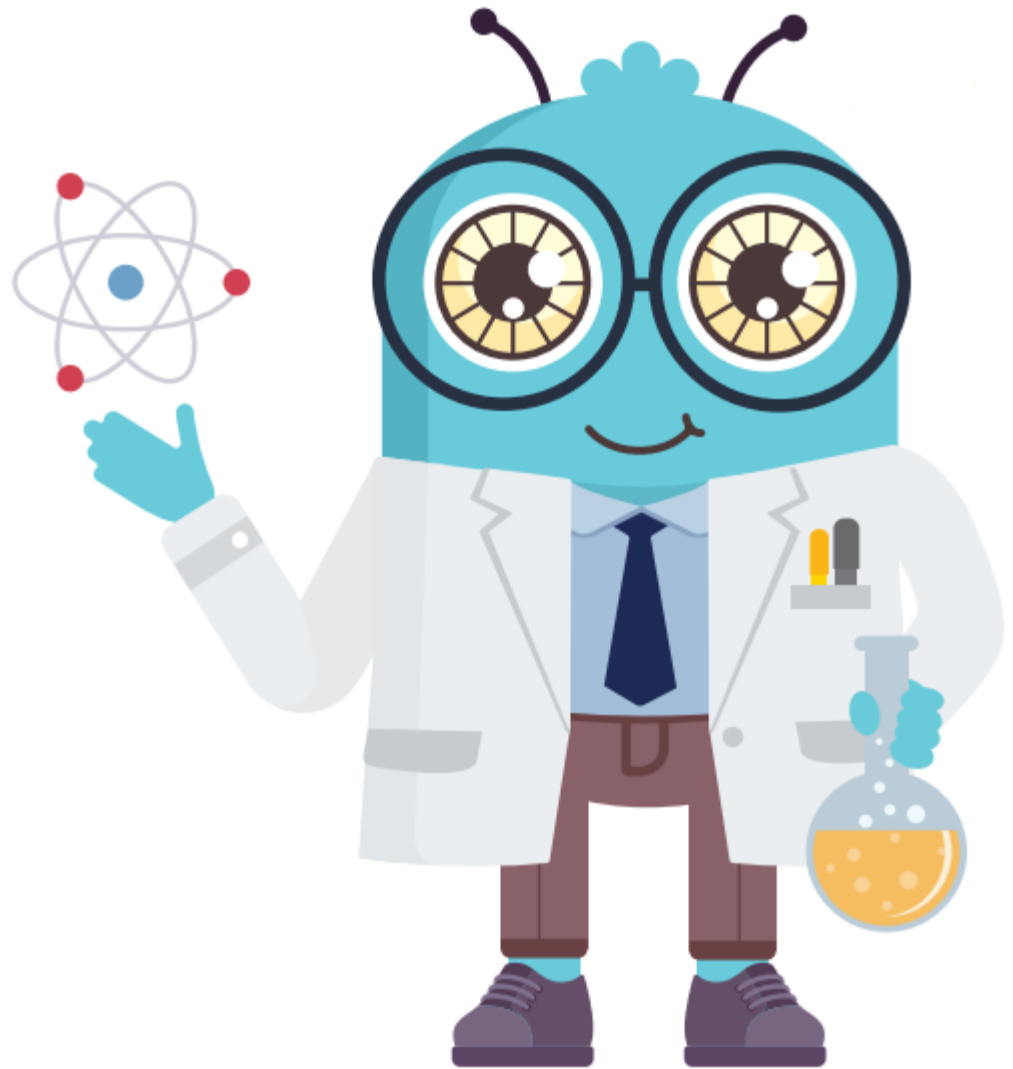


Chapter 15

실전 프로젝트



목차

1. 연락처 관리 툴 Ver 1.0
2. 사진 편집 툴 Ver 1.0

01

연락처 관리 툴 Ver 1.0

1. 연락처 관리 툴 Ver 1.0

1. 프로그램 개요

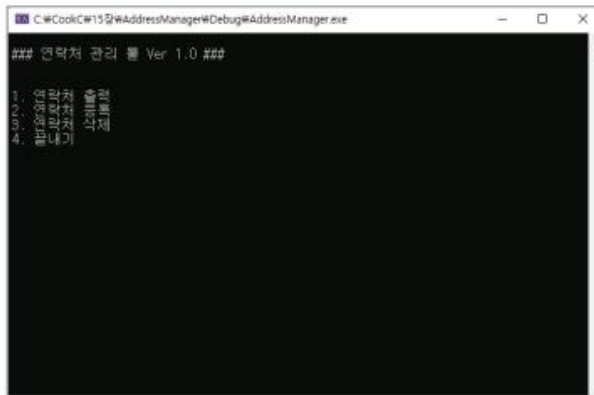
▪ 연락처 관리 프로그램

- 실행 4가지 항목 중 한가지 숫자를 입력하면 해당 작업이 실행됨
 - 1. 연락처 출력
 - 2. 연락처 등록
 - 3. 연락처 삭제
 - 4. 끝내기
- 처음에는 입력된 연락처가 없으므로 2번을 눌러 필요한 연락처를 입력함
- 입력된 내용은 'C:\temp\Wjuso.txt'에 저장
- 프로그램을 마치려면 '4. 끝내기'를 선택

