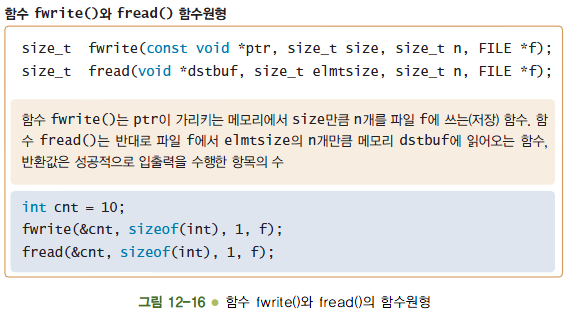
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C2\_실습과제\_11주** | **이름 :** | **이갑성** | **학번 :** | **20135151** |

* **강의내용**

1. 이진 파일 입출력

****

* + **첫 번째 인자: 입출력 될 자료의 주소값**
  + **두 번째 인자: 입출력 될 자료 항목의 바이트 크기**
  + **세 번째 인자: 입출력 될 항목의 개수이며**
  + **마지막 인자: 입출력 될 파일 포인터**

#include <stdio.h>

#include <stdlib.h>

struct personscore{

int number; //번호

char name[40]; //이름

int mid; //중간성적

int final; //기말성적

int quiz; //퀴즈성적

};

typedef struct personscore pscore;

void printhead();

int main(){

char fname[] = "score.bin";

char line[80];

FILE \*f;

int cnt = 0;

pscore score;

if (fopen\_s(&f, fname, "wb" ) != 0 ){ //쓰기 모드로 파일 열기

printf("파일이 열리지 않습니다.\n");

exit(1);

}

while (1){

printf("이름과 성적(중간, 기말, 퀴즈)을 입력 하세요.\n");

fgets(line, 80, stdin); //키보드로부터 입력

if (feof(stdin))

break;

//표준 입력의 한 줄을 구조체의 멤버별로 자료를 입력

//sscanf(line, "%s %d %d %d", score.name, &score.mid, &score.final, &score.quiz);

sscanf\_s(line, "%s %d %d %d", score.name, 40,&score.mid, &score.final, &score.quiz);

score.number = ++cnt;

fwrite(&score, sizeof(pscore), 1, f);

}

fclose(f);

if (fopen\_s(&f, fname, "rb" ) != 0 ){

printf("파일이 열리지 않습니다.\n");

exit(1);

}

printhead();

fread(&score, sizeof(pscore), 1, f);

while (!feof(f)){ //표준 출력에 쓰기

fprintf(stdout, "%6d%18s%8d%8d%8d\n",

score.number, score.name, score.mid, score.final, score.quiz);

fread(&score, sizeof(pscore), 1, f);

}

printf("%s\n", " -------------------------------------------------");

fclose(f);

return 0;

}

void printhead() {

printf("%s\n", " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

printf("%8s%15s%10s%8s%8s\n", "번호", "이름", "중간", "기말", "퀴즈");

printf("%s\n", " -------------------------------------------------");

}

|  |
| --- |
| * 실행결과 |
| * 파일내용 |

1. 파일 임의 접근

* fseek() : 파일 위치 이동
* ftell() : 현재 파일 위치까지의 거리를 byte로 반환
* rewind() : 파일 위치를 파일 시작점(0)으로 이동

fseek(fp, 0L, SEEK\_END); //파일 위치를 파일 마지막으로 이동



* 파일에 저장된 전체 레코드 개수 계산하기

int block\_number=ftell(f)/sizeof(pscore)

* n번째 레코드로 파일 위치 이동

fseek(f, sizeof(pscore)\*(n-1), SEEK\_SET);

* 현재 파일 위치까지의 거리를 byte수로 반환

long loc =ftell(f);

#include <stdio.h>

void printhead();

void printscore(FILE \*f);

void appendscore(FILE \*f, int cnt);

struct personscore{

int number; //번호

char name[40]; //이름

int mid, final, quiz; //중간성적, 기말성적, 퀴즈성적

};

typedef struct personscore pscore;

int main(){

char fname[] = "score.bin";

FILE \*f;

int cnt = 0;

long offset = 0;

pscore score;

if (fopen\_s(&f,fname, "a+" ) != 0) {

printf("파일이열리지않습니다.\n");

exit(1);

