

- (1) 그레이 스케일 영상을 이진화 처리하는 프로그램을 작성하라
/image/conan.pgm (임계값 100)과 (10점)
/image/horse.pgm (임계값 54와 180)을 이진화 처리하라 (20점).

```

#include <stdio.h>
int main(int argc, char * argv[]) //argc는 외부에서 입력한 문자열 개수,
{
    //argv[]는 외부에서 입력한 문자열들
    int image[600][800]; //800x600 영상까지 처리가능
    int x, y; //반복문에서 사용할 정수 변수 x, y 선언
    char M, N; //문자 변수 M과 N 선언
    int XX, YY, MAX; //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2; //파일 변수 file1과 file2 선언
    file1 = fopen(argv[1], "r"); //외부에서 입력한 두 번째 문자열을 읽기모드로 열
    fscanf(file1, "%c", &M); //헤더의 매직넘버를 읽어 들임 'P'
    fscanf(file1, "%c", &N); //헤더의 매직넘버를 읽어 들임 '2'
    fscanf(file1, "%d", &XX); //입력영상의 가로크기 읽어 들임
    fscanf(file1, "%d", &YY); //입력영상의 세로크기 읽어 들임
    fscanf(file1, "%d", &MAX); //입력영상의 최대 명암도 읽어 들임

    for(y = 0; y < YY; y++) //반복문을 사용하여 입력영상의 픽셀 값 읽어 들임
        for(x = 0; x < XX; x++)
            fscanf(file1, "%d", &image[y][x]);

    for(y = 0; y < YY; y++) //반복문과 if문을 사용하여
        for(x = 0; x < XX; x++) //명암도가 100 이하는 명암도를 0으로
            if(image[y][x] <= 100) //명암도가 100 초과는 명암도를 255로 이진화 함
                image[y][x] = 0;
            else
                image[y][x] = 255;

    file2 = fopen(argv[2], "w"); //외부에서 입력한 세 번째 문자열을 쓰기모드로 열
    fprintf(file2, "%c", M); //읽어 들인 두 번째 문자열 파일의 헤더 매직넘버를
    fprintf(file2, "%cWn", N); //세 번째 문자열 파일에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX); //읽어 들인 최대 명암도를 입력

    for(y = 0; y < YY; y++){
        for(x = 0; x < XX; x++){ //반복문을 이용하여 이진화 시킨 픽셀 값을
            fprintf(file2, "%4d", image[y][x]); //세 번째 문자열 파일에 입력
        }
        fprintf(file2, "Wn");
    }
    return 0;
}

```

What are you doing?

```
[kimhyungho@raspberrypi4 ~]gcc bin.c -o bin
[kimhyungho@raspberrypi4 ~]ls -l bin
-rwx----- 1 kimhyungho class2020 8104  4월 29 21:46 bin*
[kimhyungho@raspberrypi4 ~]xv /image/conan.pgm &
[1] 15926
[kimhyungho@raspberrypi4 ~]bin /image/conan.pgm out.pgm
[kimhyungho@raspberrypi4 ~]xv out.pgm &
[2] 15932
[kimhyungho@raspberrypi4 ~]
```

