**전화번호부 설계**



**소프트웨어공학**

**권세진 교수**

**Ai소프트웨어학과 2학년**

**201920276**

**이봉규**

**목차**

**1.요구분석**

**-개요**

**-포함해야 할 기능**

**2.설계**

**-각 기능별 설계 flowchart**

**3.구현**

**-전체 flowchart**

**-주요 함수 설명**

**-각 기능별 구현**

**4.테스트 결과**

**-명령어 삽입 테스트 스크린샷**

**-결과분석**

**1. 요구분석**

**-개요**

기술이 발전함에 따라 누군가의 전화번호가 필요할 때, 책으로 된 전화번호부를 펼치는 사람은 이제 찾아볼 수 없다. 길을 외우고 다니거나 길 안내가 필요할 때 지도를 펼쳐보거나, 전달해야 하는 말을 편지로 보내는 사람은 쉽게 찾아볼 수 없듯, 두껍고 불편한 책으로 되어있는 전화번호부 대신 자신이 들고 다니는 휴대기기에 수많은 사람의 전화번호를 기록하고 찾을 수 있도록 전화번호부가 필요하게 되었다.

전화번호부란 기억력에 한계가 있는 인간을 대신해 사람들의 연락처를 기록하며 기존에 쓰이던 종이로 된 전화번호부보다 여러 가지 정보를 보다 쉽게 추가하며, 수정이 간단하고 필요한 연락처만을 검색 기능을 활용하여 빠르게 열람할 수 있다는 장점이 있다. 현대인의 필수품인 스마트폰엔 모두 기본으로 들어가 있으며, 최근에는 핸드폰이랑 연동할 수 있는 스마트패드, 랩톱, 데스크톱에도 전화번호부가 탑재된 경우가 있다.

전화번호부에 필요하다고 생각하는 기능들은 다음과 같다.

**-전화번호 추가**

자신이 추가하고자 하는 연락처를 쉽고 빠르게 저장 할 수 있어야 한다. 사용자에게 이름과 전화번호를 직접 입력받아 저장한다.

**-전화번호부 리스트 출력**

지금까지 저장된 연락처들을 한눈에 볼 수 있도록 데이터베이스에 있는 정보들을 일정량 화면에 출력하는 기능이다. 데이터에 저장된 연락처가 많아질수록 리스트를 활용하기 어려워지지만, 연락처가 많지 않을 때는 한눈에 모든 연락처를 확인할 수 있는 리스트가 훨씬 편리하다. 리스트에 있는 데이터를 불러왔을 때 한 화면에 다 안 나올 정도로 많은 데이터가 한 번에 출력된다면 오히려 찾기 어려워질 수 있으므로 일정량 잘라서 일부 데이터만 한 화면에 출력하고 나머지 데이터는 다음 페이지로 출력한다.

**-전화번호 삭제**

사용자가 추가한 연락처를 삭제할 수 있는 기능이다. 이제는 사용하지 않는 연락처나 잘못 기입된 연락처를 삭제하며 전화번호부를 정리하는데 사용한다.

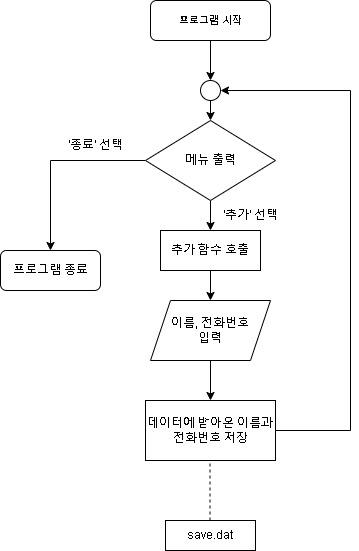
**-전화번호 검색**

한눈에 리스트를 볼 수 있을 정도면 문제가 생기지 않으나, 한눈에 볼 수 없을 정도로 전화번호부에 사람이 많아지거나 빠르게 찾기를 원한다면 검색 기능을 활용해야 한다. 사용자는 이름이나 전화번호를 직접 입력했을 때, 입력한 내용이 포함된 연락처를 쉽게 검색할 수 있다.

