

응용5번

---헤더파일---

mystruct.h

```
#ifndef __MYSTRUCT_H__
#define __MYSTRUCT_H__

#define NAME_SIZE 16

typedef struct score
{
    int id;
    char name[NAME_SIZE];
    int korean;
    int english;
    int math;
    int all;
    float avg;
}score;

#endif
```

구조체 이름 score
id : 학번
name : 이름
korean : 국어점수
english : 영어점수
math : 수학점수
all : 총점수
avg : 평균
구조체 별칭 score

input.h

```
#ifndef __INPUT_H__
#define __INPUT_H__

void input(char *file);

#endif
```

output.h

```
#ifndef __OUTPUT_H__
#define __OUTPUT_H__

void output(char *file);

#endif
```

append.h

```
#ifndef __APPEND_H__
#define __APPEND_H__

void append(char *file, int id, char *name, int korean, int english, int math);

#endif
```

process.h

```
#ifndef __process_H__
#define __process_H__

void process(char *file);

#endif
```

search.h

```
#ifndef __SEARCH_H__
#define __SEARCH_H__

void search(char *file, int search_opt, char *optarg);

#endif
```

함수파일

input.c

```
#include<stdio.h>
#include<string.h>
#include"mystruct.h"

void input(char *file)
{
    score stu;
    printf("성적 입력 프로그램입니다.\n");
    FILE *fp;
    fp=fopen(file,"wb+");
    if(fp==NULL)
    {
        perror("파일을 여는데 오류가 발생하였습니다.");
        exit(1);
    }
    while(1)
    {
        memset(&stu,0,sizeof(stu));
        printf("학번을 입력하세요 (종료 0) : ");
        scanf("%d",&stu.id);
        if(stu.id==0)
        {
            printf("0을 입력하여 종료합니다.\n");
            break;
        }
        __fpurge(stdin);
        printf("이름을 입력하세요 : ");
        fgets(stu.name,sizeof(stu.name),stdin);
        stu.name[strlen(stu.name)-1]='\0';
        __fpurge(stdin);
        printf("국어점수를 입력하세요 : ");
        scanf("%d",&stu.korean);
        __fpurge(stdin);
        printf("영어점수를 입력하세요 : ");
        scanf("%d",&stu.english);
        __fpurge(stdin);
        printf("수학점수를 입력하세요 : ");
        scanf("%d",&stu.math);
        __fpurge(stdin);
        stu.all= stu.korean + stu.english + stu.math;
        stu.avg= stu.all/3;

        fwrite(&stu,sizeof(score),1,fp);
    }
    fclose(fp);
}
```

output.c

```
#include<stdio.h>
#include"mystruct.h"
void output(char *file)
{
    score stu;
    FILE *fp;
    int n;
    printf("성적 출력 프로그램입니다.\n");
    fp=fopen(file,"rb");
    if(fp==NULL)
    {
        perror("파일을 여는데 오류가 발생하였습니다.");
        exit(1);
    }
    printf("%8s\t%8s\t%8s\t%8s\t%8s\t%8s\t%8s\n", "학번 ", "이름 ", "국어 ", "영어 ", "수학 ", "총점 ", "평균 ");
    while ((n=fread(&stu,sizeof(score),1,fp))>0)
    {
        printf("%8d%7s\t%7d\t%7d\t%7d\t%7d\t%.2f\n",stu.id, stu.name, stu.korean, stu.english, stu.math, stu.all, stu.avg);
    }
    fclose(fp);
}
```

