|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2020\_1\_C\_12주 | 학번 : | 20175105 | 이름 : | 곽영주 |

* **강의 내용**

1. 입력과 출력으로 사용할 두 개 파일 오픈

|  |
| --- |
| [프로그램 소스]  #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h> // FILE 구조체에 대한 형 선언 포함  #include <stdlib.h> // exit() 함수 선언  int main() {  FILE \*ifp, \*ofp; // 파일 구조체 포인터 변수 선언  ifp = fopen("D:/Cprogram/source.txt", "r"); // source.txt 파일을 읽기전용으로 오픈  // 파일 개방에 실패하면 fopen()함수는 NULL 포인터 리턴  // 따라서 개방에 성공 했는지를 검사해야 함  if (ifp == NULL) {  printf("입력 파일 오픈 실패\n");  return 1;  }  printf("입력 파일이 오픈 되었습니다.\n");  // target.txt 파일을 쓰기전용으로 오픈  if (fopen\_s(&ofp, "D:/Cprogram/target.txt", "w") != 0) {  printf("출력 파일 오픈 실패\n");  exit(1); // 함수 강제 종료  }  printf("출력 파일이 오픈 되었습니다.\n");  return 0;  } |
| [실행 결과] |

1. 서식화 된 파일 입출력(구조체)

|  |
| --- |
| [프로그램 소스]  #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <stdlib.h>  typedef struct {  char name[20];  int mid, final;  }grade;  FILE \*f\_open(char \*, char \*);  int main() {  FILE\* fp;  grade temp;  int res;  fp = f\_open("D:/Cprogram/grade.txt", "r");  if (fp == NULL)  exit(1);  while (1) {  // fscnaf() 함수는 데이터의 입력이 끝나면 -1(EOF)을 리턴  res = fscanf(fp, "%s %d %d", temp.name, &temp.mid, &temp.final);  if (res == EOF)  break;    /\*  fscanf(fp, "%s %d %d", temp.name, &temp.mid, &temp.final);  if(feof(fp))  break;  \*/  fprintf(stdout, "%s %d %d %d\n", temp.name, temp.mid, temp.final, temp.mid + temp.final);  }  fp = f\_open("D:/Cprogram/grade.txt", "a"); // 추가모드로 오픈  fprintf(stdout, "이름과 성적(중간, 기말)을 입력하세요 >> ");  fscanf(stdin, "%s %d %d", temp.name, &temp.mid, &temp.final);  while (!feof(stdin)) {  fprintf(fp, "%s %d %d\n", temp.name, temp.mid, temp.final);  fscanf(stdin, "%s %d %d", temp.name, &temp.mid, &temp.final);  }  fclose(fp);  return 0;  }  FILE \*f\_open(char \*dir, char \*mode) {  FILE \*fp;  fp = fopen(dir, mode);  return fp;  } |
| [실행 결과] |

1. 이진 파일 입출력 – 파일을 사용하여 배열 원소 처리

|  |
| --- |
| [프로그램 소스]  #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <stdlib.h>  int main() {  char\* filename = "D:/Cprogram/myfile.bin";  double d[10] = { 10.23,19.87,1002.23,12.9,0.897,11.45,75.34,0.0,1.01,875.875 };  int i;  FILE \*fp;  if ((fp = fopen(filename, "wb")) == NULL) {  printf("cannot open file");  exit(1);  }  fwrite(d, sizeof(double), 10, fp); // 배열 원소 10개를 한꺼번에 이진파일에 저장  fclose(fp);  if ((fp = fopen(filename, "rb")) == NULL) {  printf("cannot open file");  exit(1);  }  for (i = 0; i < 10; i++)  d[i] = -1.0;  fread(d, sizeof(double), 10, fp);  fclose(fp);  for (i = 0; i < 10; i++)  printf("%d] %7.3lf\n", i + 1, d[i]);  printf("\n");  return 0;  } |
| [실행 결과] |

* **내용 점검**

1. 다음과 같은 변수 선언 문장을 가정하자. 다음 각각의 문장에서 문법 오류 및 논리 오류를 찾아 수정 하시오.