xv 3.10a: /image/conan.pgm <Jung, Minchul>

xv 3.10a: out.pgm <Jung, Minchul>

fscnaf로 읽기모드로 연 입력영상 /image/conan.pgm의 헤더와 데이터 포맷 부분을 읽어 들였습니다. 읽어 들인 데이터 포맷을 반복문과 조건문을 사용하여 임계값 100을 기준으로 명암도가 100 이하인 부분은 명암도를 0으로, 명암도가 100 초과인 부분은 명암도를 255로 바꿔 영상을 이진화 시켰습니다. 입력영상에서 읽어 들인 영상의 헤더 부분과 이진화 시킨 영상의 데이터 포맷을 쓰기모드로 연 out.pgm에 fprintf를 사용하여 입력했습니다. 이 후 실행과정에서 -o 옵션을 사용하여 컴파일 한 bin.c의 실행파일을 a.out 대신 bin으로 만들어주었습니다. 그리고 ls -l bin을 사용하여 생성 된 실행파일을 확인하였고 xv와 후면작업 &을 사용하여 입력영상 /image/conan.pgm과 bin으로 얻어 낸 출력영상 out.pgm을 동시에 출력하였습니다.

```

#include <stdio.h>
int main(int argc, char * argv[]) //argc는 외부에서 입력한 문자열 개수,
{
    //argv[]는 외부에서 입력한 문자열들
    int image[600][800]; //800x600 영상까지 처리가능
    int x, y; //반복문에서 사용할 정수 변수 x, y 선언
    char M, N; //문자 변수 M과 N 선언
    int XX, YY, MAX; //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2; //파일 변수 file1과 file2 선언
    file1 = fopen(argv[1], "r"); //외부에서 입력한 두 번째 문자열을 읽기모드로 열
    fscanf(file1, "%c", &M); //헤더의 매직넘버를 읽어 들임 'P'
    fscanf(file1, "%c", &N); //헤더의 매직넘버를 읽어 들임 '2'
    fscanf(file1, "%d", &XX); //입력영상의 가로크기 읽어 들임
    fscanf(file1, "%d", &YY); //입력영상의 세로크기 읽어 들임
    fscanf(file1, "%d", &MAX); //입력영상의 최대 명암도 읽어 들임

    for(y = 0; y < YY; y++) //반복문을 사용하여 입력영상의 픽셀 값 읽어 들임
        for(x = 0; x < XX; x++)
            fscanf(file1, "%d", &image[y][x]);

    for(y = 0; y < YY; y++) //반복문과 if문을 사용하여
        for(x = 0; x < XX; x++) //명암도가 54 이하는 명암도를 0으로
            if(image[y][x] <= 54) //명암도가 54 초과는 명암도를 255로 이진화 함
                image[y][x] = 0;
            else
                image[y][x] = 255;

    file2 = fopen(argv[2], "w"); //외부에서 입력한 세 번째 문자열을 쓰기모드로 열
    fprintf(file2, "%c", M); //읽어 들인 두 번째 문자열 파일의 헤더 매직넘버를
    fprintf(file2, "%cWn", N); //세 번째 문자열 파일에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX); //읽어 들인 최대 명암도를 입력

    for(y = 0; y < YY; y++){
        for(x = 0; x < XX; x++){ //반복문을 이용하여 이진화 시킨 픽셀 값을
            fprintf(file2, "%4d", image[y][x]); //세 번째 문자열 파일에 입력
        }
        fprintf(file2, "Wn");
    }
    return 0;
}

```

What are you doing?

```
[kimhyungho@raspberrypi4 ~]gcc bin.c -o bin
[kimhyungho@raspberrypi4 ~]xv /image/horse.pgm &
[1] 15992
[kimhyungho@raspberrypi4 ~]bin /image/horse.pgm out1.pgm
[kimhyungho@raspberrypi4 ~]xv out1.pgm &
[2] 15996
[kimhyungho@raspberrypi4 ~]
```

xv 3.10a: /image/horse.pgm <Jung, Minchul>

xv 3.10a: out2.pgm <Jung, Minchul>

앞에서 수행한 (1)번 과제에서 임계값만 바꿔 이진화 시켰습니다. fscnaf로 읽기모드로 연 입력영상 /image/horse.pgm의 헤더와 데이터 포맷 부분을 읽어 들였습니다. 읽어 들인 데이터 포맷을 반복문과 조건문을 사용하여 임계값 54를 기준으로 명암도가 54 이하인 부분은 명암도를 0으로, 명암도가 54 초과인 부분은 명암도를 255로 바꿔 영상을 이진화 시켰습니다. 입력영상에서 읽어 들인 영상의 헤더 부분과 이진화 시킨 영상의 데이터 포맷을 쓰기모드로 연 out1.pgm에 fprintf를 사용하여 입력했습니다. 이 후 실행과정에서 -o 옵션을 사용하여 컴파일 한 bin.c의 실행파일을 a.out 대신 bin으로 만들어주었습니다. xv와 후면작업 &을 사용하여 입력영상 /image/horse.pgm과 bin으로 얻어낸 출력영상 out1.pgm을 동시에 출력하였습니다.



```

#include <stdio.h>
int main(int argc, char * argv[]) //argc는 외부에서 입력한 문자열 개수,
{
    //argv[]는 외부에서 입력한 문자열들
    int image[600][800]; //800x600 영상까지 처리가능
    int x, y; //반복문에서 사용할 정수 변수 x, y 선언
    char M, N; //문자 변수 M과 N 선언
    int XX, YY, MAX; //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2; //파일 변수 file1과 file2 선언
    file1 = fopen(argv[1], "r"); //외부에서 입력한 두 번째 문자열을 읽기모드로 열
    fscanf(file1, "%c", &M); //헤더의 매직넘버를 읽어 들임 'P'
    fscanf(file1, "%c", &N); //헤더의 매직넘버를 읽어 들임 '2'
    fscanf(file1, "%d", &XX); //입력영상의 가로크기 읽어 들임
    fscanf(file1, "%d", &YY); //입력영상의 세로크기 읽어 들임
    fscanf(file1, "%d", &MAX); //입력영상의 최대 명암도 읽어 들임


    for(y = 0; y < YY; y++) //반복문을 사용하여 입력영상의 픽셀 값 읽어 들임
        for(x = 0; x < XX; x++)
            fscanf(file1, "%d", &image[y][x]);

    for(y = 0; y < YY; y++) //반복문과 if문을 사용하여
        for(x = 0; x < XX; x++) //명암도가 180 이하는 명암도를 0으로
            if(image[y][x] <= 180) //명암도가 180 초과는 명암도를 255로 이진화 함
                image[y][x] = 0;
            else
                image[y][x] = 255;

    file2 = fopen(argv[2], "w"); //외부에서 입력한 세 번째 문자열을 쓰기모드로 열
    fprintf(file2, "%c", M); //읽어 들인 두 번째 문자열 파일의 헤더 매직넘버를
    fprintf(file2, "%cWn", N); //세 번째 문자열 파일에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX); //읽어 들인 최대 명암도를 입력


    for(y = 0; y < YY; y++){
        for(x = 0; x < XX; x++){ //반복문을 이용하여 이진화 시킨 픽셀 값을
            fprintf(file2, "%4d", image[y][x]); //세 번째 문자열 파일에 입력
        }
        fprintf(file2, "Wn");
    }
    return 0;
}

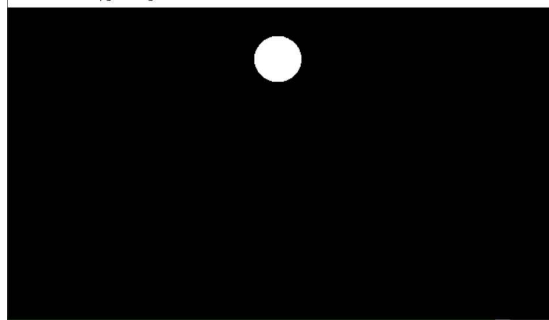
```