}

offset = (long) sizeof(pscore); //구조체 하나의 크기

fseek(f, 0L, SEEK\_END); //파일위치를 마지막으로 이동

if (ftell(f) != 0){ //파일에 저장된 자료가 있으면

printscore(f);

fseek(f, -offset, SEEK\_CUR); //파일의 현재포인터에서 한 학생 만큼 뒤로 이동

fread(&score, sizeof(pscore), 1, f); //마지막 학생을 읽음

cnt = score.number; //마지막 학생의 번호 저장

printf("\n제일 마지막 번호가 %d번 입니다. \n\n", cnt);

fseek(f, 0L, SEEK\_END); //파일 위치를 마지막으로 이동

}

appendscore(f, cnt);

printscore(f);

fclose(f);

return 0;

}

void printhead() {

printf("%s\n", " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

printf("%8s%15s%10s%8s%8s\n", "번호", "이름", "중간", "기말", "퀴즈");

printf("%s\n", " -------------------------------------------------");

}

void appendscore(FILE \*f, int cnt){

pscore score;

char line[80];

printf("추가할 이름과 성적(중간, 기말, 퀴즈)을 입력하세요.\n\n");

fgets(line, 80, stdin);

while (!feof(stdin)) {

sscanf\_s(line, "%s %d %d %d", score.name, 40, &score.mid, &score.final, &score.quiz);

score.number = ++cnt;

fwrite(&score, sizeof(pscore), 1, f);

fgets(line, 80, stdin);

}

}

void printscore(FILE \*f) {

pscore score;

rewind(f); //파일 위치를 처음으로 이동

printhead();

fread(&score, sizeof(pscore), 1, f);

while (!feof(f)) { //표준 출력장치로 출력

fprintf(stdout, "%6d%18s%8d%8d%8d\n",

score.number, score.name, score.mid, score.final, score.quiz);

fread(&score, sizeof(pscore), 1, f);

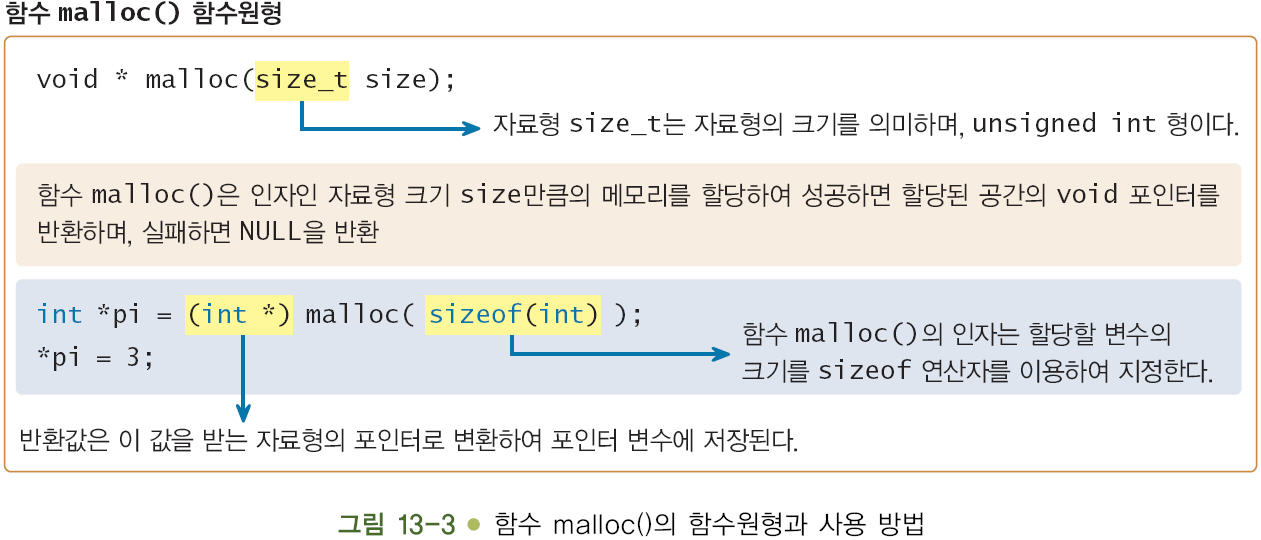
}

fprintf(stdout, "%s\n", " -----------------------------------------------");

