찾기 기능은 지원하지 않는다. 입력, 수정, 삭제 기능은 사용자가 입력한 정보만 접근 가능하며, 전체보기는 아무런 제약이 없다.
- 보조관리자 권한: 사용자 권한과 동일하게 찾기 기능만 지원하지 않는다. 사용자 권한과 다른 점은, 보조관리자는 자신이 입력한 정보와 사용자 권한이 입력한 정보까지 접근 가능하다.
- 관리자 권한: 모든 기능 접근이 가능하다.

2.2 사용자 등록

권한을 선택 한 후 사용할 아이디 및 비밀번호를 등록 한다. 구조체 sta_Login배열에 정보가 입력 된다.

2.3 전화번호부 입력

전화번호 입력에는 이름, 나이, 전화번호, 주소를 입력 할 수 있으며 주소 입력은 선택. 주소를 입력하지 않을 시 '-'으로 대체함. 정보를 다 입력 후 "계속 입력"을 선택하면 데이터를 계속해서 입력할 수 있다. "돌아가기" 선택 시, 메뉴화면으로 돌아간다 데이터는 총 10개만 저장 가능하다.

2.4 전화번호부 수정

권한에 따라 접근이 가능한 목록이 출력되며, 목록 안에서 원하는 데이터를 선택하여 수정할 수 있다.

2.5 전화번호부 삭제

수정기능과 동일하게 권한에 따라 접근이 가능한 목록만이 출력되며, 목록 안에서 원하는 데이터를 삭제할 수 있다.

2.6 전화번호부 전체보기

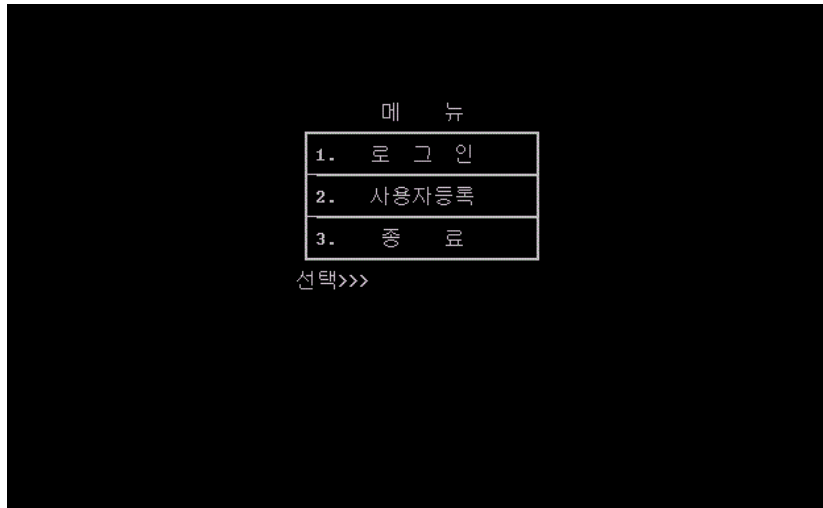
전화번호부에 입력된 모든 정보를 출력하여 볼 수 있다.

2.7 전화번호부 찾기

관리자 권한만 접근 할 수 있는 기능이며, 원하는 정보를 이름, 전화번호, 나이, 주소로 찾을 수 있다.

3. 사용자 인터페이스 및 매뉴얼

3.1 초기화면



사용자가 프로그램을 접속하면 나오는 초기 화면이다.

신규 사용자인 경우 2번 "사용자 등록"을 선택하여, 로그인 시도할 아이디, 비밀번호를 생성해야 된다. 기존 아이디가 있는 사용자일 경우, 1번 "로그인"을 선택하면 된다.

3.2 사용자 등록

The image displays three sequential screenshots of a user registration interface, each enclosed in a black rectangular frame.

The first screenshot shows the title "사용자 등록" (User Registration) at the top. Below it is a single input field labeled "이름 입력:" (Name Input:).

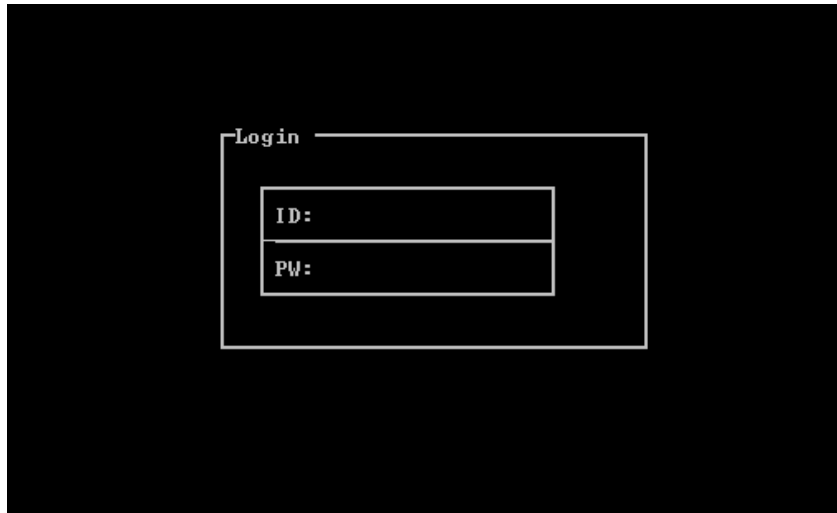
The second screenshot shows the title "사용자 등록" at the top. Below it is a section titled "권한선택" (Authority Selection) containing a list of three options: "1. 사용자 권한" (User Authority), "2. 보조관리자 권한" (Assistant Manager Authority), and "3. 관리자 권한" (Administrator Authority). Below this list, the text "권한선택>>>>" (Authority Selection>>>>) is displayed.