 What are you doing?

```

[kimhyungho@raspberrry4 ~]gcc bin.c -o bin
[kimhyungho@raspberrry4 ~]bin /image/horse.pgm out2.pgm
[kimhyungho@raspberrry4 ~]xv out2.pgm &
[3] 16008
[kimhyungho@raspberrry4 ~]
    
```





앞에서 수행한 (1)번과 (2)번 과제에서 임계값만 바꿔 이진화 시켰습니다. fscnaf로 읽기
 모드로 연 입력영상 /image/horse.pgm의 헤더와 데이터 포맷 부분을 읽어 들였습니다.
 읽어 들인 데이터 포맷을 반복문과 조건문을 사용하여 임계값 180를 기준으로 명암도가
 180 이하인 부분은 명암도를 0으로, 명암도가 180 초과인 부분은 명암도를 255로 바꿔
 영상을 이진화 시켰습니다. 입력영상에서 읽어 들인 영상의 헤더 부분과 이진화 시킨 영
 상의 데이터 포맷을 쓰기모드로 연 out2.pgm에 fprintf를 사용하여 입력했습니다. 이 후
 실행과정에서 -o 옵션을 사용하여 컴파일 한 bin.c의 실행파일을 a.out 대신 bin으로 만
 들어주었습니다. xv와 후면작업 &을 사용하여 입력영상 /image/horse.pgm과 (2)번에서
 수행한 임계값 54를 갖는 이진화 영상과 동시에 bin으로 얻어낸 출력영상 out2.pgm을
 출력하였습니다.

- (2) 컬러 영상을 채널별로 분리하는 프로그램을 작성하라.
- /image/rgb.ppm을 Red, Green, Blue로 각각 분리하라 (20점).
- /image/needhelp.ppm을 Red, Green, Blue로 각각 분리하라 (20점).

```

#include <stdio.h>
int main(int argc, char * argv[]) //argc는 외부에서 입력한 문자열 개수,
{
    //argv[]는 외부에서 입력한 문자열들
    int image1[600][800*3]; //800x600 영상까지 처리가능
    int image2[600][800*3]; //800x600 영상까지 처리가능
    int x, y; //반복문에서 사용할 정수 변수 x, y 선언
    char M, N; //문자 변수 M과 N 선언
    int XX, YY, MAX; //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2; //파일 변수 file1과 file2 선언
    file1 = fopen(argv[1], "r"); //외부에서 입력한 두 번째 문자열을 읽기모드로 열
    fscanf(file1, "%c", &M); //헤더의 매직넘버를 읽어 들임 'P'
    fscanf(file1, "%c", &N); //헤더의 매직넘버를 읽어 들임 '3'
    fscanf(file1, "%d", &XX); //입력영상의 가로크기 읽어 들임
    fscanf(file1, "%d", &YY); //입력영상의 세로크기 읽어 들임
    fscanf(file1, "%d", &MAX); //입력영상의 최대 명암도 읽어 들임


    for(y = 0; y < YY; y++) //반복문을 사용하여 입력영상의 픽셀 값 읽어 들임
        for(x = 0; x < XX*3; x++)
            fscanf(file1, "%d", &image1[y][x]);

    for(y = 0; y < YY; y++) //반복문을 사용하여 Red 분리
        for(x = 0; x < XX*3; x+=3) //R-Channel
            image2[y][x] = image1[y][x];

    file2 = fopen(argv[2], "w"); //외부에서 입력한 세 번째 문자열을 쓰기모드로 열
    fprintf(file2, "%c", M); //읽어 들인 두 번째 문자열 파일의 매직넘버를
    fprintf(file2, "%cWn", N); //세 번째 문자열 파일에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX); //읽어 들인 최대 명암도를 입력

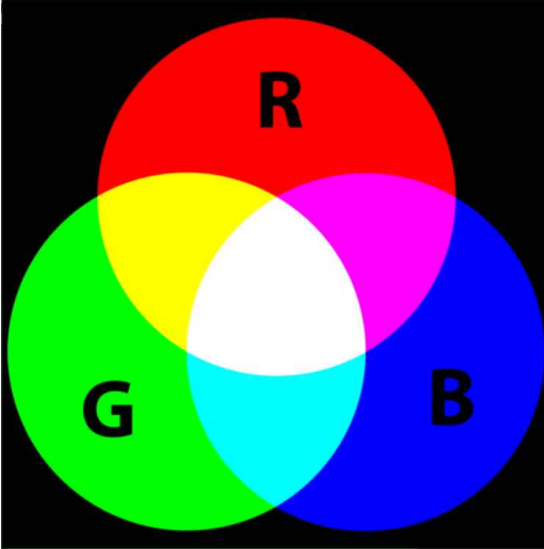
    for(y = 0; y < YY; y++){
        for(x = 0; x < XX*3; x++){ //컬러영상은 픽셀 당 값 개를 가지므로 *3
            fprintf(file2, "%4d", image2[y][x]); //반복문을 이용하여 Red로 분리한
        } //픽셀 값을 세 번째 문자열 파일에 입력
        fprintf(file2, "Wn");
    }
    return 0;
}