append.c

```
#include<stdio.h>
#include"mystruct.h"

void append(char *file, int id, char *name,int korean, int english, int math)
{
    printf("성적 추가 프로그램입니다.\n");
    printf("추가할 학생 데이터는 다음과 같습니다.\n");
    printf("%8s\t%8s\t%8s\t%8s\t%8s\n","학 번","이 름","국 어","영 어","수 학");
    printf("%8d%7s\t%7d\t%7d\t%7d\n",id,name,korean,english,math);

    int all;
    all=korean+english+math;
    float avg;
    avg=all/3;
    int i;
    score stu;
    FILE *fp;
    fp=fopen(file,"ab");
    if(fp==NULL)
    {
        perror("파일을 여는데 오류가 발생하였습니다.");
        exit(1);
    }
    stu.id=id;
    for(i=0; i<=sizeof(stu.name); i++)
    {
        stu.name[i]=name[i];
    }
    stu.korean=korean;
    stu.english=english;
    stu.math=math;
    stu.all=all;
    stu.avg=avg;
    fwrite(&stu,sizeof(score),1,fp);

    fclose(fp);
}
```

process.c

```
#include<stdio.h>

#include"mystruct.h"
void process(char *file)
{
    score stu;
    FILE *fp;
    printf("성적 처리 프로그램입니다.\n");
    fp=fopen(file,"rb");
    if(fp==NULL)
    {
        perror("파일을 여는데 오류가 발생하였습니다.");
        exit(1);
    }
    int n,count=0,total_korean=0,total_english=0,total_math=0,total=0;
    float avg,total_avg,korean_avg,english_avg,math_avg;

    while ((n=fread(&stu,sizeof(score),1,fp))>0)
    {
        count++;
        total_korean=total_korean+stu.korean;
        total_english=total_english+stu.english;
        total_math=total_math+stu.math;
        avg=avg+stu.avg;
    }

    total=total_korean+total_english+total_math;
    total_avg=avg/count;
    korean_avg=total_korean/count;
    english_avg=total_english/count;
    math_avg=total_math/count;

    printf("%5s %5s %5s %5s %5s %5s %5s %5s\n","국어 총점 ","국어 평균 ","영어 총점 ","영어 평  

    균 ","수학 총점 ","수학 평균 ","전체 총점 ","전체 평균 ");
    printf(" %5d %.2f %5d %.2f %5d %.2f %5d %.2f\n",total_korean,  

    korean_avg,total_english,english_avg,total_math,math_avg,total, total_avg);

    fclose(fp);
}
```


