|  |
| --- |
|  |
| TEAM PROJECT DOCS |
|  |

|  |
| --- |
| 5WORKS |

**개발 버전**

|  |  |  |
| --- | --- | --- |
| **버전** | **내용** | **작성자** |
| **V1.0** | **주어진 팀정보를 조건에 따라**  **정렬하여 출력하는 프로그램 개발** | **이동준** |
| **V1.1** | **팀 정보가 담긴 CSV파일을 읽어와 구조체를 활용해 검색조건에 맞게**  **결과를 정렬하여 출력하는 프로그램 개발** | **조한슬** |
| **V1.2** | **구조체, 알고리즘 내용 추가** | **이동준** |
|  |  |  |
|  |  |  |

**구조체를 활용한 정렬 프로그램**

**1. 요구사항 분석**

* 서울반 교육생 명단\_한국표준협회.csv 파일 읽어오기
* 구조체를 활용하여 팀 정보를 구조체 변수로 선언한다.
* csv파일 상의 카테고리별로 검색이 가능해야 하며, 조장 검색시 해당 팀 정보 전체가 출력된다.
* 최종 검색 결과는 “result.txt” 파일에 검색시간과 함께 저장된다.

**2. 구조설계**

* 구조체 선언(typedef 사용)
* 개인 정보(조장 여부, 이름, 회사명, 이메일, 학교, 학과)를 배열로 저장
* file\_open(), change\_str(), arr\_delete(), arr\_token(), info\_save(), search() 함수를 선언한다. 각각 파일 입출력, 더미문자 바꾸기, 필요 없는 원소 삭제, 문자열 토큰화, 개인 정보를 배열에 저장, 입력한 단어로 개인 정보를 검색할 수 있는 기능을 수행한다.
* 검색시, 순차 탐색 알고리즘 활용
  + strcmp() 활용해서 검색 단어와 개인 정보 문자 비교

**3. 입출력 Data 정의 (Data Structure)**

* 구조체 배열을 사용해서 개인 정보(조장 여부, 이름, 회사명, 이메일, 학교, 학과)데이터를 저장
* 입출력시 fopen(), fprintf() 활용
* 출력시 구조체를 사용해 배열로 저장한 데이터 사용

**4. 알고리즘 설계 및 구현**

int file\_open();

입력할 파일을 호출하고, 문자열을 가공하고 출력할 수 있는 함수들을 호출한다.

void change\_str(char str[][1024], char find\_str);

찾고자 하는 문자를 배열에서 검색한 뒤, 해당 인덱스에 공백 문자를 넣는다.

void arr\_delete(char str[][1024], char find\_str);

찾고자 하는 문자를 배열에서 검색한 뒤, 해당 원소를 배열에서 삭제하고, 배열 크기를 하나 줄인다.

void arr\_token(char str[][1024], char find\_str);

찾고자 하는 문자를 기준으로 문자열을 자른다.

void info\_save(char str[][1024], team\_info info[24]);