typedef struct personscore pscore;

pscore score;

char line[80];

FILE \*f;

int ch;

1. fread(&score, sizeof(pscore), 1, f);
2. if ((f = fopen("unit.txt", "rb")) == NULL) {

printf("파일이 열리지 않습니다.\n");

exit(1);

}

1. if ( fopen\_s(&f, "unit.txt", "rb") != 0 ){

printf("파일이 열리지 않습니다.\n");

exit(1);

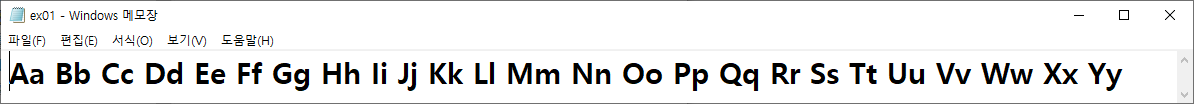
}

1. fwrite(&score, sizeof(pscore), 1, f);
2. fgets(line, 80, stdin);
3. fopen\_s(&f, "unit.txt", 'r');
4. fopen\_s(&f, "unit.txt", "r+");
5. fputc(ch, f);
6. ch = fgetc(f);
7. fclose(f);

* **프로그래밍 과제**

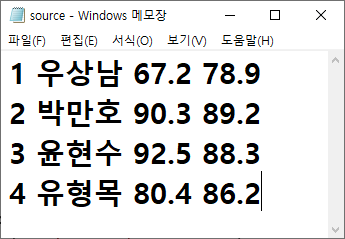
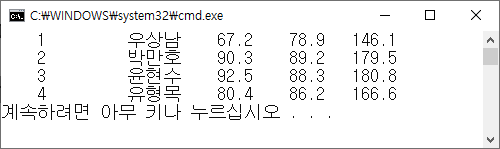
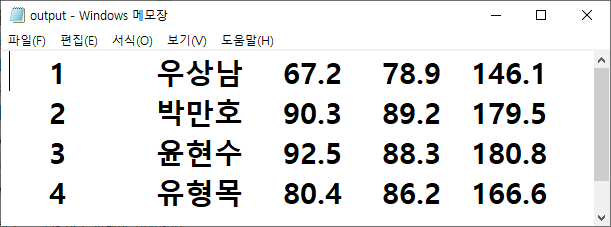
1. 다음 프로그램은 알파벳 대문자 A~Z, 소문자 a~z 까지 파일에 다음과 같은 결과를 출력하는 프로그램을 작성하려고 한다. 단일 문자 출력을 위해 fputc() 함수를 사용한다. 파일명은 “ex01.txt”에 작성되며, 작성된 파일을 열어보면 내용은 다음과 같다.

(대문자, 소문자 출력 후 공백 문자를 반복적으로 출력)



|  |
| --- |
| **[프로그램 소스]**  #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  int main() {  FILE \*fp;  char ch = 'A';  fp = fopen("D:/Cprogram/ex01.txt", "w");  // 파일 오픈 실패  if (fp == NULL) {  printf("입력 파일 오픈 실패\n");  return 1;  }    // 파일 오픈 성공  printf("입력 파일 오픈 성공\n");  while (ch <= 'Z') {  fputc(ch, fp); // 대문자 출력  fputc(ch + 32, fp); // 소문자 출력  fputc(' ', fp); // 공백 출력  ch++;  }  fclose(fp);  return 0;  } |
| **[실행 결과]** |