```

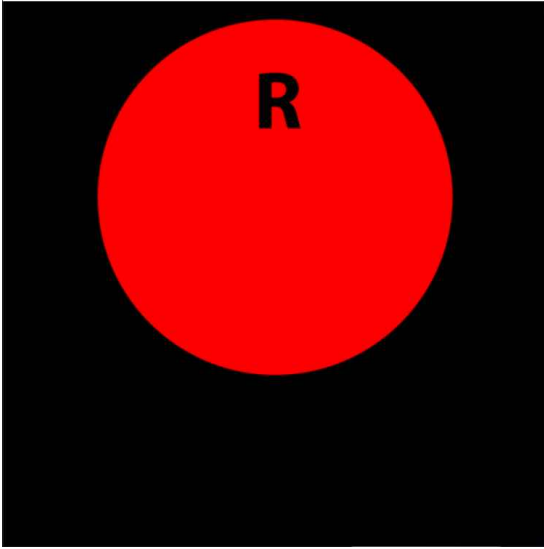

 What are you doing?

```
[kimhyungho@raspberrypi4 ~]gcc R.c -o R
[kimhyungho@raspberrypi4 ~]xv /image/rgb.ppm &
[1] 18224
[kimhyungho@raspberrypi4 ~]unlimit
[kimhyungho@raspberrypi4 ~]R /image/rgb.ppm out1.ppm
[kimhyungho@raspberrypi4 ~]xv out1.ppm &
[2] 18227
[kimhyungho@raspberrypi4 ~]
```