The third screenshot shows the title "사용자 등록" at the top. Below it are two input fields: "ID :" and "PW :".

사용자는 이름 입력과 이용할 권한을 선택한 후, 사용할 아이디와 비밀번호를 생성할 수 있다.

관리자 권한이 최고 높은 권한이며, 그 다음은 보조관리자 권한, 마지막으로 사용자 권한이 있다.

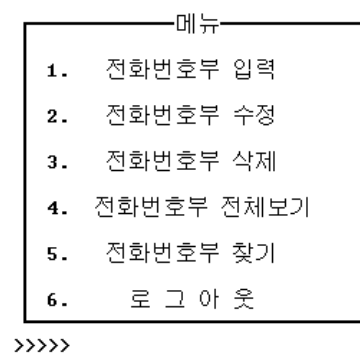
3.3 로그인

A screenshot of a login screen. It features a black background with a white rectangular box in the center. Inside this box, the word "Login" is written in a small font at the top left. Below it, there are two input fields: the first is labeled "ID:" and the second is labeled "PW:". Both labels are in a small font, and the input fields are empty.

아이디, 비밀번호를 생성한 유저면, 로그인을 통해 전화번호 프로그램을 사용할 수 있다.

만약 3번이상 로그인 실패 시, 프로그램은 자동 종료된다.

3.4 전화번호부 메뉴 화면

A screenshot of a phonebook menu screen. It has a black background. At the top center, the word "메뉴" (Menu) is written in a small font. Below it, there is a white rectangular box containing a list of six menu items, each preceded by a number: 1. 전화번호부 입력, 2. 전화번호부 수정, 3. 전화번호부 삭제, 4. 전화번호부 전체보기, 5. 전화번호부 찾기, 6. 로그아웃. Below the white box, the text ">>>>>" is displayed in a small font.

메뉴화면에서는 총 6가지의 메뉴를 선택할 수 있으며, 6번 "로그아웃"을 선택하여 로그인 화면으로 되돌아갈 수 있다.

3.5 전화번호부 입력 화면

전화번호부 입력

이름:

나이:

전화번호:010-

주소:

1. 계속 입력하기

2. 돌 아 가 기

>>>>

입력 화면에서는 이름, 나이, 전화번호, 주소를 입력할 수 있으며, 주소는 선택사항이다.

전화번호는 8개의 숫자를 입력해야 하며, 형식과 다를 경우 “전화번호를 다시 입력해주세요”라는 문고가 뜨며 전화번호 칸에 커서가 다시 포커스 된다.

전화번호부 입력을 마치면, 1번 “계속 입력하기”를 선택할 경우 다시 전화번호부 입력으로 돌아가고, 2번 “돌아가기”를 선택하면 메뉴화면으로 돌아간다.

3.6 전화번호부 수정 화면

-----수정 가능한 전화번호부 목록-----

1. 이름 : 김달현 , 전화번호: 010-1234-5678 , 나이 : 30 , 주소 : 경기도 여주시

2. 이름 : 김달현 , 전화번호: 010-1234-5678 , 나이 : 30 , 주소 : 경기도 광주시

3. 이름 : 장은희 , 전화번호: 010-1111-2222 , 나이 : 30 , 주소 : 경기도 여주시

4. 이름 : 유대균 , 전화번호: 010-1111-2222 , 나이 : 20 , 주소 : 경기도 광주시

5. 이름 : 은희장 , 전화번호: 010-9586-6763 , 나이 : 25 , 주소 : -

-----메인으로 돌아가려면 0을 입력하세요.-----

번호입력>>>>

수정 항목 선택

1. 이름 수정

2. 전화번호 수정

3. 나이 수정

4. 주소 수정

5. 수정 취소

>>>>

수정하길 원하는 정보의 번호를 입력 후, 수정할 항목을 선택한다.

사용자 권한은 동급 권한자가 입력한 내용만 수정 가능하다.

보조관리자 권한은 사용자권한과 동급 권한자가 입력한 내용만 수정 가능하다.

관리자 권한은 모든 사용자들의 정보 수정이 가능하다.