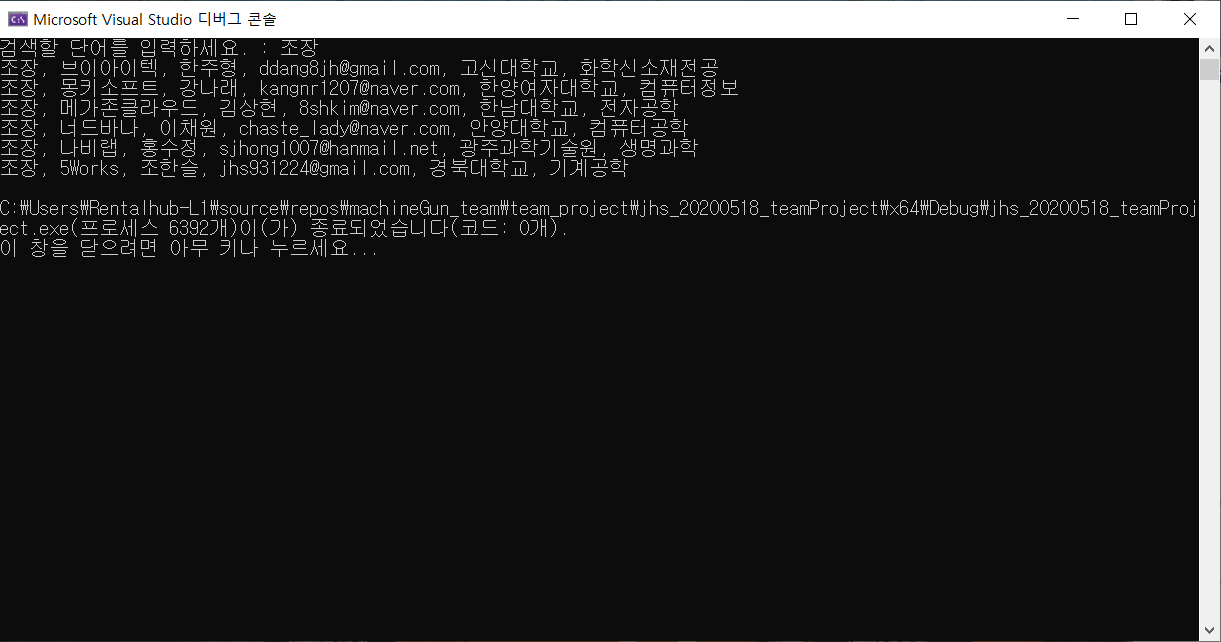
입력된 파일을 구조체 배열에 알맞게 저장한다.

void search(team\_info info[24]);