xv 3.10a: /image/rgb.ppm <Jung, Minchul>



xv 3.10a: out.ppm <Jung, Minchul>



fscanf로 읽기모드로 연 입력영상 /image/rgb.ppm의 헤더와 데이터 포맷 부분을 읽어 들였습니다. 읽어 들인 데이터 포맷을 반복문을 사용하여 붉은색 부분을 분리했습니다. 붉은색은 x좌표 0, 3, 6, 9,... 에 존재하기 때문에 반복문으로 x는 0부터 3씩 증가시켜 image1배열의 R부분 픽셀 값을 image2배열에 넣었습니다. 입력영상에서 읽어 들인 영상의 헤더 부분과 추출한 붉은색 데이터 포맷을 쓰기모드로 연 out1.pgm에 fprintf를 사용하여 입력했습니다. 이 후 실행과정에서 -o 옵션을 사용하여 컴파일 한 R.c의 실행파일을 a.out 대신 R로 만들어주었고 unlimit을 이용하여 메모리를 늘려주었습니다. 그리고 xv와 후면작업 &을 사용하여 입력영상 /image/rpg.ppm과 R로 얻어낸 출력영상 out1.pgm을 동시에 출력하였습니다.

```

#include <stdio.h>
int main(int argc, char * argv[]) //argc는 외부에서 입력한 문자열 개수,
{
    //argv[]는 외부에서 입력한 문자열들
    int image1[600][800*3]; //800x600 영상까지 처리가능
    int image2[600][800*3]; //800x600 영상까지 처리가능
    int x, y; //반복문에서 사용할 정수 변수 x, y 선언
    char M, N; //문자 변수 M과 N 선언
    int XX, YY, MAX; //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2; //파일 변수 file1과 file2 선언
    file1 = fopen(argv[1], "r"); //외부에서 입력한 두 번째 문자열을 읽기모드로 열
    fscanf(file1, "%c", &M); //헤더의 매직넘버를 읽어 들임 'P'
    fscanf(file1, "%c", &N); //헤더의 매직넘버를 읽어 들임 '3'
    fscanf(file1, "%d", &XX); //입력영상의 가로크기 읽어 들임
    fscanf(file1, "%d", &YY); //입력영상의 세로크기 읽어 들임
    fscanf(file1, "%d", &MAX); //입력영상의 최대 명암도 읽어 들임

    for(y = 0; y < YY; y++) //반복문을 사용하여 입력영상의 픽셀 값 읽어 들임
        for(x = 0; x < XX*3; x++)
            fscanf(file1, "%d", &image1[y][x]);

    for(y = 0; y < YY; y++) //반복문을 사용하여 Green 분리
        for(x = 1; x < XX*3; x+=3) //G-Channel
            image2[y][x] = image1[y][x];

    file2 = fopen(argv[2], "w"); //외부에서 입력한 세 번째 문자열을 쓰기모드로 열
    fprintf(file2, "%c", M); //읽어 들인 두 번째 문자열 파일의 매직넘버를
    fprintf(file2, "%cWn", N); //세 번째 문자열 파일에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX); //읽어 들인 최대 명암도를 입력

    for(y = 0; y < YY; y++){
        for(x = 0; x < XX*3; x++){ //컬러영상은 픽셀 당 값 개를 가지므로 *3
            fprintf(file2, "%4d", image2[y][x]); //반복문을 이용하여 Green으로 분리한
        } //픽셀 값을 세 번째 문자열 파일에 입력
        fprintf(file2, "Wn");
    }
    return 0;
}

```

What are you doing?

```
[kimhyungho@raspberrypi4 ~] gcc G.c -o G
[kimhyungho@raspberrypi4 ~] G /image/rgb.ppm out2.ppm
[kimhyungho@raspberrypi4 ~] xv out2.ppm &
[3] 18240
[kimhyungho@raspberrypi4 ~]
```