**-전화번호 수정**

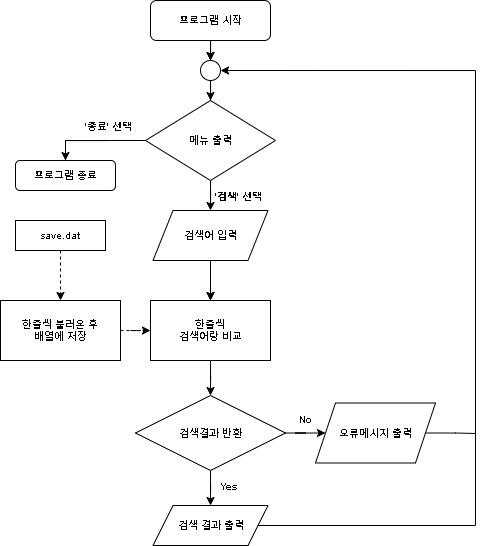
만약 전화번호부에 추가 할 때 연락처를 삭제하려 하거나 이름이나 번호를 수정해야 할때, 전화번호부에서 연락처를 찾은 후에 수정할 수 있어야 한다. 기본적으로 검색 기능이나 리스트 출력 기능을 같이 사용하여 연락처를 검색한다. 그 후에 나온 결과물을 수정할 것인지 완전히 삭제시킬 것인지 사용자는 선택할 수 있다.

**2.설계**



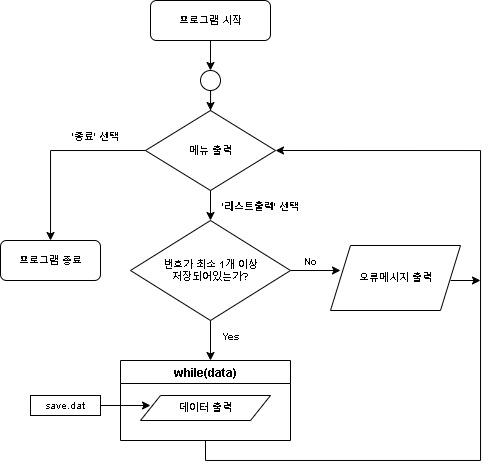
**-전화번호 추가**

가장 기본적으로 전화번호 추가는 사용자에게서 이름과 전화번호를 받아와 그것을 데이터에 저장한다. 사용자가 ‘전화번호 추가’ 항목을 선택하면 string 형식으로 읽어온 후, 개개인의 연락처를 기록한다. 데이터베이스에 개행을 기준으로 구분하여 기록하여서 프로그램을 종료하여도 연락처가 남아있으며, 후에 검색이나 수정 처리를 해야 할 때에도 불러오는 작업이 수월해진다.



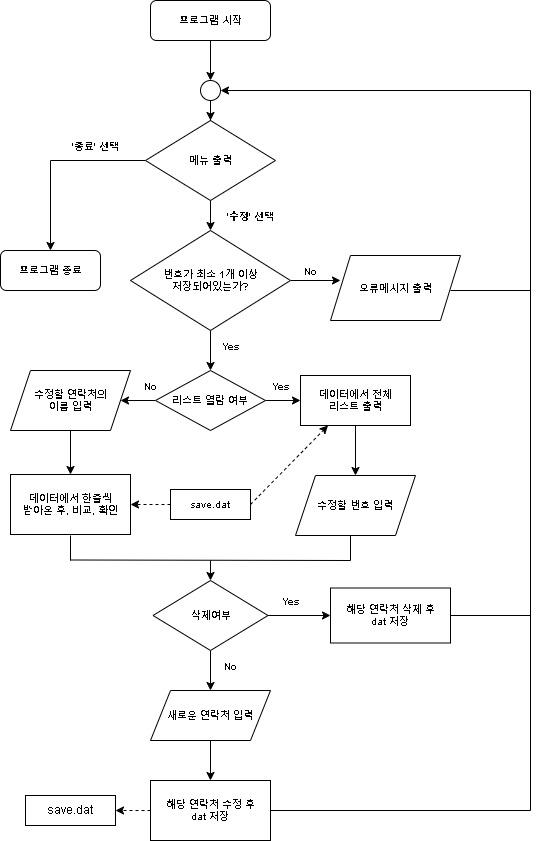
**- 전화번호 검색**

사용자가 메뉴에서 ‘검색’ 기능을 실행하면 사용자가 직접 검색어를 입력하여 연락처를 탐색하는 기능이다. 기존의 데이터파일에서 한 명분의 데이터를 char 배열에 정리한 후, 정리된 데이터를 사용자가 입력한 검색어랑 비교하여 검색어가 포함된 연락처가 있다면 그것들을 모두 출력한다. 이름뿐만 아니라 전화번호로 검색했을 때에도 결과를 출력할 수 있으며, 같은 검색어를 포함한 모든 결과물을 리스트로 정리하여 사용자에게 출력한다. 검색 결과가 없을 시에는 검색어를 포함한 결과를 찾을 수 없다는 오류메시지를 사용자에게 출력한다.