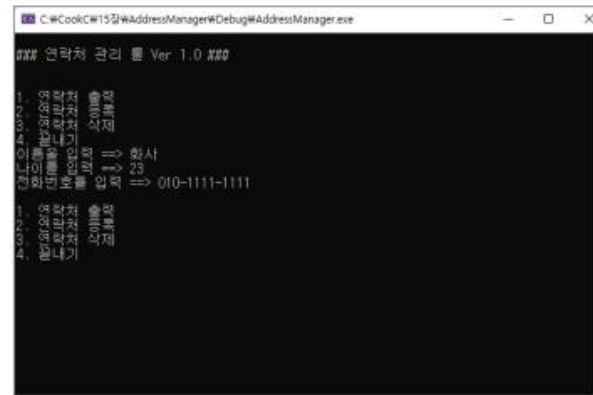
1. 연락처 관리 툴 Ver 1.0

1. 프로그램 개요

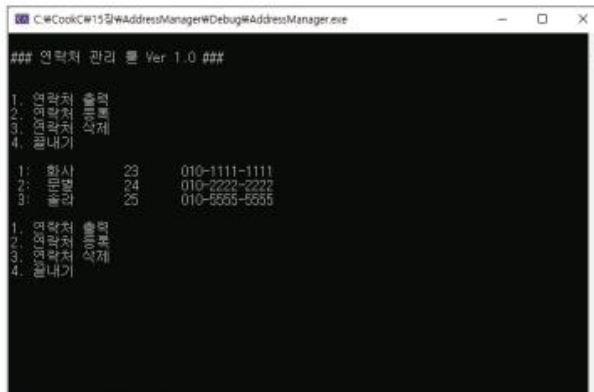
■ 연락처 관리 프로그램



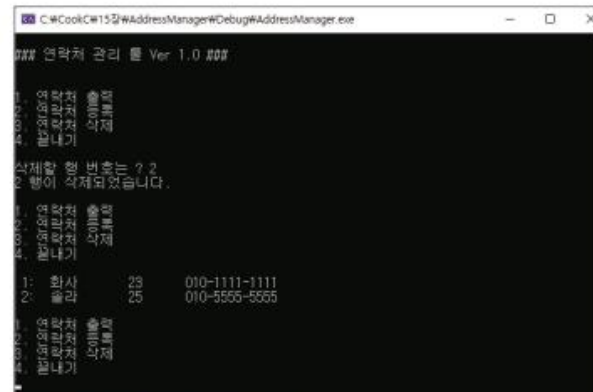
프로그램 초기 화면



연락처 등록 화면



저장된 연락처 보기



연락처 삭제 후 현재 연락처 출력

그림 15-1 연락처 관리 툴 Ver 1.0 실행 화면

1. 연락처 관리 툴 Ver 1.0

2. 프로그램 구현

- ① 친구의 연락처는 파일에 저장해서, 프로그램을 종료해도 기록이 남음
→ 파일 처리 함수인 `fopen_s()`, `fclose()`, `fgets()`, `fputs()` 함수를 사용
- ② 연락처를 출력, 입력, 삭제하는 기능들은 별도의 함수로 작성하며, 4가지 중 하나를 선택할 수 있는 메뉴 화면도 별도의 함수로 작성
- ③ `main()` 함수에서는 `while`문을 써서 사용자가 종료를 선택할 때까지 반복하는 기능을 넣고, 사용자의 입력에 따라 별도의 함수를 각각 호출
→ `while`문 및 `switch~case`문을 사용

