파일 크기만큼 읽기

1 #include <stdio.h>  
 2 #include <stdlib.h>  
 3 #include <string.h>  
 4   
 5 int main(void){  
 6 FILE \*fp;  
 7 char \*buffer;  
 8 int count, size;  
 9   
 10 fp = fopen("RegressionData.txt", "r");  
 11 fseek(fp, 0, SEEK\_END);  
 12 size = ftell(fp);  
 13 buffer = malloc(size+1);  
 14 memset(buffer, 0, size+1);  
 15   
 16 fseek(fp, 0, SEEK\_SET);  
 17 count = fread(buffer, size, 1, fp);  
 18 printf("%s size : %d, count : %d\n", buffer, size, count);  
 19 fclose(fp);  
 20 free(buffer);  
 21   
 22 return 0;  
 23 }

학생 이름, 수를 입력하면 파일에 저장하는 프로그램

#include <stdio.h>

Int main(void){

Char name[50];

Int marks, I, num;

Printf(“Enter number of students : “);

Scanf(“%d”, &num);

FILE \*fptr;

Fptr = (fopen(“C://student.txt”, “w”));

If(fptr == NULL){

Printf(“Error!”);

Exit(1);

}

For(I = 0; I < num; ++i){

Printf(“For student%d\nEnter name : “, i+1);

Scanf(“%s”, name);

Printf(“Enter marks : “);

Scanf(“%d”, &marks);

Fprintf(“fptr, “\nName: %s \nMarks=%d \n”, name, marks);

}

Fclose(fptr);

Return -;

}

파일크기 구하기

#define \_CRT\_SECURE\_NO\_WARNINGS // fopen 보안 경고로 인한 컴파일 에러 방지

#include <stdio.h> // fopen, fseek, ftell, fclose 함수가 선언된 헤더 파일

int main()

{

int size;

FILE \*fp = fopen("hello.txt", "r"); // hello.txt 파일을 읽기 모드(r)로 열기.

// 파일 포인터를 반환

fseek(fp, 0, SEEK\_END); // 파일 포인터를 파일의 끝으로 이동시킴

size = ftell(fp); // 파일 포인터의 현재 위치를 얻음

printf("%d\n", size); // 13

fclose(fp);

return 0;

}

파일에서 문자열 읽기

#define \_CRT\_SECURE\_NO\_WARNINGS // fopen 보안 경고로 인한 컴파일 에러 방지

#include <stdio.h> // fopen, fseek, ftell, fread, fclose 함수가 선언된 헤더 파일

#include <stdlib.h> // malloc, free 함수가 선언된 헤더 파일

#include <string.h> // memset 함수가 선언된 헤더 파일

int main()

{

char \*buffer;

int size;

int count;

FILE \*fp = fopen("hello.txt", "r"); // hello.txt 파일을 읽기 모드(r)로 열기.

// 파일 포인터를 반환

fseek(fp, 0, SEEK\_END); // 파일 포인터를 파일의 끝으로 이동시킴

size = ftell(fp); // 파일 포인터의 현재 위치를 얻음

buffer = malloc(size + 1); // 파일 크기 + 1바이트(문자열 마지막의 NULL)만큼 동적 메모리 할당

memset(buffer, 0, size + 1); // 파일 크기 + 1바이트만큼 메모리를 0으로 초기화

fseek(fp, 0, SEEK\_SET); // 파일 포인터를 파일의 처음으로 이동시킴

count = fread(buffer, size, 1, fp); // hello.txt에서 파일 크기만큼 값을 읽음

printf("%s size: %d, count: %d\n", buffer, size, count);

// Hello world! size: 13, count: 1: 파일의 내용, 파일 크기, 읽은 횟수 출력

fclose(fp); // 파일 포인터 닫기

free(buffer); // 동적 메모리 해제

return 0;

}

파일 부분적으로 읽고 쓰기

#define \_CRT\_SECURE\_NO\_WARNINGS // fopen 보안 경고로 인한 컴파일 에러 방지

#include <stdio.h> // fopen, fseek, fread, fclose 함수가 선언된 헤더 파일

#include <string.h> // memset 함수가 선언된 헤더 파일

int main()

{

char buffer[10] = { 0, };

FILE \*fp = fopen("hello.txt", "r"); // hello.txt 파일을 읽기 모드(r)로 열기.