**-리스트 출력**

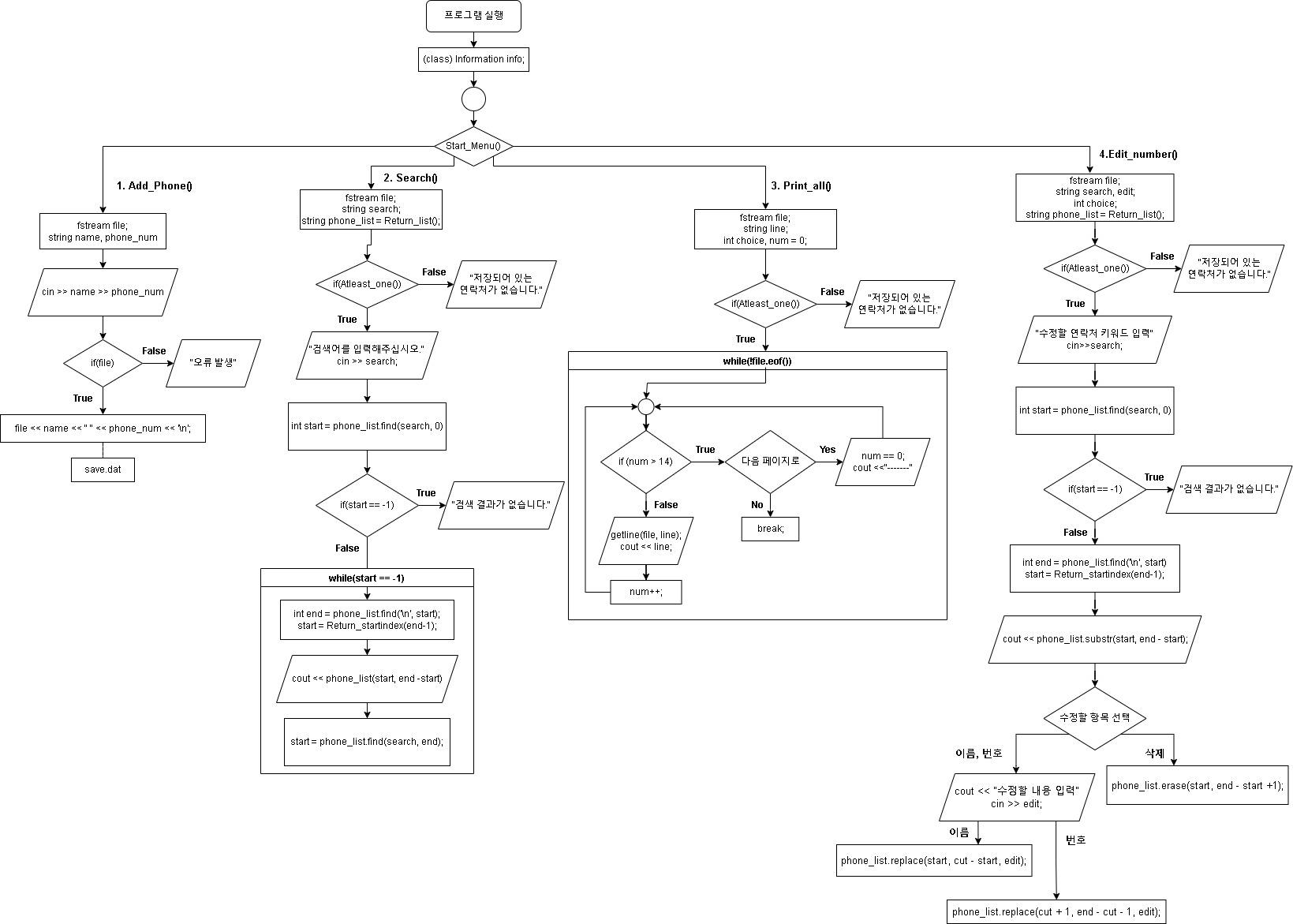
사용자가 전화번호부 리스트 출력을 원할 때, 메뉴에서 리스트 출력을 선택할 수 있다. 프로그램이 사용자에게 리스트 출력을 입력받으면, 프로그램은 일단 저장된 연락처가 있는지 확인한 후, 없으면 오류메시지를 출력하게 된다. 만약 저장되어 있는 연락처가 1개 이상일 경우, 프로그램은 데이터에서 while문을 통하여 마지막 줄을 리턴 받을 때까지 데이터를 받아오고, 출력하는 과정을 반복한다.