1. 연락처 관리 툴 Ver 1.0

2. 프로그램 구현

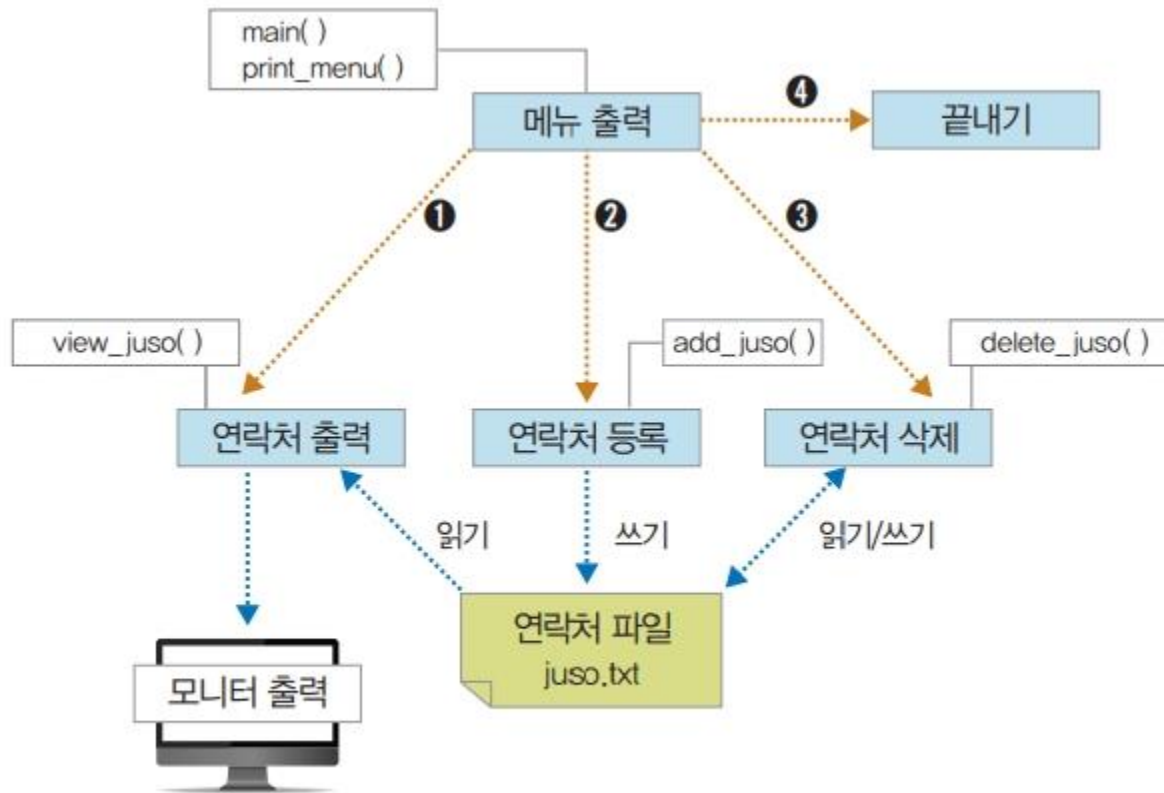


그림 15-2 프로그램의 전체 흐름

1. 연락처 관리 툴 Ver 1.0

■ 선언부

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 #include <string.h>
04
05 // 연락처 구조체를 정의한다.
06 struct address {
07     char name[30];
08     char age[5];
09     char phone[15];
10 };
11
12 void print_menu( );
13 void view_juso( );
14 void add_juso( );
15 void delete_juso( );
16
17 // 연락처 파일명을 전역변수로 고정한다.
18 const char *fname = "c:\\temp\\juso.txt";
```

①

②

③

1. 연락처 관리 툴 Ver 1.0

- ① 연락처 구조체를 정의. 연락처는 이름/나이/전화번호 세 개로 구성했으며, 필요하다면 주소나 이메일 등의 다른 내용을 추가해도 됨
- ② 프로그램에서 사용될 함수 4개를 선언- `print_menu()` : 초기의 메뉴화면을 출력
 - `view_juso()` : 현재 저장된 `C:\temp\juso.txt` 파일에서 연락처를 읽어서 화면에 출력
 - `add_juso()` : 새로 입력하는 연락처를 `C:\temp\juso.txt` 파일에 저장
 - `delete_juso()` : 사용자가 선택한 연락처를 파일에서 삭제함
- ③ 사용자가 입력할 정보를 `C:\temp\juso.txt` 파일에 저장하기 위해서 파일명을 변수에 고정함

1. 연락처 관리 툴 Ver 1.0

▪ main() 함수

```
01 void main ( )
02 {
03     char select=0; ①
04     puts(" \n### 연락처 관리 툴 Ver 1.0 ### \n");
05
06     while(select != 4) ②
07     {
08         print_menu( ); ③
09         select = getch( );
10
11         switch(select) ④
12         {
13             case '1': view_juso( );
```

1. 연락처 관리 툴 Ver 1.0

- main() 함수

```
14         break;
15         case '2': add_juso( );
16         break;
17         case '3': delete_juso( );
18         break;
19         case '4': return;
20         break;
21         default: printf( "\n 잘못 입력했어요. 다시 선택하세요.\n" );
22     }
23 }
24
25 return;
26 }
```

1. 연락처 관리 툴 Ver 1.0

- ① 사용자가 선택하는 번호를 저장하기 위해 select 변수를 선언, 초깃값으로는 4(종료)가 아닌 임의 값을 넣으면 됨. 여기서는 그냥 0을 넣음
- ② 사용자가 입력한 값이 4가 아닌 동안에는 계속 반복
- ③ 8행 - 사용자에게 보여줄 메뉴를 출력
9행 - 사용자로부터 1~4까지 입력 값을 받음
- ④ 사용자가 입력한 값에 따라 각각 다른 기능을 하는 함수를 호출
사용자가 4를 입력하면 25행의 return문에 의해 프로그램을 종료
1~4 이외의 값을 사용자가 입력하면 21행의 오류 메시지를 출력하고 다시 6행으로 가서 처음부터 반복함

1. 연락처 관리 툴 Ver 1.0

▪ 초기 메뉴 출력