}

|  |
| --- |
| * 실행결과 |
| * 파일내용 |

1. 동적 메모리 할당



#include <stdio.h>

#include <stdlib.h>

int main(void) { //동적 메모리에 여러 성적을 저장하여 합과 평균을 구하여 출력

int \*ary = NULL;

int i = 0, n = 0, sum = 0;

printf("입력할 점수의 개수를 입력>> ");

scanf\_s("%d", &n);

//입력한 개수 만큼의 메모리 할당

if ((ary = (int \*)malloc(sizeof(int)\*n)) == NULL) {

printf("메모리 할당에 문제가 있습니다.");

exit(1);

}

printf("%d개의 점수 입력>> ", n);

for (i = 0; i < n; i++) {

scanf\_s("%d", (ary + i)); //할당된 공간의 주소값 계산

sum += \*(ary + i); //sum += ary[i];, 할당된 공간 참조

}

printf("합: %d 평균: %.1f\n", sum, (double)sum / n);

free(ary); //할당 받은 메모리 해제

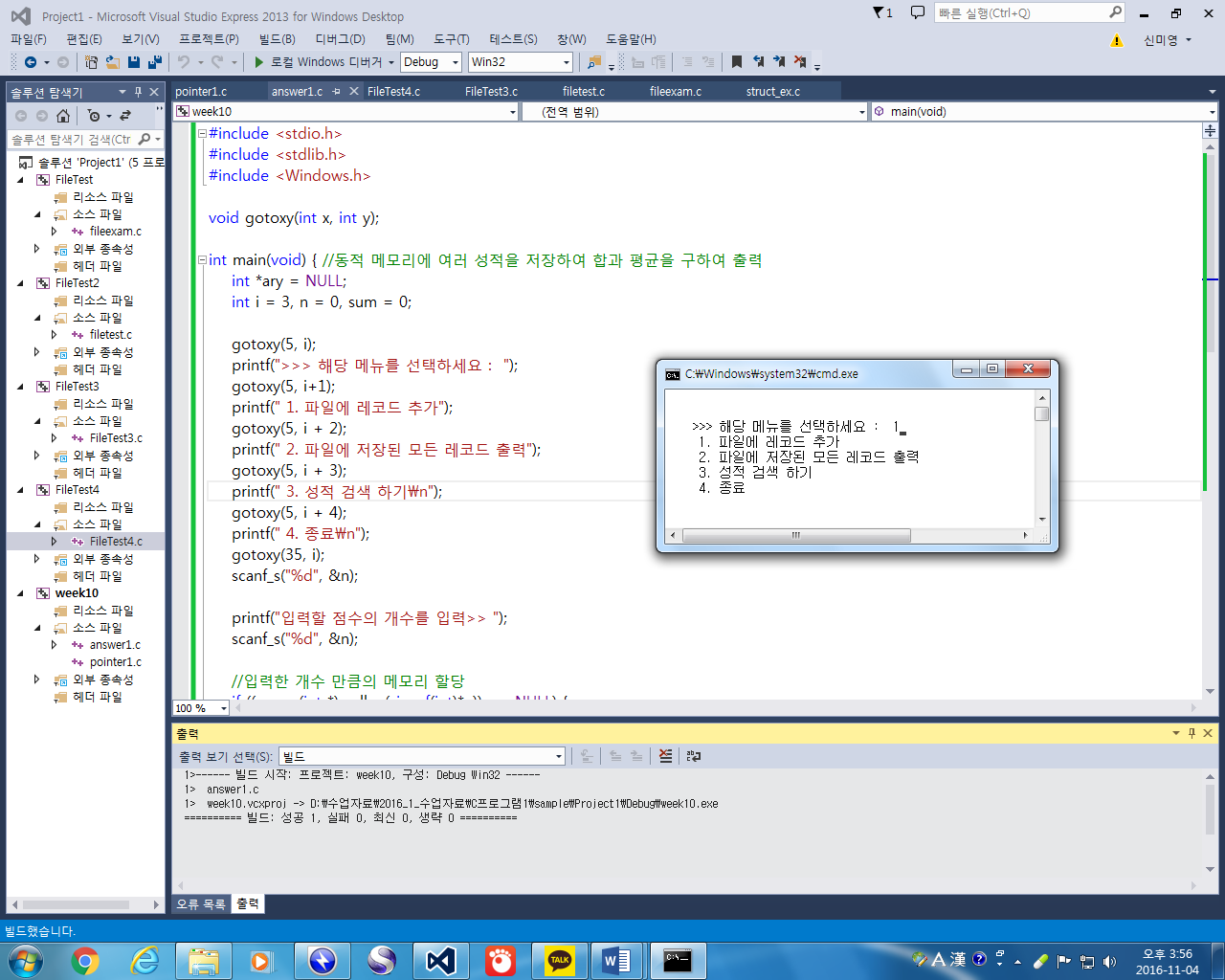
return 0;