xv 3.10a: /image/rgb.ppm <Jung, Minchul>

xv 3.10a: out2.ppm <Jung, Minchul>

fscanf로 읽기모드로 연 입력영상 /image/rgb.ppm의 헤더와 데이터 포맷 부분을 읽어 들였습니다. 읽어 들인 데이터 포맷을 반복문을 사용하여 녹색 부분을 분리했습니다. 녹색은 x좌표 1, 4, 7, 10,... 에 존재하기 때문에 반복문으로 x는 1부터 3씩 증가시켜 image1배열의 G부분 픽셀 값을 image2배열에 넣었습니다. 입력영상에서 읽어 들인 영상의 헤더 부분과 추출한 초록색 데이터 포맷을 쓰기모드로 연 out2.pgm에 fprintf를 사용하여 입력했습니다. 이 후 실행과정에서 -o 옵션을 사용하여 컴파일 한 G.c의 실행파일을 a.out 대신 G로 만들어주었고 앞에서 이미 앞 과제에서 unlimit을 이용하여 메모리를 늘려주었기 때문에 따로 unlimit을 사용하지는 않았습니다. 그리고 xv와 후면작업 &을 사용하여 입력영상 /image/rpg.ppm, R로 얻어낸 출력영상 out1.pgm과 G로 얻어낸 출력영상 out2.pgm을 동시에 출력하였습니다.

```

#include <stdio.h>
int main(int argc, char * argv[]) //argc는 외부에서 입력한 문자열 개수,
{
    //argv[]는 외부에서 입력한 문자열들
    int image1[600][800*3]; //800x600 영상까지 처리가능
    int image2[600][800*3]; //800x600 영상까지 처리가능
    int x, y; //반복문에서 사용할 정수 변수 x, y 선언
    char M, N; //문자 변수 M과 N 선언
    int XX, YY, MAX; //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2; //파일 변수 file1과 file2 선언
    file1 = fopen(argv[1], "r"); //외부에서 입력한 두 번째 문자열을 읽기모드로 열
    fscanf(file1, "%c", &M); //헤더의 매직넘버를 읽어 들임 'P'
    fscanf(file1, "%c", &N); //헤더의 매직넘버를 읽어 들임 '3'
    fscanf(file1, "%d", &XX); //입력영상의 가로크기 읽어 들임
    fscanf(file1, "%d", &YY); //입력영상의 세로크기 읽어 들임
    fscanf(file1, "%d", &MAX); //입력영상의 최대 명암도 읽어 들임


    for(y = 0; y < YY; y++) //반복문을 사용하여 입력영상의 픽셀 값 읽어 들임
        for(x = 0; x < XX*3; x++)
            fscanf(file1, "%d", &image1[y][x]);

    for(y = 0; y < YY; y++) //반복문을 사용하여 Blue 분리
        for(x = 2; x < XX*3; x+=3) //B-Channel
            image2[y][x] = image1[y][x];

    file2 = fopen(argv[2], "w"); //외부에서 입력한 세 번째 문자열을 쓰기모드로 열
    fprintf(file2, "%c", M); //읽어 들인 두 번째 문자열 파일의 매직넘버를
    fprintf(file2, "%cWn", N); //세 번째 문자열 파일에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX); //읽어 들인 최대 명암도를 입력

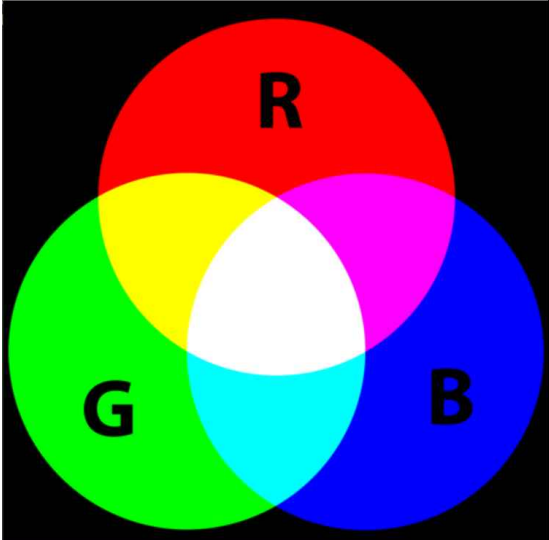
    for(y = 0; y < YY; y++){
        for(x = 0; x < XX*3; x++){ //컬러영상은 픽셀 당 값 개를 가지므로 *3
            fprintf(file2, "%4d", image2[y][x]); //반복문을 이용하여 Blue로 분리한
        } //픽셀 값을 세 번째 문자열 파일에 입력
        fprintf(file2, "Wn");
    }
    return 0;
}

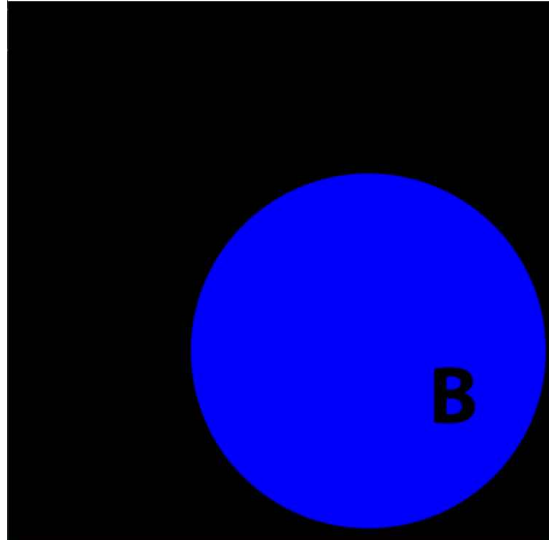
```

 What are you doing?

```

[kimhyungho@raspberrypi4 ~]gcc B.c -o B
[kimhyungho@raspberrypi4 ~]B /image/rgb.ppm out3.ppm
[kimhyungho@raspberrypi4 ~]xv out3.ppm &
[4] 18250
[kimhyungho@raspberrypi4 ~]
    
```