```
01 // 최초 사용자의 선택을 위한 메뉴를 출력한다.  
02 void print_menu( )  
03 {  
04     printf("\n");  
05     printf("1. 연락처 출력 \n");  
06     printf("2. 연락처 등록 \n");  
07     printf("3. 연락처 삭제 \n");  
08     printf("4. 끝내기 \n");  
09 }
```

❶ 단순히 화면에 사용자가 선택할 메뉴를 도움말로 출력

사용자는 이 메뉴 화면을 보고 원하는 기능을 선택함

1. 연락처 관리 툴 Ver 1.0

■ 연락처 출력

```
01 // 연락처 파일에서 기존에 입력된 내용을 읽어서 출력한다.
```

```
02 void view_juso( )
```

```
03 {
```

```
04     char str[200] = "";
```

```
05     FILE *rfp, *wfp;
```

```
06     int i;
```

```
07
```

```
08     rfp=fopen(fname, "r");
```

```
09
```

```
10     // 만약 연락처 파일이 없다면 빈 파일을 만든다.
```

```
11     if(rfp == NULL)
```

```
12     {
```

```
13         wfp=fopen(fname, "w");
```

```
14         fclose(wfp);
```

```
15         rfp=fopen(fname, "r");
```

```
16     }
```

```
17
```

①

②

1. 연락처 관리 툴 Ver 1.0

■ 연락처 출력

```
18 // 기존의 연락처를 모두 읽어서 출력한다.
19 for(i = 1; ; i++) // i는 계속 1씩 증가하는 무한 루프이다. ③
20 {
21     fgets(str, 200, rfp);
22
23     iffeof(rfp))
24         break;
25
26     printf("%2d: %s", i, str);
27 }
28
29 // i가 1이면 실제 파일에는 내용이 없다.
30 if(i == 1) ④
31     puts("\n ** 연락처 파일에 전화번호가 하나도 없어요. **\n");
32
33 fclose(rfp);
34 }
```

1. 연락처 관리 툴 Ver 1.0

- ① 파일 포인터를 정의하고 파일을 읽기 모드로 오픈
- ② 최초로 이 프로그램을 실행하면 비어 있는 juso.txt 파일을 자동으로 생성해 줌
- ③ 무한루프. 26행에서 연락처의 순번(i)과 연락처 내용을 함께 출력
- ④ 이때 juso.txt 파일에 저장된 내용이 없다면 23행, 24행에서 i값이 1인 상태로 for 문을 빠져 나옴
- ⑤ i값이 1이면 31행에서 전화번호가 하나도 없다는 메시지를 출력

1. 연락처 관리 툴 Ver 1.0

■ 연락처 추가

```
01 // 연락처를 추가한다.
02 void add_juso( )
03 {
04     struct address adr = {"", "", ""};
05     char wstr[200] = "";
06     FILE *wfp;
07
08     wfp=fopen(fname, "a"); // 파일을 추가 모드로 연다. — ①
09
10     printf("이름을 입력 ==> "); — ②
11     gets(adr.name);
12     printf("나이를 입력 ==> ");
13     gets(adr.age);
14     printf("전화번호를 입력 ==> ");
15     gets(adr.phone);
16
```

1. 연락처 관리 툴 Ver 1.0

■ 연락처 추가

```
17 // 입력된 3개의 값을 하나의 문자열로 만든다.  
18 strcat(wstr, adr.name);  
19 strcat(wstr, "\t");  
20 strcat(wstr, adr.age);  
21 strcat(wstr, "\t");  
22 strcat(wstr, adr.phone);  
23 strcat(wstr, "\n");  
24  
25 fputs(wstr, wfp); // 파일에 문자열을 쓴다.  
26  
27 fclose(wfp);  
28 }
```

③

- ② 연락처를 등록하기 위해 우선 구조체 변수(4행에서 선언)에 이름/나이/전화번호를 별도로 입력받음
- ③ 입력된 값들을 모두 하나의 문자열(wstr)로 만든 후 파일에 씀

1. 연락처 관리 툴 Ver 1.0

▪ 연락처 삭제

```
01 // 연락처 파일에서 선택한 연락처를 제거한다.
02 void delete_juso( )
03 {
04     // 연락처 파일의 내용 전체를 저장하기 위한 2차원 배열이다.
05     char read_str[50][200] = { "", } ; // 연락처는 50개로 가정한다. — ①
06
07     char str[200] = "";
08     FILE *rfp, *wfp;
09     int del_line, i, count=0;
10
11     rfp=fopen(fname, "r");
12
13     // 연락처 파일이 없으면 돌아간다.
14     if(rfp == NULL) — ②
15     {
16         puts("\n!! 연락처 파일이 없습니다. !!\n");
17         return;
18     }
19
```

1. 연락처 관리 툴 Ver 1.0

■ 연락처 삭제