search.c

```
#include<stdio.h>
#include<string.h>

#include"mystruct.h"

void search(char *file,int search_opt,char *optarg)
{
    score stu;
    int n,return_strcmp,search_id,exist_check=0;
    int best_stu;
    float tmp;
    printf("성적 탐색 프로그램입니다.\n");
    FILE *fp;
    char search_name[sizeof(stu.name)];
    fp=fopen(file,"rb");
    if(fp==NULL)
    {
        perror("파일을 여는데 오류가 발생하였습니다.");
        exit(1);
    }
    switch(search_opt)
    {
        case 'n':
            memset(search_name,0,sizeof(search_name));
            strncpy(search_name,optarg,sizeof(search_name));
            printf("이름으로 성적을 찾습니다. (이름: %s)\n",search_name);
            printf("%8s\t%8s\t%8s\t%8s\t%8s\t%8s\t%8s\t%8s\n","학번","이름","국어","영어","수학","중점","평균");
            while ((n=fread(&stu,sizeof(score),1,fp))>0)
            {
                return_strcmp=strcmp(stu.name,search_name);
                if(return_strcmp==0)
                {
                    printf("%8d%7s\t%7d\t%7d\t%7d\t%7d\t%.2f\n",stu.id, stu.name, stu.korean, stu.english, stu.math, stu.all, stu.avg);
                    exist_check++;
                }
            }
            if(exist_check==0)
            {
                printf("해당 학생의 성적은 없습니다.\n");
                exit(3);
            }
            fclose(fp);
            break;
        case 'i':
            search_id=atoi(optarg);
            printf("학번으로 성적을 찾습니다. (학번: %s)\n",optarg);
            printf("%8s\t%8s\t%8s\t%8s\t%8s\t%8s\t%8s\t%8s\n","학번","이름","국어","영어","수학","중점","평균");
            while ((n=fread(&stu,sizeof(score),1,fp))>0)
            {
                if(stu.id==search_id)
                {
                    printf("%8d%7s\t%7d\t%7d\t%7d\t%7d\t%.2f\n",stu.id, stu.name, stu.korean, stu.english, stu.math, stu.all, stu.avg);
                    exist_check++;
                }
            }
            if(exist_check==0)
            {
                printf("해당 학생의 성적은 없습니다.\n");
                exit(3);
            }
            fclose(fp);
            break;
    }
}
```

```

case 'o':
    printf("%8s\t%8s\t%8s\t%8s\t%8s\t%8s\t%8s\n", "학번", "이름", "국어", "영어", "수학", "총점", "평균");
    tmp=0;
    while ((n=fread(&stu,sizeof(score),1,fp))>0)
    {
        if(stu.avg>=tmp)
        {
            best_stu=stu.id;
        }
    }
    fseek(fp,0L,SEEK_SET);
    while ((n=fread(&stu,sizeof(score),1,fp))>0)
    {
        if(best_stu==stu.id)
        {
            printf("%8d%7s\t%7d\t%7d\t%7d\t%7d\t%.2f\n",stu.id, stu.name, stu.korean, stu.e
nglish, stu.math, stu.all, stu.avg);
        }
    }
    fclose(fp);
    break;
case 's':
    printf("%8s\t%8s\t%8s\t%8s\t%8s\t%8s\t%8s\n", "학번", "이름", "국어", "영어", "수학", "총점", "평균");
    tmp=0;
    while ((n=fread(&stu,sizeof(score),1,fp))>0)
    {
        if(stu.avg>=tmp)
        {
            tmp=stu.avg;
        }
        else if(stu.avg<tmp)
        {
            tmp=stu.avg;
        }
    }
    fseek(fp,0L,SEEK_SET);
    while ((n=fread(&stu,sizeof(score),1,fp))>0)
    {
        if(stu.avg==tmp)
        {
            printf("%8d%7s\t%7d\t%7d\t%7d\t%7d\t%.2f\n",stu.id, stu.name, stu.korean, stu.e
nglish, stu.math, stu.all, stu.avg);
        }
    }
    fclose(fp);
    break;
default :
    printf("아무 옵션도 선택하지 않아 종료합니다.\n");
    fclose(fp);
    break;
}

```

input.c

성적을 입력하는 input함수입니다. char*file로 main에서 argv[1]을 받아옵니다. (데이터파일)
score stu로 구조체를 선언해줍니다. fp파일포인터를 하나 선언해주고 file을 wb+권한으로 fopen합니다.
이때 open하지 못하여 null을 리턴받을시 오류메시지를 출력해주었습니다. open오류시 리턴값은 1로 통일해주었습니다. 반복문으로 사용자에게 학생의 성적을 입력받습니다. memset으로 구조체를 초기화해준 후 scanf문으로 %d 형에대해서 값을 입력받습니다. %s입력을 받을때는 fgets함수를 stdin입력으로 받아주었습니다. fgets함수는 엔터 입력또한 받으므로 개행문자를 삭제하기 위해 stu.name배열의 strlen-1값을 'W0'으로 할당해주었습니다.
그후 stu.all, stu.avg를 총점, 평균으로 저장해줍니다. fwrite함수를 호출하여 stu구조체에 struct score의 사이즈만큼 쓰기 해줍니다. 전에 배열을 쓰려다가 교수님께서 하나씩 받아 저장하면 된다는 조언을 받아 루프문을 돌리면 되겠다 생각하여 이렇게 작성하였습니다. 쓰기 작업이 끝나면 fclose를 이용해 fp를 닫아줍니다.