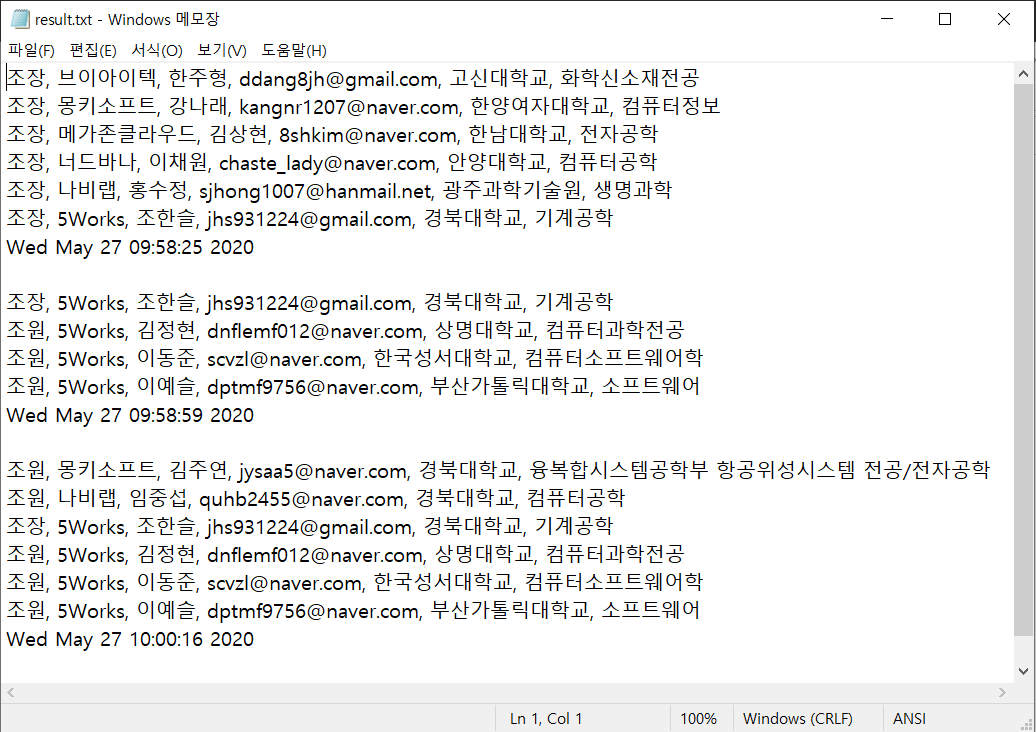
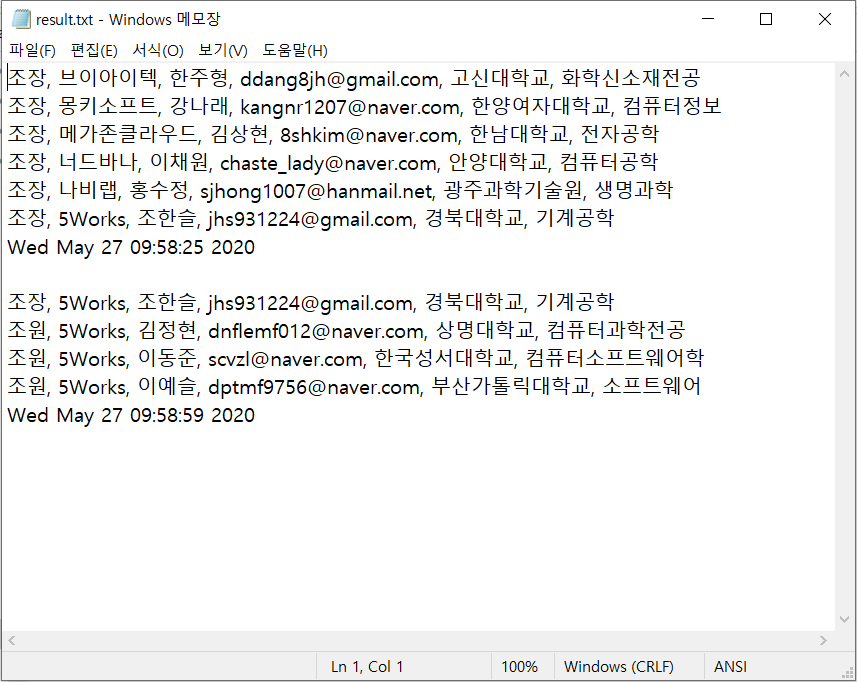
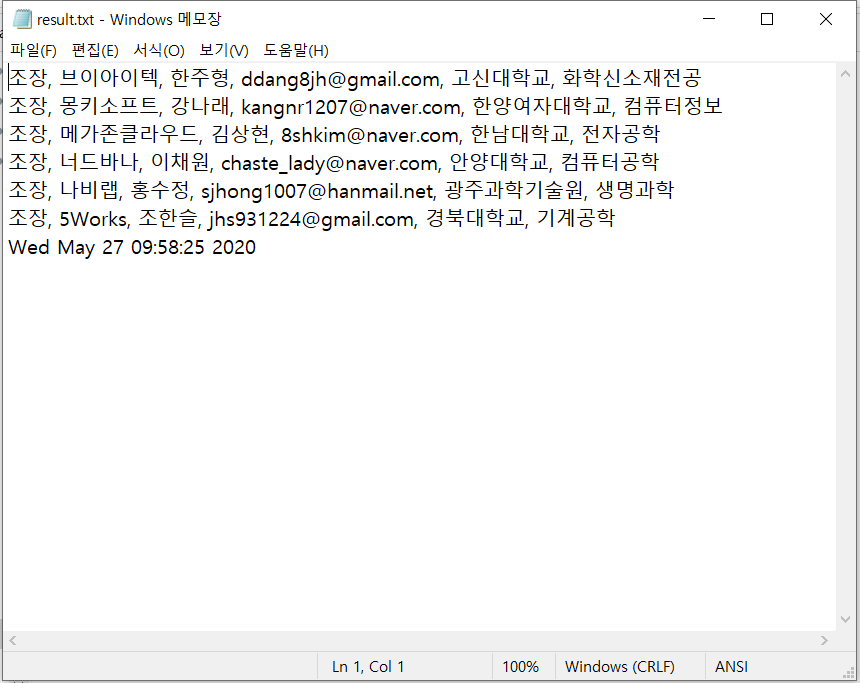
검색할 단어를 사용자로부터 입력받은 뒤, 구조체 배열에서 검색한 뒤 결과를 콘솔창과 텍스트 파일로 출력한다.