**- 연락처 수정**

사용자가 입력한 정보들을 찾아서 그 정보를 수정할 수 있다. 사용자가 메뉴에서 ‘수정’ 기능을 실행하면 제일 처음으로 데이터베이스에 저장된 데이터가 최소 1개 이상 있는지 검사한다. 없다면 오류메시지를 출력한다.

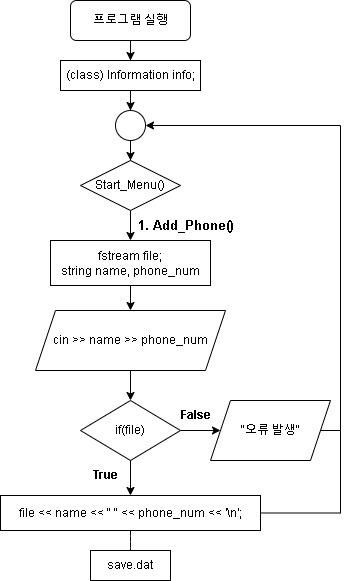
그 후 사용자는 리스트를 출력하여 선택할 것인지, 검색 기능을 이용하여 선택할 건지 입력한다. 만약 찾으려는 데이터와 일치하는 데이터가 있다면 프로그램은 그 데이터를 출력하고 삭제할 것인지, 수정하여 다시 저장할 것인지 사용자의 선택에 따라서 작동하게 된다. 삭제할경우 삭제한 항목은 문자열에서 지워지게 되고, 정리된 문자열은 다시 데이터 파일에 뒤집어 씌워지게 된다.

**3.구현**

**1. 중요함수 설명**

| **Start\_menu()**: 프로그램 실행 시에 제일 먼저 출력되며, 메뉴를 출력하고 사용자에게 번호를 입력받아 그에 맞는 기능 실행.  **Add\_Phone()**: 전화번호부에 새로운 연락처를 추가하는 함수. 사용자에게 새롭게 연락처에 추가할 이름과 전화번호를 띄어쓰기로 구분하여 입력받아 데이터의 마지막에 추가함.  **Search():** 전화번호부에서 원하는 연락처를 검색하는 함수. 실행 시에 사용자에게 검색어를 입력받으며, 데이터에 저장되어있는 연락처가 있는지 확인 후, 데이터의 연락처를 문자열로 옮겨 넣음. 그 후에 전체 문자열에서 검색어를 포함한 연락처가 있는지 검색 후 있으면 그것들을 출력함.  **Print\_all():** 전체 연락처를 출력해주는 함수. 실행 시에 데이터에 저장되어있는 연락처가 있는지 확인 후, 15줄 단위로 데이터에서 연락처를 한 줄씩 받아와 출력, 15줄이 넘으면 다음 페이지 15줄을 출력할 것인지 출력 기능을 정지할 것인지 선택할 수 있다.  **Edit\_number()**: 기존에 있던 연락처를 수정하는 함수. 데이터에 저장되어있는 연락처가 있는지 확인 후, 데이터의 연락처를 문자열로 옮겨 넣음. 사용자에게 수정할 연락처를 입력받고, 검색기능으로 존재하는지 확인, index 위치 연산을 마친 후에 string.replace 기능으로 연락처를 수정함. 사용자의 선택에 따라 string.erase 기능으로 그 연락처를 삭제할 수도 있다.  **Save\_array(string)**: 수정 기능으로 수정된 연락처가 포함된 문자열을 데이터에 다시 저장하는 함수. 이때 원래 데이터에 있던 내용은 모두 삭제되고 받아온 문자열로 대체되고 데이터 파일이 저장된다.  **bool Atleast\_one():** 데이터에 최소 1개 이상의 연락처가 있는지 검사 후 bool 형식으로 리턴하는 함수.  **string Return\_list()**: 데이터에 있는 모든 문자를 불러와 문자열에 저장하는 함수. 크기에 맞게 리사이즈한 문자열에 저장 후, 그 문자열을 리턴한다.  **int Return\_startindex(int)**: 검색과 수정기능에서 이름이나 번호를 수정할 때, 검색어가 단어 중간에 있어 첫 index가 중간으로 잡혀 이름이나 번호 일부만 수정이나 검색되는 것을 방지하기 위한 함수. 연락처 한 줄의 index를 넣으면 개행문자를 기준으로 그 줄의 처음을 찾아 인덱스를 리턴한다.  ex) “디지털논리회로 01012345678” 에서 사용자가 검색창에 ‘논리’ 검색 ->  원래 string.find 기능으로 ‘논리’가 포함되어 start가 ‘논’ 따로 계산한 end가 ‘8’이 되어  “논리회로 01012345678”이 출력되지만 이 함수를 거치면 start를 줄의 맨 처음으로 옮긴다.  그리하여 “디지털논리회로 01012345678”이 정상적으로 출력됨. |
| --- |