output.c

data출력함수인 output함수입니다. score stu로 구조체를 선언해줍니다. 파일포인터 fp와 fread함수 사용시 사용될 변수 n을 선언해줍니다. fp로 main에서 받아온 데이터파일을 rb권한으로 열어줍니다. 이때 NULL리턴시 오류를 출력해줍니다. 그후 1열에 학번 이름 국어 영어 수학 총점 평균을 출력해줍니다.
fread함수를 이용하여 score구조체크기만큼 fp에서 읽어옵니다. 이때 읽어온 stu.id, stu.name ... stu.avg를 출력해줍니다. 이 함수는 더 이상 불러올 데이터가 없을때까지 while반복문에 의해 루프됩니다. 그렇기 때문에 입력단계에서 입력한 데이터를 순차적으로 불러와 한줄씩 학생데이터를 출력합니다.

append.c

데이터를 추가하는 append함수입니다. 메인에서 데이터파일과 argv값을 불러와 함수에서 사용합니다.
먼저 추가하는 학생의 데이터를 출력하도록 해주었습니다. 단순한 메인에서 불러온 값들의 출력입니다.
argument로 총점과 평균은 받아오지 않았기 때문에 int all과 float avg를 선언하여 총점과 평균값을 저장해줍니다.
그리고 stu구조체를 선언해준뒤 fp파일포인터 선언후 데이터파일을 ab권한으로 열어줍니다. ab권한은 append binary권한으로 기존파일을 유지한채 파일의 끝에서 추가합니다.
그후 stu.id...stu.avg값에 입력받은값을 저장합니다. stu.name의 경우 char *name 포인터형으로 받았기 때문에 반복문을 이용하여 값을 저장해주었습니다. 그후 fp를 닫아줍니다.

process.c

성적을 처리하는 process함수입니다. 메인에서 data파일을 받아 사용합니다. score stu로 stu구조체를 선언해줍니다. fp파일포인터를 선언해주고 fopen으로 데이터파일을 rb권한으로 열기합니다. 각 과목의 총점, 전체총점, 전체 평균을 저장할 변수들을 선언해줍니다.
fread함수를 score 구조체사이즈만큼 루프돌려 toal값들을 누적연산합니다. 이때 평균값을 구할 때 나눌 몫인 count도 count++를 해줍니다. 그후 총점과 평균부분은 총점=국어+수학+영어로 할당해주고 평균부분은 총점을 전에 선언해주었던 count값으로 나눠주어 과목평균과 전체평균을 저장해줍니다.
그후 print문으로 해당 값들을 출력해주고 fp파일포인터를 닫아줍니다.

search.c

성적 검색함수 search함수입니다. stu구조체를 선언해줍니다. 그후 fread함수를 위한 변수n과 strcmp의 리턴값을 저장해줄 return_strcmp, atoi(optarg)를 저장할 search_id, 학생의 성적 존재 여부를 검사할 exist_check를 선언해주었습니다. best_stu는 o 옵션에서 1명만 출력하기위해 선언해주었고, tmp는 평균을 이용하여 가장 좋은 성적의 학생을 탐색할 때 avg값을 임시로 저장하기 위해 사용하였습니다.

<옵션 n>

fp파일포인터를 선언해주고 optarg값을 strncpy를 이용하여 저장할 search_name을 선언해줍니다. 데이터를 rb권한으로 열고 메인에서 getopt를 이용하여 받은 search_opt를 이용해 switch문을 돌려줍니다. n인 경우 main에서 받은 optarg를 strncpy를 이용하여 search_name에 저장해줍니다. 그후 while반복문을 통해 fread함수를 호출하여 fp파일포인터가 가리키는파일을 score구조체의 사이즈만큼 구조체 stu에 읽어옵니다.
읽어오는 과정에서 name값이 일치하는지 검사하기위해 strcmp함수를 사용하여 stu.name과 search_name이 길이와 내용이 동일한지 검사합니다. 이때 strcmp는 stu.name이 search_name보다 작은 경우 0보다 작은값, 같은 경우 0, 큰 경우 0보다 큰값을 리턴하기 때문에 이 리턴값이 0인경우를 조건문을 걸어 printf문으로 해당 오프셋의 구조체를 출력해줍니다. 이때 exist_check값을 ++해줍니다. fread가 더 이상 읽어올것이없어 반복문이 종료되면 exist_check를 검사해줍니다. 하나라도 출력될 경우 ++해주었기 때문에 0인 경우 성적이 없다는 메시지를 출력합니다.