}

|  |
| --- |
| * 실행결과 |

* 프로그램 과제

1. “파일 임의 접근”에서 제시된 예제를 선택된 메뉴에 따라 처리될 수 있도록 수정하시오. 4를 입력하면 반복문이 종료되도록 한다. 3번은 입력된 이름과 일치하는 레코드를 검색하여 성적을 출력하도록 한다.



|  |
| --- |
| [소스]  #include <stdio.h>  #include <stdlib.h>  struct personscore {  int number; //번호  char name[40]; //이름  int mid; //중간성적  int final; //기말성적  int quiz; //퀴즈성적  };  typedef struct personscore pscore;  int menu();  void addFile(FILE \*);  void printFile(FILE \*);  void searchFile(FILE \*);  void printhead();  int main()  {  int select;  char fname[] = "score.bin";  FILE \*f;  if (fopen\_s(&f, fname, "a+") != 0) {  printf("파일이열리지않습니다.\n");  exit(1);  }  while ((select = menu()) != 4)  {  switch (select)  {  case 1:  addFile(f);  break;  case 2:  printFile(f);  break;  case 3:  searchFile(f);  break;  default:  break;  }  }  printf("\n \n==========프로그램을 종료합니다.========== \n\n");  fclose(f);  }  int menu()  {  int select;  printf("==============메뉴==============\n");  printf("1. 파일에 레코드 추가 \n");  printf("2. 파일에 저장된 모든 레코드 출력 \n");  printf("3. 성적 검색 하기 \n");  printf("4. 종료 \n");  printf(">>> 해당파일 메뉴를 선택하세요 : ");  scanf("%d", &select);  fflush(stdin);  return select;  }  void printhead()  {  printf("%s\n", " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  printf("%8s%15s%10s%8s%8s\n", "번호", "이름", "중간", "기말", "퀴즈");  printf("%s\n", " -------------------------------------------------");  }  void addFile(FILE \*f) //성적 추가 입력  {  int cnt = 0;  long offset = 0;  pscore score;  char line[80];  fseek(f, 0L, SEEK\_END);  if (ftell(f) == 0)  {  printf("파일에 데이터가 없습니다. \n");  }  rewind(f);  offset = (long) sizeof(pscore); //구조체 하나의 크기  fseek(f, 0L, SEEK\_END); //파일위치를 마지막으로 이동  if (ftell(f) != 0) { //파일에 저장된 자료가 있으면  fseek(f, -offset, SEEK\_CUR); //파일의 현재포인터에서 한 학생 만큼 뒤로 이동  fread(&score, sizeof(pscore), 1, f); //마지막 학생을 읽음  cnt = score.number; //마지막 학생의 번호 저장  printf("\n제일 마지막 번호가 %d번 입니다. \n\n", cnt);  fseek(f, 0L, SEEK\_END); //파일 위치를 마지막으로 이동  }  printf("추가할 이름과 성적(중간, 기말, 퀴즈)을 입력하세요.\n\n");  while (1)  {  fgets(line, 80, stdin);  if (feof(stdin))  {  break;  }  sscanf\_s(line, "%s %d %d %d", score.name, 40, &score.mid, &score.final, &score.quiz);  score.number = ++cnt;  fwrite(&score, sizeof(pscore), 1, f);  }  }  void printFile(FILE \*f) //파일에 저장되있는 성적 출력  {  pscore score;  rewind(f); //파일 위치를 처음으로 이동  printhead();  fseek(f, 0L, SEEK\_END);  if (ftell(f) == 0)  {  printf("파일에 데이터가 없습니다. \n");  }  rewind(f);  while (1) { //표준 출력장치로 출력  fread(&score, sizeof(pscore), 1, f); //파일에 저장되어있는 정보들을 읽어온다.  if (feof(f))  {  break;  }  fprintf(stdout, "%6d%18s%8d%8d%8d\n",  score.number, score.name, score.mid, score.final, score.quiz);  }  fprintf(stdout, "%s\n", " -----------------------------------------------");  }  void searchFile(FILE \*f) //파일에 저장되있는 정보 찾기  {  int i;  pscore score;  char name[10];  fseek(f, 0L, SEEK\_END);  if (ftell(f) == 0)  {  printf("파일에 데이터가 없습니다. \n");  }  rewind(f);  printf("찾고자하는 학생의 이름을 입력하세요: ");  scanf("%s", name);  fread(&score, sizeof(pscore), 1, f);  while (!feof(f))  {  if (strcmp(name, score.name) == 0)  {  printhead();  fprintf(stdout, "%6d%18s%8d%8d%8d\n", score.number, score.name, score.mid, score.final, score.quiz);  break;  }  else  {  fread(&score, sizeof(pscore), 1, f);  }  }  } |
| [실행결과] |
| [파일내용] |