```
20  printf("\n삭제할 행 번호는 ? ");
21  scanf("%d", &del_line);
22
23  for(i = 0; i < 50; i++) // 파일에 있는 동안에만 수행한다.
24  {
25      strcpy(str, "");
26      fgets(str, 200, rfp);
27
28      if(i+1 != del_line) // 삭제하는 행이 아니라면 추가한다.
29      {
30          strcpy(read_str[i], str);
31          count ++;
32      }
33      else
34          printf("%d 행이 삭제되었습니다. \n", del_line);
35
36      iffeof(rfp))
37          break;
38  }
39
```

1. 연락처 관리 툴 Ver 1.0

■ 연락처 삭제

```
40  fclose(rfp);
41
42  // 파일을 쓰기 모드로 열고 새로운 내용을 쓴다.
43  wfp=fopen(fname, "w"); ⑤
44
45  for(i = 0; i < count; i++) ⑥
46      fputs(read_str[i], wfp);
47
48  fclose(wfp);
49 }
```

02

사진 편집 툴 Ver 1.0

2. 사진 편집 툴 Ver 1.0

1. 프로그램 개요

- 제약사항

- ① 간단한 8비트(1바이트) 회색(Gray) 이미지로, 512×512 픽셀크기의 raw 파일만을 처리함
- ② 원본 사진 파일은 'C:\temp\bird.raw'로 고정, 변환이 완료되고 사용자가 저장한 사진 파일의 이름은 'result.raw'로 고정

- *.raw

- 512×512 픽셀 크기의 회색(Gray) 이미지
- 가장 단순한 형식의 이미지 파일, 별도의 헤더 정보 없이 각 픽셀이 바로 영상값을 갖는 파일임
- raw 파일을 포토샵(또는 페인트샵)에서 읽으려면 이 raw 파일의 픽셀 크기를 사용자가 미리 알고 있어야 함

2. 사진 편집 툴 Ver 1.0

1. 프로그램 개요

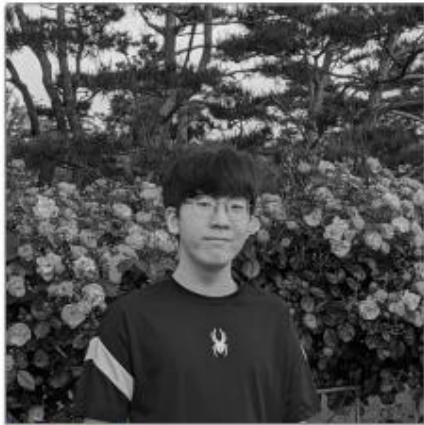


그림 15-3 사진 편집 툴 Ver 1.0 실행 화면

- 1번, 2번, 3번을 각각 누르면 bird.raw 파일에 이미지 처리가 진행됨
- S는 결과를 저장하고 X는 프로그램을 종료
- S로 처리한 결과의 저장은 'C:\Wtemp\Wresult.raw' 파일로 생성됨

2. 사진 편집 툴 Ver 1.0

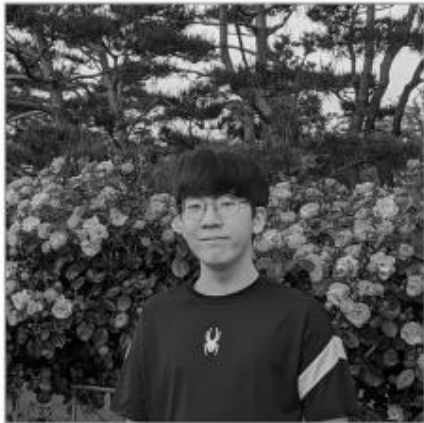
1. 프로그램 개요



원본 이미지



반전된 이미지



좌우 대칭 이미지



상하 대칭 이미지

그림 15-4 사진 편집 툴 Ver 1.0으로 이미지를 처리한 결과