수정 전 정보

이름: 장은희

나이: 30

전화번호:010-1111-2222

주소: 경기도 여주시

→

이름 수정

이름: 장은희

나이: 25

전화번호:010-1111-2222

주소: 경기도 여주시

수정 항목 선택 후, 수정할 내용을 입력하면 수정된 내용이 저장된다.

3.7 전화번호부 삭제 화면

삭제 가능한 전화번호부 목록	
1. 이름 : 감달현, 전화번호: 010-1234-5678, 나이 : 30, 주소 : 경기도 여주시	
2. 이름 : 감달현, 전화번호: 010-1234-5678, 나이 : 30, 주소 : 경기도 광주시	
3. 이름 : 장은희, 전화번호: 010-1111-2222, 나이 : 30, 주소 : 경기도 여주시	
4. 이름 : 유대균, 전화번호: 010-1111-2222, 나이 : 20, 주소 : 경기도 광주시	
5. 이름 : 은희장, 전화번호: 010-9586-6763, 나이 : 25, 주소 : -	
번호입력<0번 종료>>>>	

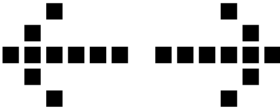
전화번호부 수정 화면과 마찬가지로, 원하는 정보의 번호를 입력하면, 정보가 삭제되며, 숫자는 1
부터 다시 정렬된다.

사용자 권한은 동급 권한자가 입력한 내용만 삭제 가능하다.

보조관리자 권한은 사용자권한과 동급 권한자가 입력한 내용만 삭제 가능하다.

관리자 권한은 모든 사용자들의 정보를 삭제할 수 있다.

3.8 전화번호부 출력 화면

이름: 감달현 나이: 30 전화번호: 010-1234-5678 주소: 경기도 여주시	
이름: 감달현 나이: 30 전화번호: 010-1234-5678 주소: 경기도 광주시	
이름: 장은희 나이: 30 전화번호: 010-1111-2222 주소: 경기도 여주시	
이름: 유대균 나이: 20 전화번호: 010-1111-2222 주소: 경기도 광주시	
이름: 은희장 나이: 25 전화번호: 010-0555-0555 주소: -	
	돌아가기<0번>, 다음페이지<2번>

입력된 모든 정보를 출력하며, 5개 이상의 정보가 있으면, 다음페이지를 넘어가야 그 다음 정보들을 볼 수 있다.

3.9 전화번호부 찾기 화면

찾기 메뉴

1. 이름으로 검색

2. 전화번호로 검색

3. 나이로 검색

4. 주소로 검색<도,시>

5. 메인 메뉴

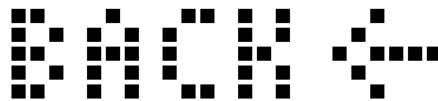
>>>>

검색 모드

이름 : 장은희

1. 이름 : 장은희 | 전화번호: 010-1111-2222 | 나이 : 30 | 주소 : 경기도 여주시 |

돌아가기<아무키>



전화번호부 찾기 화면은 관리자 권한인 사용자만 접근할 수 있다. 찾기 메뉴에서 원하는 선택지를 선택한 후, 출력하고 싶은 정보를 입력하면, 검색모드 하단에 정보가 출력된다.