1. 입력 받은 개수 만큼의 실수를 저장할 수 있는 메모리를 할당 받아 초기화 한 후 출력하는 프로그램을 작성하시오.

|  |
| --- |
| [소스]  #include <stdio.h>  #include <stdlib.h>  int main()  {  double \*ary = NULL;  int i = 0, n = 0;  printf("입력할 개수를 입력하세요. >>>");  scanf("%d", &n);  if ((ary = (double \*)malloc(sizeof(double)\*n)) == NULL)  {  printf("메모리 할당에 문제가 있습니다. \n");  exit(1);  }  printf("%d개의 초기화작업 실시! \n", n);  for (i = 0; i < n; i++)  {  printf("%d번째의 실수 초기화작업: ", i+1);  scanf("%lf", (ary + i));  }  for (i = 0; i < n; i++)  {  printf("%d번째의 실수값: %0.2lf \n", i+1, \*(ary + i));  }  } |
| [실행결과] |

1. “동적 메모리 할당”에서 제시된 예제를 다음과 같은 구조체를 처리할 수 있도록 수정하시오. 단, 함수를 정의하여 처리 할 것.

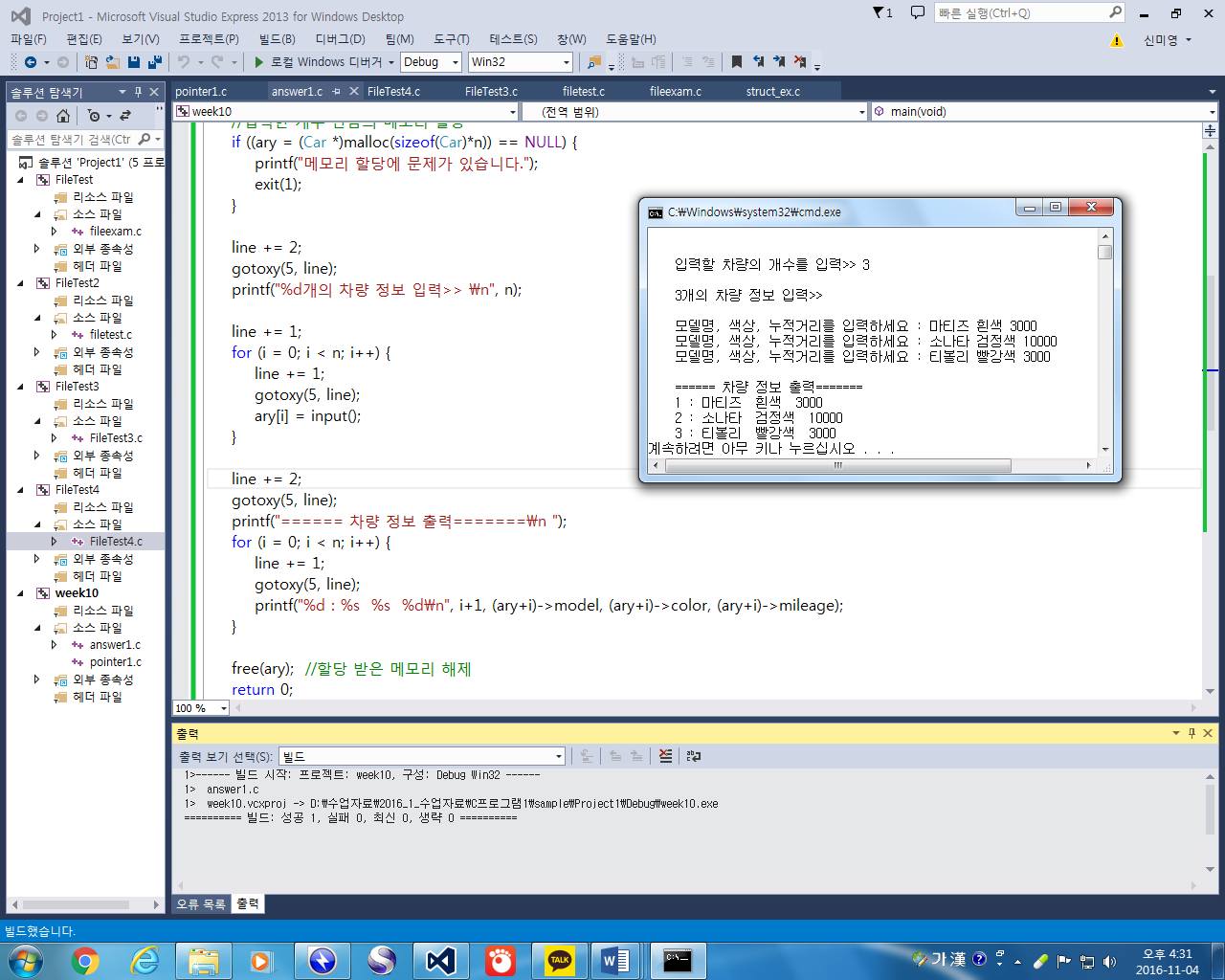
typedef struct{

char model[20]; //모델명

char color[10]; //색상

int mileage; //누적 거리

}Car;



|  |
| --- |