**5. 프로그램 실행결과**

1. 콘솔 출력



1. result.txt 검색결과 저장 (검색시간 포함)



**6. 구현 코드**

struct.h

|  |
| --- |
| #pragma once |
|  |  |
|  | typedef struct team\_info |
|  | { |
|  | char leader\_yn[20]; |
|  | char corp\_name[80]; |
|  | char name[80]; |
|  | char email[80]; |
|  | char univ[80]; |
|  | char major[80]; |
|  | }team\_info; |
|  |  |
|  | int file\_open(); |
|  | void change\_str(char str[][1024], char find\_str); |
|  | void arr\_delete(char str[][1024], char find\_str); |
|  | void arr\_token(char str[][1024], char find\_str); |
|  | void info\_save(char str[][1024], team\_info info[24]); |
|  | void search(team\_info info[24]); |
|  | void search\_team\_info(team\_info info[24], char str); |

main.c

|  |
| --- |
| #include <stdio.h> |
|  | #include <string.h> |
|  | #include "struct.h" |
|  |  |
|  | int main(void) |
|  | { |
|  |  |
|  | file\_open(); |
|  |  |
|  | return 0; |
|  | } |

algorithm.c

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS |
|  | #include <stdio.h> |
|  | #include <string.h> |
|  | #include "struct.h" |
|  | #include <memory.h> |
|  | #include <time.h> |
|  |  |
|  | int file\_open() |
|  | { |
|  | FILE\* ifp; |
|  | int ch; |
|  | char str[500][1024]; |
|  | int count = 0; |
|  | int info\_count = 0; |
|  | char\* word; |
|  | team\_info info[24]; |
|  |  |
|  | ifp = fopen("D:\\result\\서울반 교육생 명단\_한국표준협회.csv", "r"); |
|  |  |
|  | if (ifp == NULL) |
|  | { |
|  | printf("파일이 열리지 않았습니다.\n"); |
|  | return 1; |
|  | } |
|  |  |
|  | while (1) |
|  | { |
|  |  |
|  | word = fgets(str[count], 1024, ifp); |
|  |  |
|  | if (word == NULL) |
|  | { |
|  | break; |
|  | } |
|  |  |
|  | count++; |
|  |  |
|  | } |
|  |  |
|  | arr\_delete(str, '?'); |
|  | change\_str(str, ','); |
|  | arr\_token(str, ','); |
|  | arr\_delete(str, '\"'); |
|  | info\_save(str, info); |
|  | search(info); |
|  | fclose(ifp); |
|  |  |
|  | }; |
|  |  |
|  | void change\_str(char str[][1024], char find\_str) |
|  | { |
|  | int i, j; |
|  |  |
|  | for (i = 1; i < 425; i++) |
|  | { |
|  | for (j = 0; j < 1023; j++) |
|  | { |
|  | if (str[i][j] == find\_str) |
|  | { |
|  | if (str[i][j + 1] == find\_str) |
|  | { |
|  | str[i][j] = '\0'; |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  |  |
|  | } |
|  |  |
|  | void arr\_delete(char str[][1024], char find\_str) |
|  | { |
|  | int i, j; |
|  |  |
|  | for (i = 1; i < 425; i++) |
|  | { |
|  | for (j = 0; j < 1024; j++) |
|  | { |
|  | if (str[i][j] == find\_str) |
|  | { |
|  | memmove(str[i] + j, str[i] + j + 1, strlen(str[i]) - j); |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  | void arr\_token(char str[][1024], char find\_str) |
|  | { |
|  | int i, j; |
|  | int count = 0; |
|  |  |
|  | for (i = 1; i < 425; i++) |