4. 코드 설명

Code	해설
<pre> //로그인 함수 int login(void){ int i,j,i_idx,i_Count=0; char ca_TmpID[20],ca_TmpPw[20]; for(i=0;i<MAX;i++){ if(sta_Login[i].ca_Id != NULL){ i_Count++; } } for(j=0;j<FAIL;j++){ system("cls"); Alarm(12); // 로그인 ID,PW 입력 디자인 if(j==2){ gotoxy(22,4);printf("다음을 오류시 프로그램을 자동 종료합니다.ㄷ"); } gotoxy(34,9);scanf("%s",&ca_TmpID); gotoxy(34,11);scanf("%s",&ca_TmpPw); for(i=0;i<i_Count;i++){ if(strcmp(ca_TmpID,sta_Login[i].ca_Id) == 0){ if(strcmp(ca_TmpPw,sta_Login[i].ca_Pwd) == 0){ system("cls"); i_idx = i; return i_idx; }else{ gotoxy(31,16);printf("ㄷPw가 틀렸습니다.ㄷ"); Sleep(1000); break; // for(i=0;i<i_Count;i++) 은 종료 상위 for문은 통과함 } } if(i == i_Count-1){ gotoxy(25,16);printf("ㄷ일련하신 ㄷ는 없는 ID 입니다.ㄷ",ca_TmpID); Sleep(1000); } //for(i=0;i<i_Count;i++) } Sleep(1000); return -1; } } </pre>	<p>[login()-로그인 함수]</p> <p>로그인 3번 이상 실패시 프로그램 자동 종료. 3번째 시도시 프로그램 종료 경고문 표시.</p>
<pre> // 사용자 등록 함수 void Regit(void){ int i; Alarm(10); // 사용자 이름 등록 디자인 char tmp[SIZE]; bool b_Pass = false; gotoxy(35,11);scanf("%s",&sta_Login[all_LoginCnt].ca_Name); printf("권한 입력 : \n"); while(1){ Alarm(4); gotoxy(35,17);scanf("%d",&sta_Login[all_LoginCnt].i_Auth); if(sta_Login[all_LoginCnt].i_Auth > 0 && sta_Login[all_LoginCnt].i_Auth < 4){ break; }else{ gotoxy(18,17);printf("ㄷ권한 선택은 1~3 입니다.ㄷ"); Sleep(1000); rewind(stdin); } } system("cls"); gotoxy(24,11);printf("ㄷ사용자 ID와 Pw를 생성해 주세요.ㄷ"); Sleep(1000); while(1){ system("cls"); Alarm(11); // 사용자 등록 ID,Pw 입력 디자인 gotoxy(30,10); scanf("%s",&tmp); for(i=0;i<all_LoginCnt;i++){ if(strcmp(tmp, sta_Login[i].ca_Id) == 0){ gotoxy(19,18);printf("ㄷ동일한 ID가 있습니다. ID를 새로 입력하세요.ㄷ"); Sleep(1000); break; } if(i >= all_LoginCnt-1){ strcpy(sta_Login[all_LoginCnt].ca_Id,tmp); b_Pass = true; } //for(i=0;i<all_LoginCnt;i++) } //for(i=0;i<all_LoginCnt;i++) if(b_Pass){ // 동일한 ID가 없으면 while 루프 빠져나옴 break; } } //while(1) gotoxy(30,12);scanf("%s",&sta_Login[all_LoginCnt].ca_Pwd); do{ i = PwCheck(); }while(i != 0); // pw 재확인시 일치하면 종료되면 루프 all_LoginCnt++; // 사용자 등록 완료시 index 증가 } //void Regit(int idx) // 패스워드 체크 함수 int PwCheck(void){ char ca_Password[20]; // password 확인을 위한 입력 변수 gotoxy(21,14);printf("Pw 재확인 : "); gotoxy(35,14);scanf("%s",&ca_Password); return strcmp(ca_Password,sta_Login[all_LoginCnt].ca_Pwd); // pw 재확인시 일치한 pw가 동일하면 0 return } //int PwCheck(int idx) </pre>	<p>[Regit()-사용자 등록 함수]</p> <ol style="list-style-type: none"> 1. 사용자, 보조관리자, 관리자 권한을 선택하여 아이디와 비밀번호를 생성할 수 있음. 2. 비밀번호 재확인시 일치하면 등록한 정보를 구조체 Login배열에 넣음. 3. 중복된 아이디 입력 불가능. <p>[PwCheck()-비밀번호 재확인 메서드]</p> <p>비밀번호 재확인 메서드. PwCheck()함수에서 strcmp()함수로 비밀번호가 일치하면 return값으로 0을 아니면 0외의 값을 전달.</p>

```

//입력함수
void Input(void){
    int i; //문자는 끝때 쓰는 변수
    char ch_ans; //주소 묻기 안묻기 Y or N
    int i_menu1; //종류부 다스리고 계속 묻지 않아함

    for(i=1;){ //전화번호부 입력 실행 디자인 출력
        if(alli_Cnt<10){
            Alarm(14); //전화번호부 입력 디자인 출력
            gotoxy(36,9);scanf("%s",&sta_Info[alli_Cnt].ca_Name);
            for(i=1;){
                Alarm(14); //전화번호부 입력 디자인 출력
                gotoxy(36,9);printf("%s",sta_Info[alli_Cnt].ca_Name);
                gotoxy(36,12);
                while (!scanf("%d",&sta_Info[alli_Cnt].ui_Age)){ //숫자로 입력 불가능할때 숫자