| [소스]  #include <stdio.h>  #include <stdlib.h>  typedef struct {  char model[20]; //모델명  char color[10]; //색상  int mileage; //누적 거리  }Car;  void input\_info(Car\*, int);  void print\_info(Car\*, int);  int main()  {  Car \*ary = NULL;  int n;  printf("\n\n입력할 차량의 개수를 입력>>> ");  scanf("%d", &n);  if ((ary = (Car \*)malloc(sizeof(Car)\*n)) == NULL)  {  printf("메모리 할당에 문제가 있습니다 \n");  exit(1);  }  input\_info(ary, n); // 정보입력 함수  print\_info(ary, n); //정보출력 함수    }  void input\_info(Car \*ary, int n)  {  int i;  printf("\n\n%d 개의 차량 정보 입력 >>> \n\n", n);  for (i = 0; i < n; i++)  {  printf("모델명, 색상, 누적거리를 입력하세요: ");  scanf("%s %s %d", (ary + i)->model, (ary + i)->color, &(ary + i)->mileage);  }  }  void print\_info(Car \*ary, int n)  {  int i;  printf("========차량 정보 출력========\n\n\n");  for (i = 0; i < n; i++)  {  printf("%d : %s %s %d \n", i + 1, (ary + i)->model, (ary + i)->color, (ary + i)->mileage);  }  } |
| [실행결과] |

1. 1번 문제의 파일에 저장된 레코드 개수만큼 메모리를 할당 받은 후, 할당 받은 메모리에 모든 레코드를 저장하고 출력하는 프로그램을 작성하시오.