|  | { |
|  | for (j = 0; j < 1024; j++) |
|  | { |
|  | if (str[i][0] == ',') |
|  | { |
|  | str[i][0] = '\n'; |
|  | } |
|  | else |
|  | { |
|  | if (str[i][j] == '\"') |
|  | { |
|  | count++; |
|  | if (count == 2) |
|  | { |
|  | count = 0; |
|  |  |
|  | } |
|  | } |
|  | else if (count == 0 && str[i][j] == find\_str) |
|  | { |
|  | str[i][j] = '\n'; |
|  | } |
|  | } |
|  |  |
|  | } |
|  | } |
|  | } |
|  |  |
|  | void info\_save(char str[][1024], team\_info info[24]) |
|  | { |
|  | int i; |
|  | int info\_count = 0; |
|  |  |
|  | for (i = 1; i < 425; i++) |
|  | { |
|  | if (str[i][0] != '\0') |
|  | { |
|  |  |
|  | int sc = 0; |
|  |  |
|  | char\* ptr = strtok(str[i], "\n"); |
|  | while (ptr != NULL) |
|  | { |
|  | switch (sc) |
|  | { |
|  | case 0: |
|  | if (str[i][0] == '\n') |
|  | { |
|  | strcpy(info[info\_count].leader\_yn, "조원"); |
|  | sc++; |
|  | } |
|  | else |
|  | { |
|  | strcpy(info[info\_count].leader\_yn, "조장"); |
|  | sc++; |
|  | break; |
|  | } |
|  | case 1: |
|  | strcpy(info[info\_count].corp\_name, ptr); |
|  | sc++; |
|  | break; |
|  | case 2: |
|  | strcpy(info[info\_count].name, ptr); |
|  | sc++; |
|  | break; |
|  | case 3: |
|  | strcpy(info[info\_count].email, ptr); |
|  | sc++; |
|  | break; |
|  | case 4: |
|  | strcpy(info[info\_count].univ, ptr); |
|  | sc++; |
|  | break; |
|  | case 5: |
|  | strcpy(info[info\_count].major, ptr); |
|  | sc++; |
|  | break; |
|  | } |
|  | ptr = strtok(NULL, "\n"); |
|  |  |
|  | } |
|  | info\_count++; |
|  | } |
|  | } |
|  | } |
|  |  |
|  | void search(team\_info info[24]) |
|  | { |
|  | char str[80]; |
|  | time\_t ltime; |
|  | char timebuf[26]; |
|  | time(&ltime); |
|  | ctime\_s(timebuf, 26, &ltime); |
|  |  |
|  | FILE\* ofp; |
|  | printf("검색할 단어를 입력하세요. : "); |
|  | scanf("%s", str); |
|  |  |
|  | ofp = fopen("D:\\result\\result.txt", "a"); |
|  |  |
|  | if (ofp == NULL) |
|  | { |
|  | printf("파일을 열지 못했습니다."); |
|  | } |
|  |  |
|  | int i; |
|  | for (i = 0; i < 24; i++) |
|  | { |
|  | if (strcmp(info[i].corp\_name, str) == 0) |
|  | { |
|  | printf("%s, ", info[i].leader\_yn); |
|  | fprintf(ofp, "%s, ", info[i].leader\_yn); |
|  | printf("%s, ", info[i].corp\_name); |
|  | fprintf(ofp, "%s, ", info[i].corp\_name); |
|  | printf("%s, ", info[i].name); |
|  | fprintf(ofp, "%s, ", info[i].name); |
|  | printf("%s, ", info[i].email); |
|  | fprintf(ofp, "%s, ", info[i].email); |
|  | printf("%s, ", info[i].univ); |
|  | fprintf(ofp, "%s, ", info[i].univ); |
|  | printf("%s\n", info[i].major); |
|  | fprintf(ofp, "%s\n", info[i].major); |
|  |  |
|  |  |
|  | } |
|  |  |
|  | else if (strcmp(info[i].name, str) == 0) |
|  | { |
|  | printf("%s, ", info[i].leader\_yn); |
|  | fprintf(ofp, "%s, ", info[i].leader\_yn); |