2. 사진 편집 툴 Ver 1.0

2. 프로그램 구현

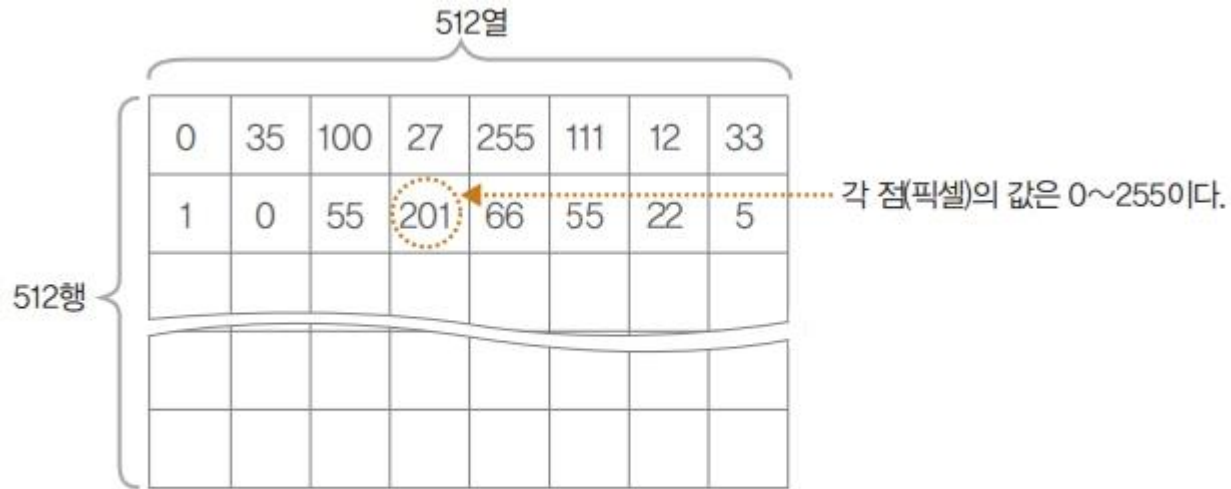


그림 15-5 raw 사진 파일의 구조

- 512×512의 이미지 파일에서 각각의 점(픽셀)은 0~255의 값
- 0에 가까울수록 검정색에 가까운 회색이 되고, 255에 가까울수록 흰색에 가까운 회색이 됨

2. 사진 편집 툴 Ver 1.0

2. 프로그램 구현

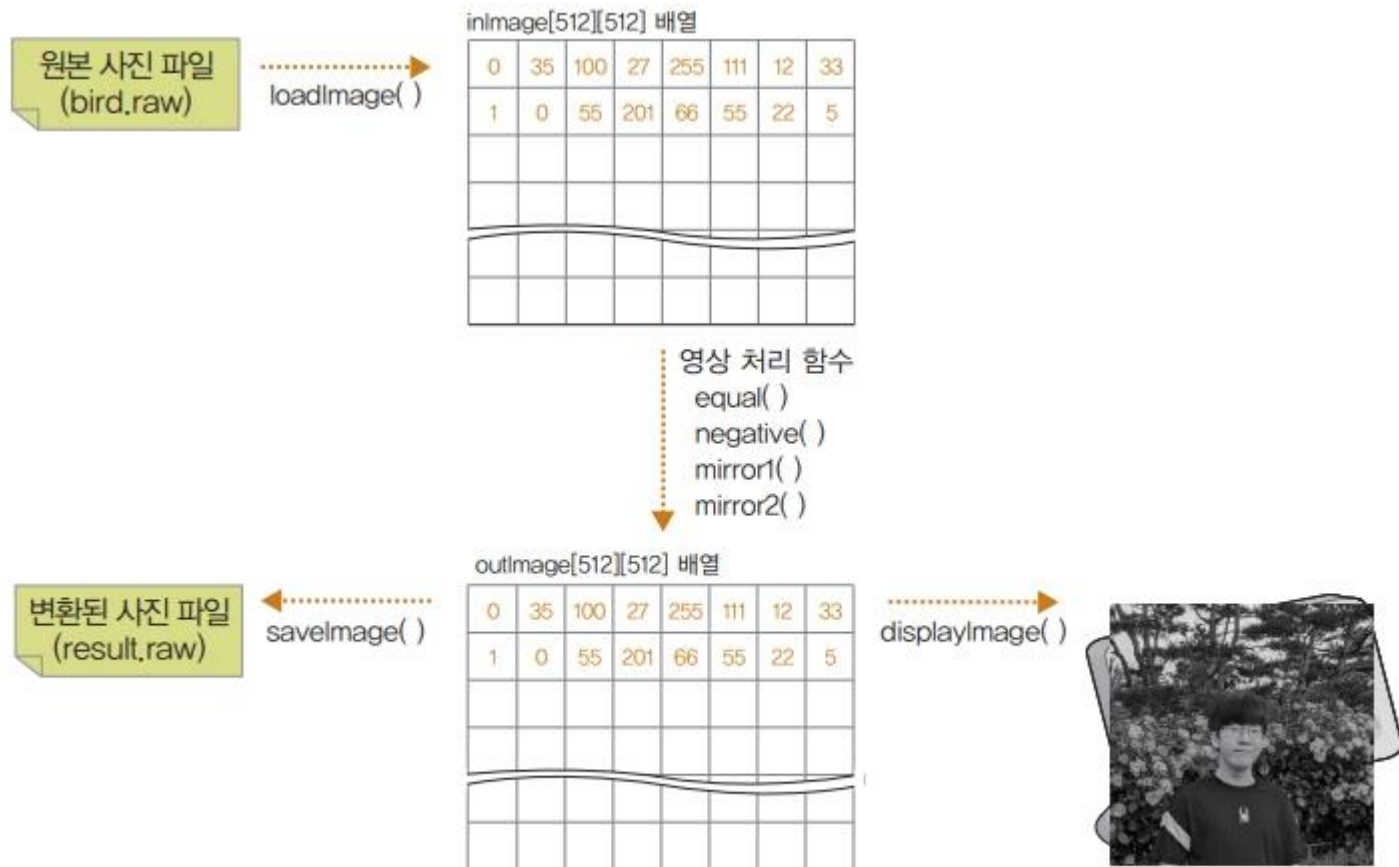


그림 15-6 프로그램의 전체 구조

2. 사진 편집 툴 Ver 1.0

2. 프로그램 구현

- **loadImage() 함수 실행**

- 원본 사진 파일인 bird.raw를 512×512 메모리인 inImage 배열에 불러들임

- **다음의 함수 실행**

- ① equal() : inImage 배열의 내용을 동일하게 outImage에 넣음
- ② negative() : inImage 배열의 값을 '255-원래값'로 만듦. 그러면 색상이 반전된 영상이 됨
- ③ mirror1() : inImage 배열의 위치를 좌우대칭이 되도록 outImage에 넣음
- ④ mirror2() : inImage 배열의 위치를 상하대칭이 되도록 outImage에 넣음

- **displayImage() 함수 실행**

- outImage 배열의 내용을 모니터에 출력해서 사용자가 눈으로 확인할 수 있도록 함

- **'S' 입력**

2. 사진 편집 툴 Ver 1.0

2. 프로그램 구현

- 선언부

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <windows.h>
03 #include <stdio.h>
04
05 void print_menu( );           // 메뉴를 출력한다.
06
07 void equal( );               // 이미지를 동일하게 만드는 함수이다.
08 void negative( );            // 이미지 반전 함수이다.
09 void mirror1( );             // 이미지 좌우 대칭 함수이다.
10 void mirror2( );            // 이미지 상하 대칭 함수이다.
11
12 void loadImage( );           // 입력 영상을 배열에 불러온다.
13 void displayImage( );        // 결과 이미지를 모니터에 출력한다.
14 void saveImage( );           // 결과 이미지를 저장한다.
15
16 // 512x512 raw 이미지 파일이다.
17 const char *fname = "c:\\temp\\bird.raw";
18
19 // 입력과 출력에 필요한 이미지 배열이다.
```