힌트) 파일에 저장된 전체 레코드 수를 계산한 후, 레코드 개수만큼 메모리 할당을 받는다

fseek(f, 0L, SEEK\_END); //파일위치를 마지막으로 이동

int block\_number=ftell(f)/sizeof(pscore) //저장된 레코드 수 계산

|  |
| --- |
| [소스]  #include <stdio.h>  #include <stdlib.h>  struct personscore {  int number; //번호  char name[40]; //이름  int mid; //중간성적  int final; //기말성적  int quiz; //퀴즈성적  };  typedef struct personscore pscore;  void printhead();  int main()  {  FILE \*f;  int block\_number; // 저장된 레코드 수  pscore \*ary = NULL;  pscore score;  int cnt = 0;  int i;    if (fopen\_s(&f, "score.bin", "rb") != 0)  {  printf("파일을 찾지 못했습니다 \n");  exit(1);  }  fseek(f, 0L, SEEK\_END); //파일위치를 마지막으로 이동  block\_number = ftell(f) / sizeof(pscore); //저장된 레코드 수 계산    rewind(f); //파일위치를 처음으로 이동  if((ary = (pscore \*)malloc(sizeof(pscore)\*block\_number)) == NULL)  {  printf("메모리를 할당에 문제가 있습니다. \n");  exit(1);  }  printhead();    fread(&score, sizeof(pscore), 1, f);  while (!feof(f))  {  (ary + cnt)->number = score.number;  strcpy((ary + cnt)->name, score.name);  (ary + cnt)->mid = score.mid;  (ary + cnt)->final = score.final;  (ary + cnt)->quiz = score.quiz;  cnt++;  fread(&score, sizeof(pscore), 1, f);    }    fclose(f);    for (i = 0; i < cnt; i++)  {  printf("%6d%18s%8d%8d%8d\n", (ary + i)->number, (ary + i)->name, (ary + i)->mid, (ary + i)->final, (ary + i)->quiz);  }  free(ary);  }  void printhead()  {  printf("%s\n", " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  printf("%8s%15s%10s%8s%8s\n", "번호", "이름", "중간", "기말", "퀴즈");  printf("%s\n", " -------------------------------------------------");  } |
| [실행결과] |

1. 3번 문제의 파일에 저장된 레코드 중 n번째에 저장된 레코드를 출력하는 프로그램을 작성하시오.

|  |
| --- |
| [소스]    #include <stdio.h>  #include <stdlib.h>  typedef struct {  char model[20]; //모델명  char color[10]; //색상  int mileage; //누적 거리  int cnt; //번호  }Car;  void input\_info(Car\*, int);  void print\_info(Car\*, int);  void find\_info(Car\*, int, int);  int main()  {  Car \*ary = NULL;  int n;  int index;  printf("\n\n입력할 차량의 개수를 입력>>> ");  scanf("%d", &n);  if ((ary = (Car \*)malloc(sizeof(Car)\*n)) == NULL)  {  printf("메모리 할당에 문제가 있습니다 \n");  exit(1);  }  input\_info(ary, n); // 정보입력 함수  print\_info(ary, n); //정보출력 함수  printf("\n\n몇번째 차량의 정보를 불러오시겠습니다. \n");  scanf("%d", &index);  find\_info(ary, n, index); //정보찾기 함수  }  void input\_info(Car \*ary, int n)  {  int i;  printf("\n\n%d 개의 차량 정보 입력 >>> \n\n", n);    for (i = 0; i < n; i++)  {  printf("모델명, 색상, 누적거리를 입력하세요: ");  scanf("%s %s %d", (ary + i)->model, (ary + i)->color, &(ary + i)->mileage);  (ary + i)->cnt = i+1;  }  }  void print\_info(Car \*ary, int n)  {  int i;  printf("========차량 정보 출력========\n\n\n");  for (i = 0; i < n; i++)  {  printf("%d : %s %s %d \n", (ary+i)->cnt, (ary + i)->model, (ary + i)->color, (ary + i)->mileage);  }  }  void find\_info(Car \*ary, int n, int index)  {  int i;  int check = 0;  for (i = 0; i < n; i++)  {  if (index == (ary + i)->cnt)  {  printf("모델명: %s 색상: %s, 누적거리 %d \n", (ary + i)->model, (ary + i)->color, (ary + i)->mileage);  check = 1;  }  }  if (check == 0)  {  printf("해당번호의 차량의 정보가 존재하지 않습니다. \n");  }  } |
| [실행결과] |