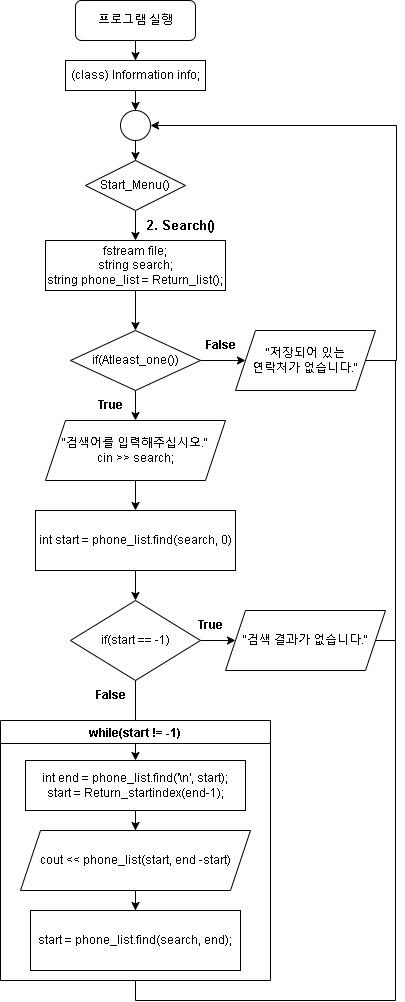
**2. 각 기능 구현부분 설명**

**-연락처 추가**

1.프로그램이 실행된 후, 연락처 추가를 선택하면 file을 불러오고 변수를 선언한다.

2.사용자에게 이름과 전화번호를 직접 입력받는다.

3.파일이 정상적으로 열리는지 검사[ if(file) ]를 실행 후 이상이 없으면 빈칸과 개행문자를 추가하여 데이터에 저장한다.

**-전화번호 검색**

1. 프로그램이 실행된 후, 연락처 검색을 선택하면 file을 불러오고 변수를 선언한다.

2. Return\_list 함수로 데이터에 있는 연락처들을 전부 문자열로 받아온다.

3. Atleast\_one 함수로 데이터에 최소 1개의 연락처가 있는지 검사한다.

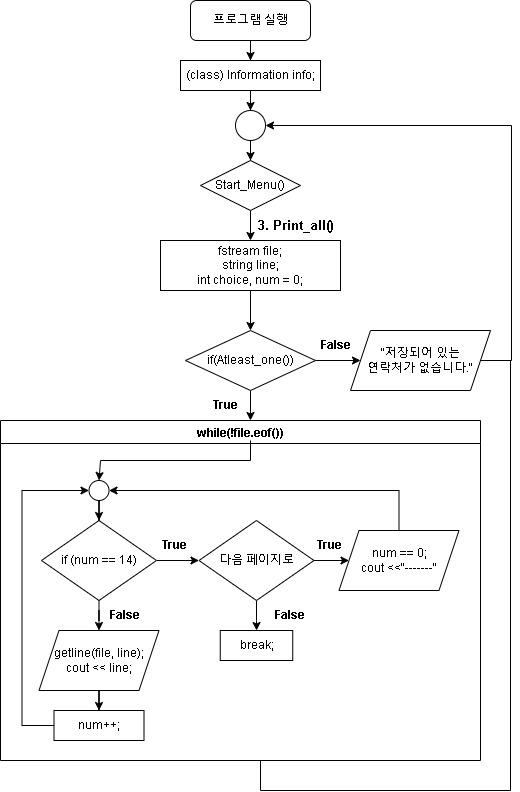
4. True라면 사용자에게 검색 키워드를 입력받는다.