                    setcolor(4,12);gotoxy(36,12);printf("숫자수만 입력 가능 다시 입력 하세요\n");
                    Sleep(800);
                    setcolor(6,15);
                    Alarm(14); //전화번호부 입력 디자인 출력
                    gotoxy(36,9);printf("%s",sta_Info[alli_Cnt].ca_Name);
                    gotoxy(36,12);while (getchar() != '\n');
                }

                if(sta_Info[alli_Cnt].ui_Age>=0 && sta_Info[alli_Cnt].ui_Age<=120){
                    break;
                }else{
                    setcolor(4,12);gotoxy(28,22);printf("나이를 다시입력 하세요.");
                    Sleep(800);
                    setcolor(6,15);
                }
            }
        }
        rewind(stdin);
        for(i=1;){
            Alarm(14); //전화번호부 입력 디자인 출력

            gotoxy(36,9);printf("%s",sta_Info[alli_Cnt].ca_Name);
            gotoxy(36,12);printf("%d",sta_Info[alli_Cnt].ui_Age);

            gotoxy(37,15);scanf("%s",&sta_Info[alli_Cnt].ca_Number);
            if(strlen(sta_Info[alli_Cnt].ca_Number)==8){ //8자리 일정한 양분.
                for(i=0;i<4;i++){
                    if(sta_Info[alli_Cnt].ca_Number[i]>=48 && sta_Info[alli_Cnt].ca_Number[i]<=57){
                        }
                    }
                }
            }else{
                setcolor(4,12);gotoxy(28,22);printf("전화번호를 다시 입력해주세요.");
                Sleep(800);
                setcolor(6,15);
            }
        }
    }
    for(i=1;){ //전화번호
        Alarm(14); //전화번호부 입력 디자인 출력
        gotoxy(36,9);printf("%s",sta_Info[alli_Cnt].ca_Name);
        gotoxy(36,12);printf("%d",sta_Info[alli_Cnt].ui_Age);
        gotoxy(37,15);printf("%s",sta_Info[alli_Cnt].ca_Number);

        gotoxy(28,22);printf("주소는 선택됩니다. 쓰시겠습니까?(Y/N)");scanf("%c",&ch_ans);
        if(ch_ans=='Y' || ch_ans=='y'){
            Alarm(14); //전화번호부 입력 디자인 출력
            gotoxy(36,9);printf("%s",sta_Info[alli_Cnt].ca_Name);
            gotoxy(36,12);printf("%d",sta_Info[alli_Cnt].ui_Age);
            gotoxy(37,15);printf("%s",sta_Info[alli_Cnt].ca_Number);
            gotoxy(28,22);printf("도와 지만 입력해 주세요.");
            gotoxy(36,18);scanf("%s",&sta_Info[alli_Cnt].ca_Addr);
            break;
        }else if(ch_ans=='N' || ch_ans=='n'){
            strcpy(sta_Info[alli_Cnt].ca_Addr,"-");
            gotoxy(28,23);printf("주소를 쓰지 않습니다.");
            Sleep(800);
            break;
        }else{
            Alarm(14); //전화번호부 입력 디자인 출력
            gotoxy(36,9);printf("%s",sta_Info[alli_Cnt].ca_Name);
            gotoxy(36,12);printf("%d",sta_Info[alli_Cnt].ui_Age);
            gotoxy(37,15);printf("%s",sta_Info[alli_Cnt].ca_Number);
            setcolor(4,12);gotoxy(28,22);printf("Y와N 중 하나만 선택하세요.");
            Sleep(700);setcolor(6,15);
        }
    }
    if(ch_ans=='Y' || ch_ans=='y')
        sta_Info[alli_Cnt].i_Auth = alli_Auth;
    alli_Cnt++;
}
if(alli_Cnt<10)
    Alarm(9);
scanf("%d",&i_menu1);
if(i_menu1==1){
    continue;
}else if(i_menu1==2){
    break;
}else if(alli_Cnt==10){
    printf("10명 까지만 입력 가능합니다.\n");
    printf("데이터를 삭제 후 입력해 주세요.");
    break;
}
if(i_menu1==1)
    for(i=1;){ //전화번호부 입력
}
}
void Input(void)