<옵션 i>

기본적인 원리는 옵션n과 같습니다만, id의 경우는 int자료형이기 때문에 strncpy와 strcmp를 사용하지 않고 string을 정수형으로 바꾸어주는 atoi함수를 사용하여 search_id에 optarg로 받은 학번을 int형으로 변환하여 저장시켜줍니다. 그후 while문으로 fread함수를 돌려주고 stu.id가 search_id와 같은 경우 옵션n과 같이 해당 오프셋의 구조체를 출력해주고 exist_check를 ++해줍니다. fread함수가 더 이상 읽을것이 없으면 while문을 탈출하고 exist_check를 검사하게됩니다. 이때도 하나라도 출력될 경우 exist_check값이 ++되기 때문에 0인 경우 성적이 없음을 출력합니다.

<옵션 o>

옵션 o는 가장 좋은 학생의 성적을 출력하는데 동점자가 있을시 마지막에 탐색한 학생을 출력하도록 합니다. tmp값을 0으로 초기화해주고 fread함수로 score구조체의 사이즈만큼 반복하여 불러옵니다. 이때 tmp값에 가장 높은 평균값을 저장하기위해 stu.avg와 tmp값을 비교해줍니다. 가장 높은 평균값을 저장하기 때문에 stu.avg>=tmp인 경우 tmp=stu.avg를 할당해주고, 이때 한학생만 출력해야하므로 이때 오프셋의 stu.id또한 best_stu에 저장합니다. 반대로 stu.avg가 tmp보다 작은 경우 tmp는 그대로 tmp입니다. 이렇게 루프문을 돌리게되면 가장 높은 성적이 tmp에 저장되게됩니다. 그후 fseek함수를 이용하여 fread하여 끝으로 이동한 오프셋을 다시 처음으로 돌려줍니다. 그후 다시 fread함수를 이용하여 아까 저장하였던 best_stu값을 stu.id와 비교하여 일치하는 경우 해당 오프셋의 stu구조체를 출력해줍니다.

<옵션 s>

옵션 s는 옵션o와 비슷하지만 동점자가있는 경우 모두 출력하도록 해주어야합니다. tmp를 0으로 초기화해주고 fread함수를 이용하여 data파일을 score구조체 사이즈만큼 stu에 읽어옵니다. 만약 stu.avg가 tmp보다 크거나 같은 경우 tmp에 stu.avg를 저장해줍니다. 작다면 tmp는 tmp그대로입니다. 이렇게 루프문을 돌려 fread가 더 이상 읽어올것이 없게되면 탈출하게되고 tmp엔 가장높은성적의 평균이 저장됩니다. fseek함수로 오프셋을 처음으로 되돌립니다. 그후 다시 fread함수를 호출하여 전에 저장하였던 tmp파일을 이용해 루프를 돌리며 stu.avg와 tmp를 검사해줍니다. 이때 두 개가 일치하는 경우 해당 오프셋의 구조체를 출력합니다. 이 경우 동점자도 같은 avg값을 가지고 있으므로 동점자 수 만큼 print문이 실행됩니다.