fscanf로 읽기모드로 연 입력영상 /image/rgb.ppm의 헤더와 데이터 포맷 부분을 읽어 들였습니다. 읽어 들인 데이터 포맷을 반복문을 사용하여 파란색 부분을 분리했습니다. 파란색은 x좌표 2, 5, 8, 11,... 에 존재하기 때문에 반복문으로 x는 2부터 3씩 증가시켜 image1배열의 B부분 픽셀 값을 image2배열에 넣었습니다. 입력영상에서 읽어 들인 영상의 헤더 부분과 추출한 파란색 데이터 포맷을 쓰기모드로 연 out3.pgm에 fprintf를 사용하여 입력했습니다. 이 후 실행과정에서 -o 옵션을 사용하여 컴파일 한 B.c의 실행파일을 a.out 대신 B로 만들어주었고 앞에서 이미 앞 과제에서 unlimited을 이용하여 메모리를 늘려주었기 때문에 따로 unlimited을 사용하지는 않았습니다. 그리고 xv와 후면작업 & 을 사용하여 입력영상 /image/rpg.ppm, R로 얻어낸 출력영상 out1.pgm, G로 얻어낸 출력영상 out2.pgm과 B로 얻어낸 출력영상 out3.pgm을 동시에 출력했습니다.

```

#include <stdio.h>
int main(int argc, char * argv[]) //argc는 외부에서 입력한 문자열 개수,
{
    //argv[]는 외부에서 입력한 문자열들
    int image1[600][800*3]; //800x600 영상까지 처리가능
    int image2[600][800*3]; //800x600 영상까지 처리가능
    int x, y; //반복문에서 사용할 정수 변수 x, y 선언
    char M, N; //문자 변수 M과 N 선언
    int XX, YY, MAX; //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2; //파일 변수 file1과 file2 선언
    file1 = fopen(argv[1], "r"); //외부에서 입력한 두 번째 문자열을 읽기모드로 열
    fscanf(file1, "%c", &M); //헤더의 매직넘버를 읽어 들임 'P'
    fscanf(file1, "%c", &N); //헤더의 매직넘버를 읽어 들임 '3'
    fscanf(file1, "%d", &XX); //입력영상의 가로크기 읽어 들임
    fscanf(file1, "%d", &YY); //입력영상의 세로크기 읽어 들임
    fscanf(file1, "%d", &MAX); //입력영상의 최대 명암도 읽어 들임


    for(y = 0; y < YY; y++) //반복문을 사용하여 입력영상의 픽셀 값 읽어 들임
        for(x = 0; x < XX*3; x++)
            fscanf(file1, "%d", &image1[y][x]);

    for(y = 0; y < YY; y++) //반복문을 사용하여 Red 분리
        for(x = 0; x < XX*3; x+=3) //R-Channel
            image2[y][x] = image1[y][x];

    file2 = fopen(argv[2], "w"); //외부에서 입력한 세 번째 문자열을 쓰기모드로 열
    fprintf(file2, "%c", M); //읽어 들인 두 번째 문자열 파일의 매직넘버를
    fprintf(file2, "%cWn", N); //세 번째 문자열 파일에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX); //읽어 들인 최대 명암도를 입력


    for(y = 0; y < YY; y++){
        for(x = 0; x < XX*3; x++){ //컬러영상은 픽셀 당 값 개를 가지므로 *3
            fprintf(file2, "%4d", image2[y][x]); //반복문을 이용하여 Red로 분리한
        } //픽셀 값을 세 번째 문자열 파일에 입력
        fprintf(file2, "Wn");
    }
    return 0;
}

```


 What are you doing?

```
[kimhyungho@raspberrypi4 ~]xv /image/needhelp.ppm &
[1] 18290
[kimhyungho@raspberrypi4 ~]R /image/needhelp.ppm out1.ppm
[kimhyungho@raspberrypi4 ~]xv out1.ppm &
[2] 18294
[kimhyungho@raspberrypi4 ~]
```

xv 3.10a: /image/needhelp.ppm <Jung, Minchul>



xv 3.10a: out1.ppm <Jung, Minchul>



앞에서 수행한 /image/rgb.ppm을 Red로 분리한 코드를 그대로 사용하였습니다. 그리고 이미 실행파일 R을 만들어 냈기 때문에 따로 컴파일도 하지 않고 입력영상만 바뀌서 실행했습니다. 입력영상 /image/needhelp.ppm을 넣어 출력영상 out1.ppm을 출력했습니다. 그리고 xv와 후면작업 &을 이용하여 입력영상과 동시에 out1.ppm을 출력했습니다.



```

#include <stdio.h>
int main(int argc, char * argv[]) //argc는 외부에서 입력한 문자열 개수,
{
    //argv[]는 외부에서 입력한 문자열들
    int image1[600][800*3]; //800x600 영상까지 처리가능
    int image2[600][800*3]; //800x600 영상까지 처리가능
    int x, y; //반복문에서 사용할 정수 변수 x, y 선언
    char M, N; //문자 변수 M과 N 선언
    int XX, YY, MAX; //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2; //파일 변수 file1과 file2 선언
    file1 = fopen(argv[1], "r"); //외부에서 입력한 두 번째 문자열을 읽기모드로 열
    fscanf(file1, "%c", &M); //헤더의 매직넘버를 읽어 들임 'P'
    fscanf(file1, "%c", &N); //헤더의 매직넘버를 읽어 들임 '3'
    fscanf(file1, "%d", &XX); //입력영상의 가로크기 읽어 들임
    fscanf(file1, "%d", &YY); //입력영상의 세로크기 읽어 들임
    fscanf(file1, "%d", &MAX); //입력영상의 최대 명암도 읽어 들임


    for(y = 0; y < YY; y++) //반복문을 사용하여 입력영상의 픽셀 값 읽어 들임
        for(x = 0; x < XX*3; x++)
            fscanf(file1, "%d", &image1[y][x]);

    for(y = 0; y < YY; y++) //반복문을 사용하여 Green 분리
        for(x = 0; x < XX*3; x+=3) //G-Channel
            image2[y][x] = image1[y][x];

    file2 = fopen(argv[2], "w"); //외부에서 입력한 세 번째 문자열을 쓰기모드로 열
    fprintf(file2, "%c", M); //읽어 들인 두 번째 문자열 파일의 매직넘버를
    fprintf(file2, "%cWn", N); //세 번째 문자열 파일에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX); //읽어 들인 최대 명암도를 입력


    for(y = 0; y < YY; y++){
        for(x = 0; x < XX*3; x++){ //컬러영상은 픽셀 당 값 개를 가지므로 *3
            fprintf(file2, "%4d", image2[y][x]); //반복문을 이용하여 Green으로 분리한
        } //픽셀 값을 세 번째 문자열 파일에 입력
        fprintf(file2, "Wn");
    }
    return 0;
}