```

[Input()-전화번호부 입력 함수]

1. 전화번호부 입력시 형식에 맞춰 입력해야 됨.

- 전화번호 입력: Strlen()함수로 문자열 길이를 8자리로 고정함. 또한 아스키코드를 사용하여 정수만 입력 허용함.

- 나이 입력: 0~120살 까지의 정수만 입력가능

2. 주소는 선택사항 이며, 주소 입력 거부 시 '-'(hyphen)으로 대체.

3. 데이터는 10개까지만 저장이 가능하며 데이터 입력을 완료하면 구조체 sta_info배열에 정보가 저장됨.


```

// 출력 페이지 처리 함수
void Output_page2(){
    int i,s,i_back1,i_back2;

    for(;;){
        system("cls");
        for(i=0;i<5;i++){
            printf("\n이름: %s\n",sta_Info[i].ca_Name);
            printf("\n나이: %d\n",sta_Info[i].ui_Age);
            NumberChange(sta_Info[i].ca_Number,1); // 전화번호 출력 변경
            printf("\n주소: %s\n",sta_Info[i].ca_Addr);
            printf("-----");
        } //for(i=0;i<5;i++) 첫페이지

        Alarm(16); // 1페이지 돌아가기 디자인 출력
        gotoxy(73,19);scanf("%d",&i_back1);

        if(i_back1==0){
            Alarm(15); // 메뉴로 돌아가기 디자인 출력
            break;
        } //if(i_back1==0)

        if(i_back1==2){
            system("cls");
            for(i=5;i<all_i_Cnt;i++){
                printf("\n이름: %s\n",sta_Info[i].ca_Name);
                printf("\n나이: %d\n",sta_Info[i].ui_Age);
                NumberChange(sta_Info[i].ca_Number,1); // 전화번호 출력 변경
                printf("\n주소: %s\n",sta_Info[i].ca_Addr);
                printf("-----");
            } //for(i=5;i<all_i_Cnt;i++)

            Alarm(17); // 2페이지 돌아가기 디자인 출력

            gotoxy(71,19);scanf("%d",&i_back2);
            if(i_back2==0){
                Alarm(15); // 메뉴로 돌아가기 디자인 출력
                break;
            } //if(i_back2==0)
        } //if(i_back1==1) 다음페이지
    } //for(;;)
} //void Output_page2()

//출력함수
void Outut(void){
    int i_menu; //전화번호 출력 및 이동하기 메뉴
    int i; //for 문 돌릴때, 전체 출력할때 사용하는 변수
    int s; // for 문 돌릴때, 전화번호 4자리씩 출력할때 사용하는 변수
    int i_back; //전체출력후 돌아가기 하고 싶을때 쓰는 변수.

    if(all_i_Cnt==0){
        system("cls");
        printf("출력할 정보가 없습니다.");
        Sleep(700);
        Alarm(15); // 메뉴로 돌아가기 디자인 출력
    } //if(all_i_Cnt==0)

    if(all_i_Cnt>0){
        for(;;){
            system("cls");
            Alarm(0);
            scanf("%d",&i_menu);
            if(i_menu==1){
                gotoxy(30,0);printf("전화 번호부 전체 출력");
                Sleep(700);

                system("cls");

                if(all_i_Cnt<=5){
                    for(i=0;i<all_i_Cnt;i++){
                        printf("\n이름: %s\n",sta_Info[i].ca_Name);
                        printf("\n나이: %d\n",sta_Info[i].ui_Age);
                        NumberChange(sta_Info[i].ca_Number,1); // 전화번호 출력 변경
                        printf("\n주소: %s\n",sta_Info[i].ca_Addr);
                        printf("-----");
                    } //for(i=0;i<all_i_Cnt;i++)
                    Alarm(18); // 출력 돌아가기 디자인 출력

                    gotoxy(71,19);scanf("%d",&i_back);

                    if(i_back==0){
                        Alarm(15); // 메뉴로 돌아가기 디자인 출력
                        break;
                    }
                } //if(all_i_Cnt<=5)

                //-----1-5명 일때 페이지수 1개-----
                if(all_i_Cnt>5 && all_i_Cnt<=10){
                    Output_page2();
                } //if(all_i_Cnt>4 && all_i_Cnt<=8) 두 페이지에 출력

            } //if(i_menu==1)

            if(i_menu==2){
                Alarm(15); // 메뉴로 돌아가기 디자인 출력
                break;
            }
        } //for(;;)
    } //if(all_i_Cnt>0)
} //void Outut(void)

```

[Output_page1,2()]-전화번호부 출력]

1. 전체 데이터를 출력하며 사용자의 편의를 위하여 2개의 페이지로 나눔. 저장된 데이터가 5개 이하일 경우 한 페이지만, 5개를 초과할 경우 두 번째 페이지를 생성함.

2. 데이터가 없을 시, "출력할 정보가 없습니다" 문구 표시.

3. '0'을 누르면 메뉴로 돌아감.

```
//아이디 수정 함수
void Edit()
{
    int i, i_Pass, i_idx=0, i_Mod; // i_Mod 수정할 전화번호부 선택 변수
    int i_idx_ModIdx = 0; // i_ModIdx : 수정할 전화번호부의 index번호
    gotoxy(30,10);printf("수정함");Sleep(500);
    Alarm(15); // 가리드 경고기

    for(i=0; i<all_i_Cnt; i++){
        system("cls");
        setColor(0,15);
        printf("%d.....수정 가능한 전화번호부 목록.....\n");
        for(i=0; i<all_i_Cnt; i++){
            if(sta_Info[i].i_Auth <= all_i_Auth){
                i_idx_ModIdx = i;
                printf("%d. 이름 : %s, ", i_idx+1, sta_Info[i].ca_Name);
                NumberChange(sta_Info[i].ca_Number, i); // 전화번호 출력 변경
                printf(" = 나이 : %d, 주소 : %s\n", sta_Info[i].ui_Age, sta_Info[i].ca_Addr);
                printf(".....\n");
                i_idx++;
            } //if(sta_Info[i].i_Auth <= all_i_Auth)
        } //for(i=0; i<all_i_Cnt; i++)
        if(i_idx == 0){
            gotoxy(20,15);printf("※수정 가능한 전화번호가 없습니다.※");
            Sleep(1000);
            break;
        }
        printf(".....백인으로 돌아가려면 0을 입력하세요.....\n");
        if(i_idx != 0){
            printf("번호입력>>>>> ");
            scanf("%d", &i_Mod);
            if(i_Mod == 0){
                Alarm(15); // 경고음 돌아가기 마지막 출력
                break;
            } else if(i_Mod < i_idx){
                gotoxy(20,20);printf("※잘못된 숫자를 선택하셨습니다.※");
                Sleep(1000);
                system("cls");
                i_idx = 0;
                continue;
            }
            system("cls");
            i_Pass = InfoModify(i_idx_ModIdx(i_Mod-1)); // 인자값은 전체 번호 리스트의 수정할 전화번호의 index일.
            if(i_Pass == -1){
                break;
            }
            i_idx = 0;
            system("cls");
        }
    }
    // Sleep(1000);
}
```

[Edit()-수정 가능 목록 함수]

1. 권한에 따른 수정 가능한 목록을 보여주며, 구조체 sta_info에서 해당 권한이 볼 수 있는 데이터만 불러옴.