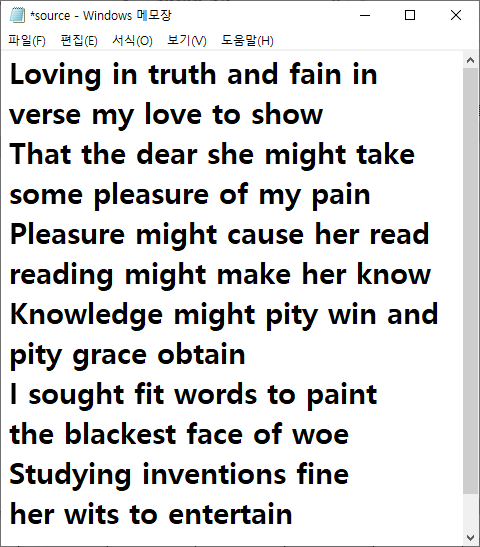
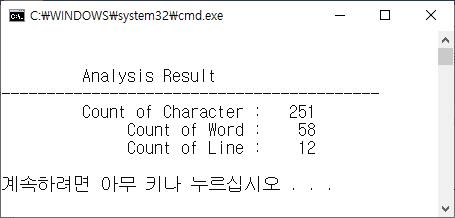
1. source.txt 파일에서 내용을 읽어 각 학생 점수의 합을 구하여 out.txt 파일에 아래의 내용과 합을 함께 출력하는 프로그램을 작성하시오. 단, 읽어온 데이터는 구조체를 사용하여 처리하며 파일 입출력 시 fscanf()함수와 fprintf() 함수를 사용하시오.

|  |
| --- |
| **[프로그램 소스]**  #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <stdlib.h>  typedef struct {  int number; // 번호  char name[20]; // 이름  double score1, score2; // 점수1, 점수2  }grade;  int main() {  FILE \*rp, \*wp;  grade data;  int res;  rp = fopen("D:/Cprogram/source.txt", "r"); // source.txt  wp = fopen("D:/Cprogram/out.txt", "w"); // out.txt  // 파일 오픈 실패  if (rp == NULL) {  printf("soure.txt 파일 오픈 실패\n");  exit(1);  }  else if (wp == NULL) {  printf("out.txt 파일 오픈 실패\n");  exit(1);  }  // 파일 오픈 성공  while (1) {  res = fscanf(rp, "%d %s %lf %lf", &data.number, data.name, &data.score1, &data.score2);  if (res == EOF)  break;  // 모니터에 출력  fprintf(stdout, "%5d %20s %10.1f %10.1f %10.1f\n", data.number, data.name, data.score1, data.score2, data.score1 + data.score2);    // out.txt에 쓰기  fprintf(wp, "%5d %20s %10.1f %10.1f %10.1f\n", data.number, data.name, data.score1, data.score2, data.score1 + data.score2);  }  fclose(rp);  fclose(wp);  return 0;  } |
| **[실행 결과]** |

1. 명령행 인자로 파일 이름을 입력하여 파일에 있는 문자 수, 단어 수, 라인 수를 출력하는 프로그램을 작성하시오. 단, 문자는 영어알파벳만을 의미하고 단어는 공백문자로 구분되는 단어로 한다.

source.txt에 저장된 내용은 다음과 같다.

|  |
| --- |
| **[프로그램 소스]**  #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <stdlib.h>  int main(int argc, char \*argv[]) {  FILE \*fp;  int res;  int ch\_cnt = 0, dan\_cnt = 0, line\_cnt = 0; // 문자 수, 단어 수, 라인 수  if (argc != 2) {  printf("사용법 : list filename\n");  exit(1);  }  fp = fopen(argv[1], "r");  if (fp == NULL) { // 오픈 실패  printf("파일 오픈 실패\n");  exit(1);  }  // 오픈 성공  while (1) {  res = fgetc(fp);  if (res == EOF)  break;    if ((res >= 'a' && res <= 'z') || (res >= 'A' && res <= 'Z')) // 문자  ch\_cnt++; // 문자 수  else if (res == '\n') { // 라인  line\_cnt++; // 라인 수  dan\_cnt++; // 단어 수  }  else if (res == ' ') { // 공백  dan\_cnt++; // 단어 수  }  }  fclose(fp);  printf("Analysis Result\n");  printf("-------------------------------------\n");  printf("Count of Character : %5d\n", ch\_cnt);  printf(" Count of Word : %5d\n", dan\_cnt);  printf(" Count of Line : %5d\n", line\_cnt);  return 0;  } |
| **[실행결과]** |