2. 사진 편집 툴 Ver 1.0

2. 프로그램 구현

- 선언부

```
20 unsigned char inImage[512][512], outImage[512][512]; 4
21
22 FILE *rfp, *wfp; 5
23 int i, k;
24
25 // 그래픽 출력을 위한 변수이다.
26 HWND hwnd; 6
27 HDC hdc;
```

2. 사진 편집 툴 Ver 1.0

■ 사진 파일을 배열에 불러오기 : loadImage()

```
01 // 입력 파일을 배열에서 읽는다.  
02 void loadImage( ) {  
03     rfp=fopen(fname, "rb");  
04     fread(inImage, sizeof(char), 512*512, rfp);  
05     fclose(rfp);  
06 }
```

- ① bird.raw 파일을 열 때 읽기 모드인 r과 바이너리 모드인 b를 동시에 사용
- ② fread()는 지정한 크기만큼 데이터를 읽어올 때 사용하는 함수로 형식은 다음과 같음

fread(읽어들일 포인터 또는 배열, 읽어들일 크기, 반복해서 읽을 개수, 파일 포인터);

2. 사진 편집 툴 Ver 1.0

- 영상처리 - 이미지의 동일 처리 : `equal()`
 - 입력 사진 배열(`inImage`)과 출력 사진 배열(`outImage`)를 동일하게 처리



그림 15-7 이미지 동일 처리의 개념

```
01 // 이미지를 동일하게 하는 함수이다.  
02 void equal( )  
03 {  
04     for(i = 0; i < 512; i++)  
05         for(k=0; k < 512; k++)  
06             outImage[i][k] = inImage[i][k];  
07 }
```


2. 사진 편집 툴 Ver 1.0

■ 영상처리 - 이미지의 반전 처리 : negative()

- 이미지를 반전시키기 위해서는 이미지 값(0~255)을 거꾸로(255~0) 만들면 됨

0 ⇨ 255		253 ⇨ 2
1 ⇨ 254	...	254 ⇨ 1
2 ⇨ 253		255 ⇨ 0

- 이를 구현하는 공식은 '반전된 색상 = 255-기존 색상'



그림 15-8 이미지 반전 처리의 개념

```
01 // 이미지를 반전하는 함수이다.
02 void negative( )
03 {
04     for(i = 0; i < 512; i++)
05         for(k=0; k < 512; k++)
06             outImage[i][k] = 255 - inImage[i][k];
07 }
```

2. 사진 편집 툴 Ver 1.0

■ 영상처리 - 이미지의 좌우대칭 처리 : mirror1()