2. 수정할 데이터를 선택할 시, 목록내에 없는 선택지이면 “잘못된 숫자를 선택하셨습니다” 문구가 뜸. 알맞게 선택 했을시 선택한 숫자를 InfoModify() 매개변수로 사용하며 InfoModify()에서 실행.

```
// 정보 수정 선택 함수
int InfoModify(int idx){
    int i_Select;

    Alarm(5);
    scanf("%d", &i_Select);
    system("cls");
    rewind(stdin);

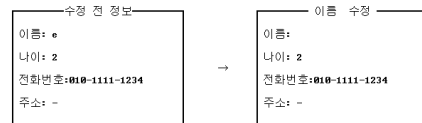
    if(i_Select == 1){
        AlarmModify(0, idx, i_Select); // 수정권 아이디 출력
        AlarmModify(1, idx, i_Select); // 수정권 아이디 출력
        gotoxy(50,8);scanf("%s", &sta_Info[idx].ca_Name);
    } else if(i_Select == 2){
        AlarmModify(0, idx, i_Select); // 수정권 아이디 출력
        AlarmModify(1, idx, i_Select); // 수정권 아이디 출력
        gotoxy(60,12);scanf("%s", &sta_Info[idx].ca_Number);
    } else if(i_Select == 3){
        AlarmModify(0, idx, i_Select); // 수정권 아이디 출력
        AlarmModify(1, idx, i_Select); // 수정권 아이디 출력
        gotoxy(50,10);
        while(!scanf("%d", &sta_Info[idx].ui_Age)){ // 수정권 입력 받기 위해 추가
            gotoxy(30,30);printf("※정수만 입력 가능 다시 입력 하세요.※");
            AlarmModify(0, idx, i_Select); // 수정권 아이디 출력
            AlarmModify(1, idx, i_Select); // 수정권 아이디 출력
            while (getchar() != '\n');
            gotoxy(50,10);
        }
    } else if(i_Select == 4){
        AlarmModify(0, idx, i_Select); // 수정권 아이디 출력
        AlarmModify(1, idx, i_Select); // 수정권 아이디 출력
        gotoxy(50,14);scanf("%s", &sta_Info[idx].ca_Addr);
    } else if(i_Select == 5){
        gotoxy(20,10);printf("※수정을 취소 하셨습니다.※");
        printf("수정을 취소 하셨습니다. \n");
        Sleep(1000);
        system("cls");
        return -1;
    } else{
        system("cls");
        printf("선택 가능 번호는 1-5 까지 입니다.\n");
        InfoModify(idx);
    }
    return 0;
}
```

[InfoModify()-전화번호부 수정 메서드]

1. 매개변수 idx와 일치하는 sta_info배열의 인덱스에 있는 데이터를 출력함.

2. 데이터 수정은 아래 사진과 같이 이름, 나이, 전화번호, 주소 중 선택해서 수정 가능.

3. 데이터 수정시, 사용자 편의를 위해 수정전 데이터도 함께 보여줌.



```
//레이더 검색 함수
void Search(){
    int i,j,k;
    int i_Cnt=0;
    int i_Scnt=0;
    int i_Search_Menu;
    unsigned int ui_Tmp;
    char ca_Tmp[SIZE];
    char ca_OHP_Tmp[SIZE];
    char ca_back[10];
    for(;;){
        system("cls");
        Alarm(6);

        scanf("%d",&i_Search_Menu);
        system("cls");
        if(i_Search_Menu==1){ // 검색모드 디자인 출력
            AlarmSearch(0); // 검색모드 디자인 출력
            gotoxy(36,3);scanf("%s",&ca_Tmp);
            for(i=0;i<allI_Cnt;i++){
                if(strcmp(sta_Info[i].ca_Name,ca_Tmp)==0){
                    i_Cnt[i_Scnt]=i;
                    i_Scnt++;
                }
            }
            //for(i=0;i<allI_Cnt;i++){
            for(i=0;i<i_Scnt;i++){
                gotoxy(i,1);printf("%d. 이름 : %s |",i+1,sta_Info[i_Cnt[i]].ca_Name);
                Numberchange(sta_Info[i_Cnt[i]].ca_Number,i); // 검색모드 출력 변경
                printf(" | 나이 : %d | 주소 : %s | \n",sta_Info[i_Cnt[i]].ui_Age,sta_Info[i_Cnt[i]].ca_Addr);
                printf("-----");
            }
            //for(;;){i_Scnt++;}
            if(i_Scnt==0){
                gotoxy(38,5);printf("※찾는 이름이 없습니다.※");
                Sleep(600);
                system("cls");
            }
            Alarm(20);
            gotoxy(1,23);printf("돌아가기(아무키)"); scanf("%s",&ca_back);
            i_Scnt=0;
        }
        //if(i_Search_Menu==1) 아무 검색 끝.
        else if(i_Search_Menu==2){
            AlarmSearch(1); // 검색모드 디자인 출력
            gotoxy(41,3);scanf("%s",&ca_Tmp);
            for(i=0;i<allI_Cnt;i++){
                if(strcmp(sta_Info[i].ca_Number,ca_Tmp)==0){
                    i_Cnt[i_Scnt]=i;
                    i_Scnt++;
                }
            }
        }
    }
}
```