// 파일 포인터를 반환

fseek(fp, 2, SEEK\_SET); // 파일 포인터를 파일 처음에서 2바이트만큼 순방향으로 이동시킴

fread(buffer, 3, 1, fp); // 3바이트만큼 읽음. 3바이트만큼 순방향으로 이동

printf("%s\n", buffer); // llo

memset(buffer, 0, 10); // 버퍼를 0으로 초기화

fseek(fp, 3, SEEK\_CUR); // 파일 포인터를 현재 위치에서 3바이트만큼 순방향으로 이동시킴

fread(buffer, 4, 1, fp); // 4바이트만큼 읽음. 4바이트만큼 순방향으로 이동

printf("%s\n", buffer); // orld

fclose(fp); // 파일 포인터 닫기

return 0;

}

기말고사

S2- 7

1 #include <stdio.h>  
 2   
 3 typedef struct{  
 4 int R, Im;  
 5 } Complex;  
 6   
 7 void add(Complex x, Complex y, Complex \*result);  
 8 void subtract(Complex x, Complex y, Complex \*result);  
 9 void multiply(Complex, Complex y, Complex \*result);  
 10 void divide(Complex x, Complex y, Complex \*result);  
 11   
 12 int main(void){  
 13 Complex x, y, result;  
 14 int co;  
 15 printf("Please enter the real and imaginary parts of x :");  
 16 scanf("%d %d", &x.R, &x.Im);  
 17 printf("Please enter the real and imaginary parts of y :");  
 18 scanf("%d %d", &y.R, &y.Im);  
 19 printf("Please enter choice(1=add, 2=subtract, 3=multiply, 4=divide) : ");  
 20 scanf("%d", &co);  
 21 if(co == 1) add(x, y, &result);  
 22 else if(co == 2) subtract(x, y, &result);  
 23 else if(co == 3) multiply(x, y, &result);  
 24 else if(co == 4) divide(x, y, &result);  
 25 return 0;  
 26 }  
 27 void add(Complex x, Complex y, Complex \*result){  
 28 printf("The result is %d + %di\n",(x.R + y.R), (x.Im + y.Im));  
 29 }  
 30 void subtract(Complex x, Complex y, Complex \*result){  
 31 printf("The result is %d + %di\n", (x.R - y.R), (x.Im - y.Im));  
 32 }  
 33 void multiply(Complex x, Complex y, Complex \*result){  
 34 printf("The result is %d + %di\n", (x.R \* y.R), (x.Im \* y.Im));  
 35 }

S2- 6 (로그인 문제)

#include <stdio.h>

int main()

{

MainMenu();

return 0;

}

void MainMenu()

{

int choice;

printf(" Welcome >> Please selecte chocie\n 1. Creat new account \n 2. Login as an admin\n 3. Login as an customer");

scanf("%d", &choice );

if(choice == 1)

{

CreatNewAcoount();

}

else if(choice == 3)

{

Login\_As\_Customer();

}

}

struct Details {

char UserName[8];

char Password [8];

char FirstName[8];

char LastName[8];

};

void CreatNewAcoount()

{

FILE \*file;

// Open the file for writing

file = fopen("Customer.txt", "wt");

if (!file)

{

printf("File could not be opened\n\a\a");

getchar();

return -1;

}

{

struct Details d1;

// Read data

printf("Enter first name: "); scanf("%s", &d1.FirstName);

printf("Enter last name: "); scanf("%s", &d1.LastName);

printf("Enter an user name: "); scanf("%s", &d1.UserName);

printf("Enter a password: "); scanf("%s", &d1.Password);

printf("The process has been done correctly");

printf("\n");

// Save data

fprintf(file, "%s\t%s\t%s\t%s\n", d1.FirstName, d1.LastName, d1.UserName, d1.Password);

}

// Close the file

fclose(file);

MainMenu();

}

void Login\_As\_Customer ()

{

char UserName[20];

char Password[10];

struct Details d1, d2;

printf("Enter username: "); scanf("%s",&UserName);

printf("Enter password: "); scanf("%s",&Password);

if (strcmp(UserName, d1.UserName) == 0 && strcmp(Password, d1.Password ) == 0) {

printf(" you have logged sucesfully");

}

else {

printf("Login Failed");

}

}