- 좌우대칭 행 = 원래의 행 / 좌우대칭 열 = 전체 크기(5) - 1 = 현재 열

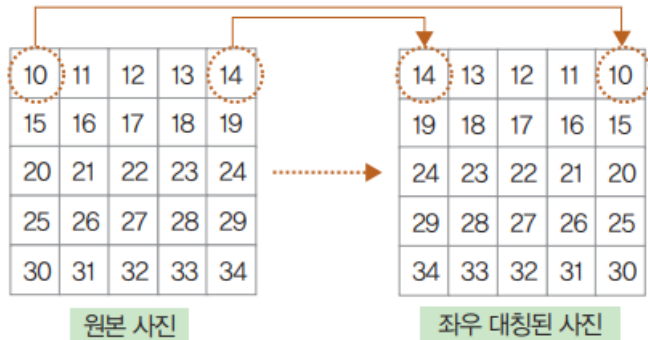


그림 15-9 이미지 좌우 대칭 처리의 개념

- 0행 0열의 위치가 0행 4열의 위치로 변환
- 행은 원래의 행과 동일하고 열만 '전체 열 크기-1-현재 열'로 계산하면 됨

```
01 // 이미지를 좌우 대칭시키는 함수이다.  
02 void mirror1( )  
03 {  
04     for(i=0; i < 512; i++)  
05         for(k=0; k < 512; k++)  
06             outImage[i][k] = inImage[i][511-k];  
07 }
```

2. 사진 편집 툴 Ver 1.0

- 영상처리 - 이미지의 좌우대칭 처리 : `mirror1()`

- ① 이미지를 좌우대칭시키려면 행(i)에 대한 것은 그대로 두고, 열(k)에 대한 것을 '현재 열 크기-1-현재 열'의 위치로 이동시키면 됨
이 이미지의 열 크기는 512이므로 변환될 `outImage`의 k값을 원본 이미지 `orImage`의 '512-1-k'를 사용한 것임

2. 사진 편집 툴 Ver 1.0

- 영상처리 - 이미지의 상하대칭 처리 : mirror2()

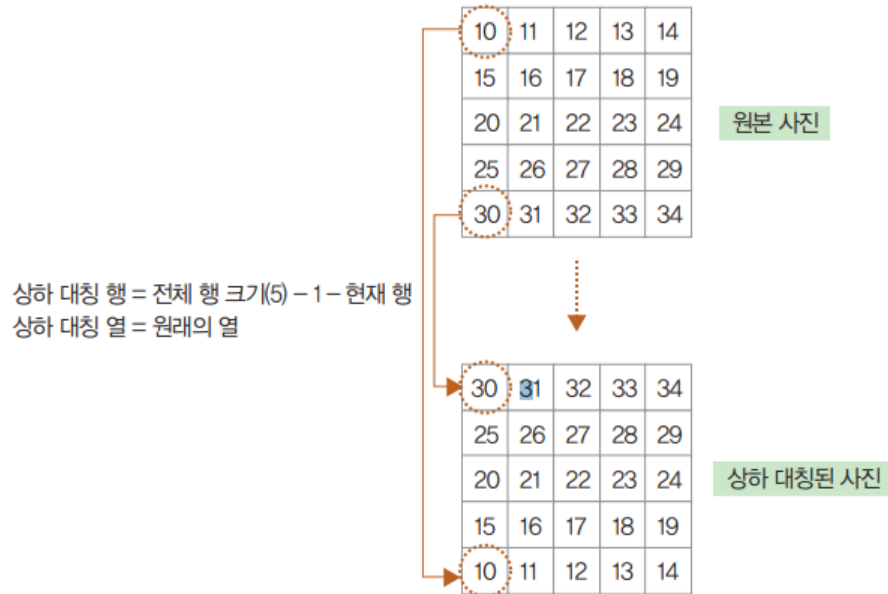


그림 15-10 이미지 상하 대칭 처리의 개념

```
01 // 이미지를 상하 대칭시키는 함수이다.  
02 void mirror2( )  
03 {  
04     for(i=0; i < 512; i++)  
05         for(k=0; k < 512; k++)  
06             outImage[i][k] = inImage[511-i][k];  
07 }
```

2. 사진 편집 툴 Ver 1.0

■ 그래픽 화면 출력 : `displayImage()`

- 출력 배열인 `outImage`의 내용을 모니터에 출력
- 이 함수는 따로 호출하지 않고 4개의 영상처리 함수가 호출된 후 바로 호출됨

```
01 // 출력 배열의 내용을 모니터에 출력한다.  
02 void displayImage( ) {  
03     int value;  
04     system("cls");  
05  
06     for(i=0; i < 512; i++)  
07         for(k=0; k < 512; k++) {  
08             value = outImage[i][k];  
09             SetPixel(hdc, k+50, i+200, RGB(value,value,value));  
10         }  
11 }
```

① 화면을 깨끗하게 지워주는 효과

② `outImage`의 픽셀 하나를 `value`에 대입하고 `SetPixel()` 함수에서 해당 위치에 픽셀 값을 찍음.

`SetPixel(핸들, 모니터 X 위치, 모니터 Y 위치, 색상값);`

감사합니다!