프로그램 main

input_prog.c

```
#include<stdio.h>
#include"mystruct.h"
#include"input.h"

int main(int argc, char *argv[])
{
    if(argc<2)
    {
        printf("데이터 파일 1개와 같이 사용하세요.\n사용예 : %s data.dat",argv[0]);
        exit(1);
    }
    char *file;
    file=argv[1];

    input(file);

    return 0;
}
```

argv로 데이터파일을 받습니다. 그후 input함수를 호출합니다.

output_prog.c

```
#include<stdio.h>
#include"mystruct.h"
#include"output.h"

int main(int argc, char *argv[])
{
    if(argc<2)
    {
        printf("파일 1개와 같이 사용하세요.\n사용예 : %s data.dat",argv[0]);
        exit(1);
    }
    char *file;
    file=argv[1];
    output(file);

    return 0;
}
```

argv로 데이터파일을 받습니다. 그후 output함수를 호출합니다.

append_prog.c

```
#include<stdio.h>

#include"mystruct.h"
#include"append.h"

int main(int argc, char *argv[])
{
    if(argc!=7)
    {
        printf("이 프로그램은 파일명,학번,이름,국어점수,영어점수,수학점수를 인자로 받아야합니다\n");
        printf(" Usage: %s filename (학번) (이름) (국어) (영어) (수학)\n",argv[0]);
        exit(2);
    }
    char *file;
    file=argv[1];
    append(file,atoi(argv[2]),argv[3],atoi(argv[4]),atoi(argv[5]),atoi(argv[6]));
    return 0;
}
```

argc가 7이 아닌 경우 사용법을 출력 후 종료합니다. 그후 argv로 파일을 받습니다. append함수를 호출합니다. append함수에 데이터파일, atoi로 변환한 정수값, argv로 받은 문자열을 전달합니다.

process_prog.c

```
#include<stdio.h>

#include"mystruct.h"
#include"process.h"

int main(int argc, char *argv[])
{
    if(argc<2)
    {
        printf("파일 1개와 같이 사용하세요.\n사용예 : %s data.dat",argv[0]);
        exit(1);
    }
    char *file;
    file=argv[1];
    process(file);

    return 0;
}
```

argv로 데이터파일을 받습니다. process함수를 호출합니다.

search_prog.c

```
#include<stdio.h>

#include"mystruct.h"
#include"search.h"

int main(int argc, char *argv[])
{
    extern int optind;
    char *file;
    file=argv[1];
    if(argc-optind<2)
    {
        printf("옵션 하나와 요구된 arguments를 같이 사용하세요.\n");
        printf(" Usage: %s (option) (required arguments)\n",file);
        printf(" -n (이름) 이름으로 성적 검색\n");
        printf(" -i (학번) 학번으로 성적 검색\n");
        printf(" -o 가장 성적 좋은 학생 출력 (등점인 경우 마지막 학생 출력)\n");
        printf(" -s 가장 성적 좋은 학생 출력 (등점인 경우 모든 학생 출력)\n");
        exit(1);
    }
    extern char *optarg;
    int search_opt;

    search_opt=getopt(argc,argv,"n:i:os");
    search(file,search_opt,optarg);

    return 0;
}
```

argc-optind가 2미만인 경우 사용법과 옵션을 출력합니다. extern char *optarg를 받고 getopt값을 저장할 search_opt변수를 선언합니다. getopt함수를 이용하여 각옵션, 옵션이 요구한 argument를 받습니다. search함수를 호출합니다. 데이터파일과 getopt로 받은옵션, optarg를 함수로 전달합니다.

ls

```
[khm970514@lily ex5]$ ls
append      append_prog.c  input.c        input_prog.o  output.h       process      process_prog.c  search.h      src.c
append.c    append_prog.o  input.h        mystruct.h    output.o       process.c    process_prog.o  search.o      src.o
append.h    bak_dir        input.o        output         output_prog.c  process.h    search          search_prog.c
append.o    input          input_prog.c   output.c       output_prog.o  process.o    search.c        search_prog.o
[khm970514@lily ex5]$
```