```



 What are you doing?

```
[kimhyungho@raspberrypi4 ~]G /image/needhelp.ppm out2.ppm
[kimhyungho@raspberrypi4 ~]xv out2.ppm &
[3] 18297
[kimhyungho@raspberrypi4 ~]
```

xv 3.10a: /image/needhelp.ppm <Jung, Minchul>



xv 3.10a: out2.ppm <Jung, Minchul>



앞에서 수행한 /image/rgb.ppm을 Green으로 분리한 코드를 그대로 사용하였습니다. 그리고 이미 실행파일 G를 만들어 냈기 때문에 따로 컴파일도 하지 않고 입력영상만 바뀌서 실행했습니다. 입력영상 /image/needhelp.ppm을 넣어 출력영상 out2.ppm을 출력했습니다. 그리고 xv와 후면작업 &을 이용하여 입력영상, out1.ppm과 out2.ppm을 동시에 출력했습니다.



```

#include <stdio.h>
int main(int argc, char * argv[]) //argc는 외부에서 입력한 문자열 개수,
{
    //argv[]는 외부에서 입력한 문자열들
    int image1[600][800*3]; //800x600 영상까지 처리가능
    int image2[600][800*3]; //800x600 영상까지 처리가능
    int x, y; //반복문에서 사용할 정수 변수 x, y 선언
    char M, N; //문자 변수 M과 N 선언
    int XX, YY, MAX; //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2; //파일 변수 file1과 file2 선언
    file1 = fopen(argv[1], "r"); //외부에서 입력한 두 번째 문자열을 읽기모드로 열
    fscanf(file1, "%c", &M); //헤더의 매직넘버를 읽어 들임 'P'
    fscanf(file1, "%c", &N); //헤더의 매직넘버를 읽어 들임 '3'
    fscanf(file1, "%d", &XX); //입력영상의 가로크기 읽어 들임
    fscanf(file1, "%d", &YY); //입력영상의 세로크기 읽어 들임
    fscanf(file1, "%d", &MAX); //입력영상의 최대 명암도 읽어 들임

    for(y = 0; y < YY; y++) //반복문을 사용하여 입력영상의 픽셀 값 읽어 들임
        for(x = 0; x < XX*3; x++)
            fscanf(file1, "%d", &image1[y][x]);

    for(y = 0; y < YY; y++) //반복문을 사용하여 Blue 분리
        for(x = 2; x < XX*3; x+=3) //B-Channel
            image2[y][x] = image1[y][x];

    file2 = fopen(argv[2], "w"); //외부에서 입력한 세 번째 문자열을 쓰기모드로 열
    fprintf(file2, "%c", M); //읽어 들인 두 번째 문자열 파일의 매직넘버를
    fprintf(file2, "%cWn", N); //세 번째 문자열 파일에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX); //읽어 들인 최대 명암도를 입력

    for(y = 0; y < YY; y++){
        for(x = 0; x < XX*3; x++){ //컬러영상은 픽셀 당 값 개를 가지므로 *3
            fprintf(file2, "%4d", image2[y][x]); //반복문을 이용하여 Blue로 분리한
        } //픽셀 값을 세 번째 문자열 파일에 입력
        fprintf(file2, "Wn");
    }
    return 0;
}

```

What are you doing?

```
[kimhyungho@raspberrry4 ~]B /image/needhelp.ppm out3.ppm
[kimhyungho@raspberrry4 ~]xv out3.ppm &
[4] 18322
[kimhyungho@raspberrry4 ~]
```

xv 3.10a: /image/needhelp.ppm <Jung, Minchul>

xv 3.10a: out3.ppm <Jung, Minchul>

앞에서 수행한 /image/rgb.ppm을 Blue로 분리한 코드를 그대로 사용하였습니다. 그리고 이미 실행파일 B를 만들어 냈기 때문에 따로 컴파일도 하지 않고 입력영상만 바뀌서 실행했습니다. 입력영상 /image/needhelp.ppm을 넣어 출력영상 out3.ppm을 출력했습니다. 그리고 xv와 후면작업 &을 이용하여 입력영상, out1.ppm과 out2.ppm과 동시에 out3.ppm을 출력했습니다.



☞ 워드파일(hwp/doc)로 작성하는 보고서 파일에는

- ① 각각의 줄을 주석 처리한 소스(화면 캡처금지, 텍스트로 작성),
- ② 리눅스 명령 실행 화면, ③ 입력 영상 ④ 출력 영상이 반드시 있어야 하며 과제는 상대 평가 됩니다.