5. string.find 기능으로 검색 키워드가 있는 곳의 첫 index를 리턴받는다.

6. 만약 키워드 단어가 존재한다면 그 단어의 첫 index를 반환하고 찾을 수 없다면 -1을 반환하는데, 이것을 판단하여 검색 결과가 존재하는지 확인한다.

7. 개행문자를 첫 index가 있는 줄에서 찾아서 문장의 마지막을 찾고, Return\_startindex 함수로 start를 문장 앞으로 옮긴다.

8. 문장을 출력한 후, 문장의 마지막에서부터 다시 검색하여 중복된 키워드가 들어간 문장을 검색한다.

**-전화번호부 출력**

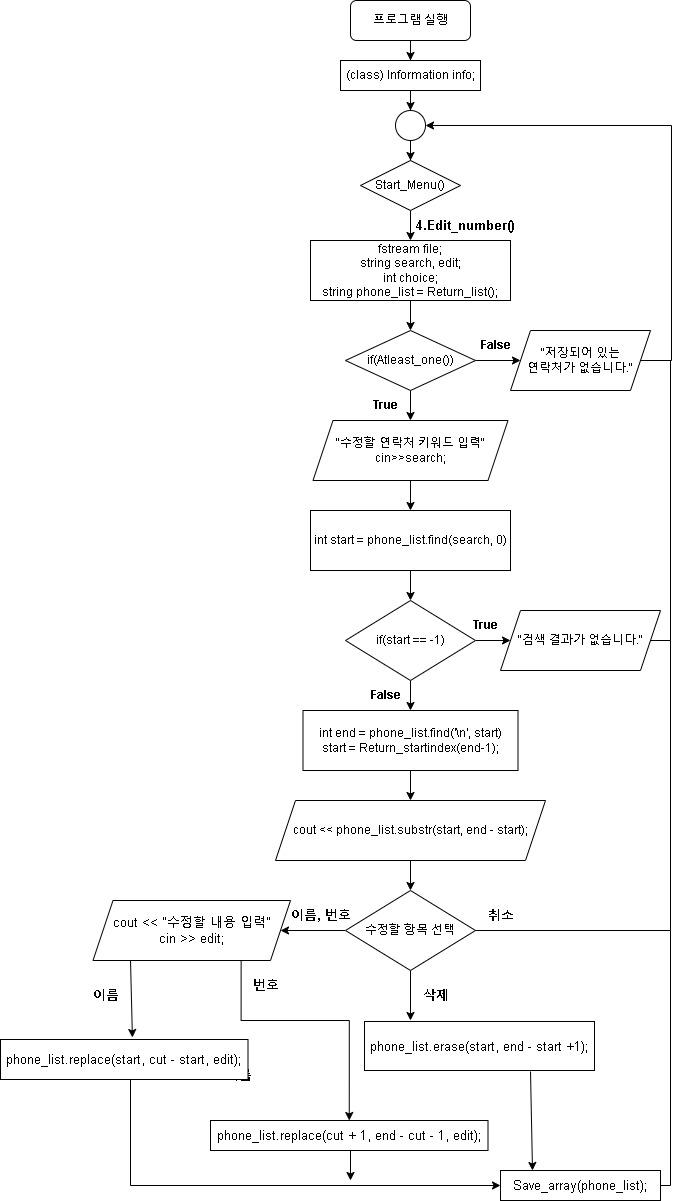
1. 프로그램이 실행된 후, 연락처 검색을 선택하면 file을 불러오고 변수를 선언한다.

2. Atleast\_one 함수로 데이터에 최소 1개의 연락처가 있는지 검사한다.

3. 가져온 연락처의 수가 15개가 될 때까지 한 줄씩 데이터에서 받아오고 출력을 반복한다.

4. 가져온 연락처의 수가 15개가 되면 페이지를 더 출력할 건지 종료할 것인지 사용자에게 입력받는다.