입력프로그램

```
[khm970514@lily ex5]$ gcc -o input input_prog.o input.o
[khm970514@lily ex5]$ ./input data.dat
성적 입력 프로그램입니다.
학번을 입력하세요 (종료 0) : 20163301
이름을 입력하세요 : 김현일
국어점수를 입력하세요 : 56
영어점수를 입력하세요 : 67
수학점수를 입력하세요 : 84
학번을 입력하세요 (종료 0) : 20163302
이름을 입력하세요 : 김현이
국어점수를 입력하세요 : 45
영어점수를 입력하세요 : 67
수학점수를 입력하세요 : 19
학번을 입력하세요 (종료 0) : 20163303
이름을 입력하세요 : 김현삼
국어점수를 입력하세요 : 74
영어점수를 입력하세요 : 28
수학점수를 입력하세요 : 93
학번을 입력하세요 (종료 0) : 20163304
이름을 입력하세요 : 김현사
국어점수를 입력하세요 : 45
영어점수를 입력하세요 : 12
수학점수를 입력하세요 : 68
학번을 입력하세요 (종료 0) : 0
0을 입력하여 종료합니다.
[khm970514@lily ex5]$
```


출력프로그램

```
[khm970514@lily ex5]$ gcc -o output output_prog.o output.o
[khm970514@lily ex5]$ ./output data.dat
성적 출력 프로그램입니다.
  학 번      이 름      국 어      영 어      수 학      총 점      평 균
20163301김 현 일      56      67      84      207 69.00
20163302김 현 이      45      67      19      131 43.00
20163303김 현 삼      74      28      93      195 65.00
20163304김 현 사      45      12      68      125 41.00
[khm970514@lily ex5]$
```

추가프로그램

```
[khm970514@lily ex5]$ gcc -o append append_prog.o append.o
[khm970514@lily ex5]$ ./append data.dat
이 프로그램은 파일명,학번,이름,국어점수,영어점수,수학점수를 인자로 받아야합니다
Usage: ./append filename (학 번) (이 름) (국 어) (영 어) (수 학)
[khm970514@lily ex5]$ ./append data.dat 20163305 김현오 99 99 99
성적 추가 프로그램입니다.
추가할 학생 데이터는 다음과 같습니다.
  학 번      이 름      국 어      영 어      수 학
20163305김 현 오      99      99      99
[khm970514@lily ex5]$ ./append data.dat 20163306 김현오 99 99 99
성적 추가 프로그램입니다.
추가할 학생 데이터는 다음과 같습니다.
  학 번      이 름      국 어      영 어      수 학
20163306김 현 오      99      99      99
[khm970514@lily ex5]$
```

처리프로그램

```
[khm970514@lily ex5]$ gcc -o process process_prog.o process.o
[khm970514@lily ex5]$ ./process data.dat
성적 처리 프로그램입니다.
국 어 총 점   국 어 평 균   영 어 총 점   영 어 평 균   수 학 총 점   수 학 평 균   전 체 총 점   전 체 평 균
    418       69.00       372       62.00       462       77.00       1252       69.33
[khm970514@lily ex5]$
```

탐색프로그램

```
[khm970514@lily ex5]$ gcc -o search search_prog.o search.o
```

```
[khm970514@lily ex5]$ ./search data.dat -n 김현오
```

성적 탐색 프로그램입니다

이름으로 성적을 찾습니다. (이름 : 김현오)

학번	이름	국어	영어	수학	총점	평균
20163305	김현오	99	99	99	297	99.00
20163306	김현오	99	99	99	297	99.00

```
[khm970514@lily ex5]$ ./search data.dat -i 20163305
```

성적 탐색 프로그램입니다

학번으로 성적을 찾습니다. (학번 : 20163305)

학번	이름	국어	영어	수학	총점	평균
20163305	김현오	99	99	99	297	99.00

```
[khm970514@lily ex5]$ ./search data.dat -o
```

성적 탐색 프로그램입니다

학번	이름	국어	영어	수학	총점	평균
20163306	김현오	99	99	99	297	99.00

```
[khm970514@lily ex5]$ ./search data.dat -s
```

성적 탐색 프로그램입니다

학번	이름	국어	영어	수학	총점	평균
20163305	김현오	99	99	99	297	99.00
20163306	김현오	99	99	99	297	99.00

```
[khm970514@lily ex5]$ █
```