|  | printf("%s, ", info[i].corp\_name); |
|  | fprintf(ofp, "%s, ", info[i].corp\_name); |
|  | printf("%s, ", info[i].name); |
|  | fprintf(ofp, "%s, ", info[i].name); |
|  | printf("%s, ", info[i].email); |
|  | fprintf(ofp, "%s, ", info[i].email); |
|  | printf("%s, ", info[i].univ); |
|  | fprintf(ofp, "%s, ", info[i].univ); |
|  | printf("%s\n", info[i].major); |
|  | fprintf(ofp, "%s\n", info[i].major); |
|  |  |
|  | if (strcmp(info[i].leader\_yn, "조장") == 0) |
|  | { |
|  | int j; |
|  | for (j = i + 1; j < i + 4; j++) |
|  | { |
|  | printf("%s, ", info[j].leader\_yn); |
|  | fprintf(ofp, "%s, ", info[j].leader\_yn); |
|  | printf("%s, ", info[j].corp\_name); |
|  | fprintf(ofp, "%s, ", info[j].corp\_name); |
|  | printf("%s, ", info[j].name); |
|  | fprintf(ofp, "%s, ", info[j].name); |
|  | printf("%s, ", info[j].email); |
|  | fprintf(ofp, "%s, ", info[j].email); |
|  | printf("%s, ", info[j].univ); |
|  | fprintf(ofp, "%s, ", info[j].univ); |
|  | printf("%s\n", info[j].major); |
|  | fprintf(ofp, "%s\n", info[j].major); |
|  | } |
|  | } |
|  |  |
|  | } |
|  |  |
|  | else if (strcmp(info[i].major, str) == 0) |
|  | { |
|  | printf("%s, ", info[i].leader\_yn); |
|  | fprintf(ofp, "%s, ", info[i].leader\_yn); |
|  | printf("%s, ", info[i].corp\_name); |
|  | fprintf(ofp, "%s, ", info[i].corp\_name); |
|  | printf("%s, ", info[i].name); |
|  | fprintf(ofp, "%s, ", info[i].name); |
|  | printf("%s, ", info[i].email); |
|  | fprintf(ofp, "%s, ", info[i].email); |
|  | printf("%s, ", info[i].univ); |
|  | fprintf(ofp, "%s, ", info[i].univ); |
|  | printf("%s\n", info[i].major); |
|  | fprintf(ofp, "%s\n", info[i].major); |
|  |  |
|  | if (strcmp(info[i].leader\_yn, "조장") == 0) |
|  | { |
|  | int j; |
|  | for (j = i + 1; j < i + 4; j++) |
|  | { |
|  | printf("%s, ", info[j].leader\_yn); |
|  | fprintf(ofp, "%s, ", info[j].leader\_yn); |
|  | printf("%s, ", info[j].corp\_name); |
|  | fprintf(ofp, "%s, ", info[j].corp\_name); |
|  | printf("%s, ", info[j].name); |
|  | fprintf(ofp, "%s, ", info[j].name); |
|  | printf("%s, ", info[j].email); |
|  | fprintf(ofp, "%s, ", info[j].email); |
|  | printf("%s, ", info[j].univ); |
|  | fprintf(ofp, "%s, ", info[j].univ); |
|  | printf("%s\n", info[j].major); |
|  | fprintf(ofp, "%s\n", info[j].major); |
|  | } |
|  | } |
|  |  |
|  | } |
|  |  |
|  | else if (strcmp(info[i].univ, str) == 0) |
|  | { |
|  | printf("%s, ", info[i].leader\_yn); |
|  | fprintf(ofp, "%s, ", info[i].leader\_yn); |
|  | printf("%s, ", info[i].corp\_name); |
|  | fprintf(ofp, "%s, ", info[i].corp\_name); |
|  | printf("%s, ", info[i].name); |
|  | fprintf(ofp, "%s, ", info[i].name); |
|  | printf("%s, ", info[i].email); |
|  | fprintf(ofp, "%s, ", info[i].email); |