5. 데이터의 마지막에 도달하게 되거나 사용자에게 종료를 입력받으면 반복문이 종료된다.



**-연락처 수정**

1. 프로그램이 실행된 후, 연락처 검색을 선택하면 file을 불러오고 변수를 선언한다.

2. Return\_list 함수로 데이터에 있는 연락처들을 전부 문자열로 받아온다.

3. Atleast\_one 함수로 데이터에 최소 1개의 연락처가 있는지 검사한다.

4. True라면 사용자에게 검색 키워드를 입력받는다.

5. string.find 기능으로 검색 키워드가 있는 곳의 첫 index를 리턴받는다.

6. 만약 키워드 단어가 존재한다면 그 단어의 첫 index를 반환하고 찾을 수 없다면 -1을 반환하는데, 이것을 판단하여 검색 결과가 존재하는지 확인한다.

7. 개행문자를 첫 index가 있는 줄에서 찾아서 문장의 마지막을 찾고, Return\_startindex 함수로 start를 문장 앞으로 옮긴다.

8. 해당 문장을 출력하고 사용자에게 선택사항을 입력받는다.

9. 삭제를 선택하면 해당 연락처를 삭제하고 메뉴로 돌아간다.

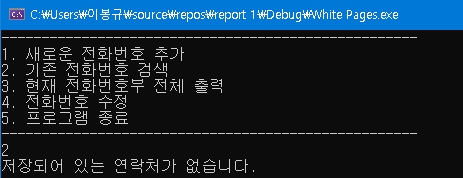
10. 이름 및 전화번호 수정을 선택하면 원래 있던 연락처에서 변경할 내용을 입력받는다.

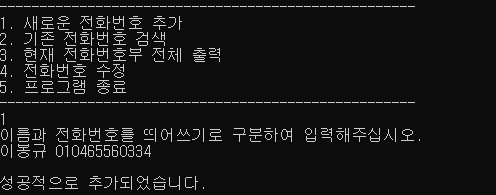
11. 해당 문장에 있는 띄어쓰기 인덱스를 가지는 cut 변수로 이름 부분, 번호 부분을 분리 후에 수정한다.

12. Save\_array함수로 수정된 문자열을 데이터에 저장한다.

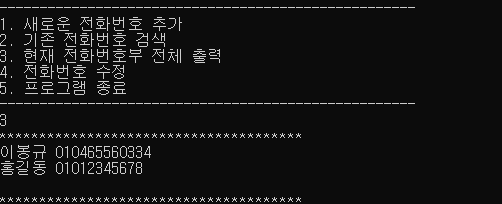
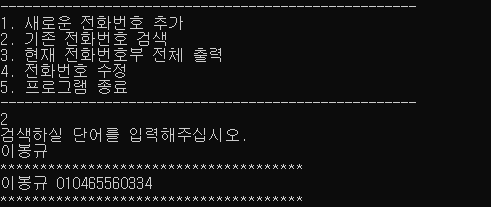
**4.테스트결과**

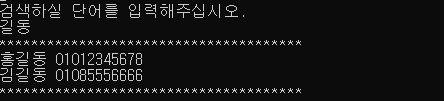
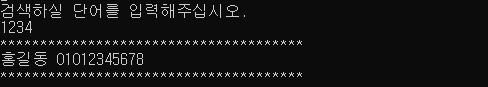
**1.명령어 삽입 테스트 스크린샷**

-저장되어 있는 연락처가 없을때

-연락처 추가

-연락처가 있을때 연락처 출력



-연락처 검색

-연락처 수정

-연락처 삭제

**2.결과분석**

-요구 분석, 설계와는 다르게 데이터에서 연락처를 한 줄씩 받아오는 것 대신 전체를 복사해서 수정하는 방식으로 설계와 다르게 프로젝트가 진행되었다.

-제일 큰 문제점으로 성은 다른데 이름이 다르거나 등등의 사유로 한 검색어로 두 개 이상의 검색 결과가 나왔을 때, 그중에서 선택하여 수정할 수 있는 시스템을 구현하지 못하였다. 현재는 여러 가지 검색 결과 중에서 제일 처음으로 나오는 결과밖에 수정하지 못한다.

현재 구현한 건 단순하게 문자열을 받아오고 문자열 내에서만 검색, 수정이 이루어지다 보니 이러한 처리가 힘들다고 생각한다.