[Search() – 전화번호부 검색 메서드]

1. 검색메뉴에는 이름, 전화번호, 나이, 주소로 검색할 수 있음. strcmp()함수로 구조체sta_info배열에서 검색한 정보와 일치하는 모든 데이터를 출력함.

검색 모드
이름 : 장은희

1. 이름 : 장은희 | 전화번호: 010-1111-2222 | 나이 : 30 | 주소 : 경기도 여주시 |

돌아가기<아무키> BACK ↩

```
// 실행 함수
void run(){
    int i_Login_Menu; //로그인메뉴변수
    int i_Menu; //메인메뉴
    int i_LoginIndex; //로그인 사용자 index 번호

    for(;;){ //프로그램시작
        i_Login_Menu=Login_Menu(); //메뉴함수호출후 메뉴리턴
        system("cls"); // - kom 추가
        if(i_Login_Menu==1){
            i_LoginIndex=Login();
            if(i_LoginIndex == -1){ // 로그인 3회 실패시 프로그램 종료
                Alarm(21);
                break;
            }
            allI_Auth=sta_Login[i_LoginIndex].i_Auth;
            strcpy(allca_Name,sta_Login[i_LoginIndex].ca_Name);
            Alarm(1); // 로그인 완료
            Sleep(2000);
            for(;;){
                i_Menu=Menu();
                system("cls");
                if(i_Menu==1){
                    Input();
                }
                else if(i_Menu==2){
                    Edit();
                }
                else if(i_Menu==3){
                    Delete();
                }
                else if(i_Menu==4){
                    Output();
                }
                else if(i_Menu==5){
                    if(allI_Auth==3){
                        Search();
                    }
                    else {
                        gotoxy(20,18);printf("※현재 권한으로는 검색이 불가능합니다.※");
                        Sleep(1000);
                    }
                }
                else if(i_Menu==6){
                    Alarm(6);
                    Sleep(2000);
                    setColor(7,6);
                    break;
                }
                else{
                    Alarm(7);
                    Sleep(1000);
                }
            }
        }
        //for(;;) - 메인메뉴
        else if(i_Login_Menu==2){
            Regit();
            system("cls");
            gotoxy(25,18);printf("※사용자등록이 완료되었습니다.※");
            Sleep(1000);
        }
        else if(i_Login_Menu==3){
            Alarm(21); // 종료인사 메시지
            Sleep(1000);
            break;
        }
        else{
            Alarm(7);
            Sleep(1000);
        }
    }
} // for(;;) 전체 프로그램
```

[run() – 전체 프로그램 실행 함수]

1. Login_Menu()함수로 메뉴 호출 후 리턴된 값에 따른 함수를 실행함.

2. 구조체Sta_Login배열의 사용자 이름이 담긴 정보를 strcpy()함수를 통해 전역변수allca_Name배열에 복사해 넣음. allca_Name변수는 사용자가 로그인 시 해당 사용자 이름을 띄우기 위해 사용됨.

5. 어려웠던 점 및 느낀 점

<p>어려웠던 점</p>	<p>1. 처음 진행하는 팀 프로젝트 처음 접해보는 팀 프로젝트로, 초기에는 체계적이지 못하였지만, 서로 많은 소통을 통하여 각자 알맞은 역할을 담당하여 원활히 프로젝트를 진행할 수 있었음.</p> <p>2. 다른 팀원 코드 해독 어려움 자신이 짰 코드 외에 다른 사람 코드를 처음 접하므로, 코드 해독에 있어서 어려움이 있었음. 하지만 팀원이 달은 주석을 참고 하며 계속해서 코드를 본 결과 팀원들과 함께 원하는 방향으로 프로젝트를 완수할 수 있었음.</p>
<p>느낀 점</p>	<ul style="list-style-type: none"> - 팀 프로젝트에서 어느 팀원이 어느 부분을 담당하였는지 주석으로 알리는 것이 필요함. - 팀원들과의 협업을 위하여 기능마다 자신의 의도를 주석으로 알리는 것이 중요함. - 초기에 팀원들과 지역변수 및 전역변수를 구분하기 쉽도록 변수명을 지정해야 됨.
<p>맡은 역할</p>	<ol style="list-style-type: none"> 1. 전화번호부 입력 2. 전화번호부 전체 출력 3. 콘솔 전체 디자인 4. 버그 점검 및 수정