|  | printf("%s, ", info[i].univ); |
|  | fprintf(ofp, "%s, ", info[i].univ); |
|  | printf("%s\n", info[i].major); |
|  | fprintf(ofp, "%s\n", info[i].major); |
|  |  |
|  | if (strcmp(info[i].leader\_yn, "조장") == 0) |
|  | { |
|  | int j; |
|  | for (j = i + 1; j < i + 4; j++) |
|  | { |
|  | printf("%s, ", info[j].leader\_yn); |
|  | fprintf(ofp, "%s, ", info[j].leader\_yn); |
|  | printf("%s, ", info[j].corp\_name); |
|  | fprintf(ofp, "%s, ", info[j].corp\_name); |
|  | printf("%s, ", info[j].name); |
|  | fprintf(ofp, "%s, ", info[j].name); |
|  | printf("%s, ", info[j].email); |
|  | fprintf(ofp, "%s, ", info[j].email); |
|  | printf("%s, ", info[j].univ); |
|  | fprintf(ofp, "%s, ", info[j].univ); |
|  | printf("%s\n", info[j].major); |
|  | fprintf(ofp, "%s\n", info[j].major); |
|  | } |
|  | } |
|  |  |
|  | } |
|  |  |
|  | else if (strcmp(info[i].leader\_yn, str) == 0) |
|  | { |
|  | printf("%s, ", info[i].leader\_yn); |
|  | fprintf(ofp, "%s, ", info[i].leader\_yn); |
|  | printf("%s, ", info[i].corp\_name); |
|  | fprintf(ofp, "%s, ", info[i].corp\_name); |
|  | printf("%s, ", info[i].name); |
|  | fprintf(ofp, "%s, ", info[i].name); |
|  | printf("%s, ", info[i].email); |
|  | fprintf(ofp, "%s, ", info[i].email); |
|  | printf("%s, ", info[i].univ); |
|  | fprintf(ofp, "%s, ", info[i].univ); |
|  | printf("%s\n", info[i].major); |
|  | fprintf(ofp, "%s\n", info[i].major); |
|  |  |
|  |  |
|  | } |
|  |  |
|  | else if (strcmp(info[i].email, str) == 0) |
|  | { |
|  | printf("%s, ", info[i].leader\_yn); |
|  | fprintf(ofp, "%s, ", info[i].leader\_yn); |
|  | printf("%s, ", info[i].corp\_name); |
|  | fprintf(ofp, "%s, ", info[i].corp\_name); |
|  | printf("%s, ", info[i].name); |
|  | fprintf(ofp, "%s, ", info[i].name); |
|  | printf("%s, ", info[i].email); |
|  | fprintf(ofp, "%s, ", info[i].email); |
|  | printf("%s, ", info[i].univ); |
|  | fprintf(ofp, "%s, ", info[i].univ); |
|  | printf("%s\n", info[i].major); |
|  | fprintf(ofp, "%s\n", info[i].major); |
|  |  |
|  | if (strcmp(info[i].leader\_yn, "조장") == 0) |
|  | { |
|  | int j; |
|  | for (j = i + 1; j < i + 4; j++) |
|  | { |
|  | printf("%s, ", info[j].leader\_yn); |
|  | fprintf(ofp, "%s, ", info[j].leader\_yn); |
|  | printf("%s, ", info[j].corp\_name); |
|  | fprintf(ofp, "%s, ", info[j].corp\_name); |
|  | printf("%s, ", info[j].name); |
|  | fprintf(ofp, "%s, ", info[j].name); |
|  | printf("%s, ", info[j].email); |
|  | fprintf(ofp, "%s, ", info[j].email); |
|  | printf("%s, ", info[j].univ); |
|  | fprintf(ofp, "%s, ", info[j].univ); |
|  | printf("%s\n", info[j].major); |
|  | fprintf(ofp, "%s\n", info[j].major); |
|  | } |
|  | } |
|  |  |
|  | } |
|  | } |
|  | fprintf(ofp, "%s\n", timebuf); |
|  | fclose(ofp); |
|  | } |