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

|  |
| --- |
| 5WORKS |

**개발 버전**

|  |  |  |
| --- | --- | --- |
| **버전** | **내용** | **작성자** |
| **V1.0** |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**구조체를 활용한 정렬 프로그램**

**1. 요구사항 분석**

* 주어진 팀의 정보(팀아이디, 팀이름, 회사명)를 활용하여, 각각의 조건에 맞게 정렬한다.
* 구조체를 활용하여 팀 정보를 구조체 변수로 선언한다.
* 프로그램 실행시 숫자를 활용해 메뉴를 선택하며, 종료를 선택할 때 까지 반복 실행한다.

**2. 구조설계**

* 구조체 선언(typedef 사용)
* 팀 정보(팀 아이디, 팀 이름, 회사명)를 배열로 저장
* 팀 아이디, 팀 이름, 회사명 각각의 조건으로 정렬하는 함수 선언
* 정렬시, 버블정렬 활용
  + 숫자는 단순 크기비교, 문자는 strcmp() 활용

**3. 입출력 Data 정의 (Data Structure)**

* 구조체 사용해서 팀 정보를 저장
* 출력시 구조체를 사용해 배열로 저장한 데이터 사용
* 입출력시 fopen(), fprintf() 활용

**4. 알고리즘 설계 및 구현**

void select\_menu(team\_info\* list, int list\_len);

사용자가 메뉴를 선택할 수 있도록 화면을 구성한 뒤, 사용자가 선택한 메뉴에 따른 함수 호출을 할 수 있도록 switch문을 사용, 종료시 exit(0) 함수 사용

void sortByTeamId(team\_info\* list, int list\_len);

팀 번호순으로 정렬. 버블 정렬을 사용해서 오름차순으로 정렬을 한다.

void sortByTeamName(team\_info\* list, int list\_len);

팀 이름순으로 정렬. Strcmp() 함수를 사용해서 사전순으로 정렬을 한다.

void sortByCorpName(team\_info\* list, int list\_len);

회사명으로 정렬. Strcmp() 함수를 사용해서 사전순으로 정렬을 한다.

void print\_list(team\_info\* list, int list\_len);

사용자가 선택한 메뉴로 정렬을 한 뒤, 메모장에 입력 하고, 출력을 한다.

**5. 코딩 및 테스트**

struct.h

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

typedef struct team\_info

{

int team\_id;

char team\_name[80];

char corp\_name[80];

}team\_info;

void select\_menu(team\_info\* list, int list\_len);

void sortByTeamId(team\_info\* list, int list\_len);

void sortByTeamName(team\_info\* list, int list\_len);

void sortByCorpName(team\_info\* list, int list\_len);

void print\_list(team\_info\* list, int list\_len);

main.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

#include "struct.h"

int main(void)

{

team\_info list[6] = {

{1, "horangnavi", "navilab"},

{2, "machinegun", "fiveworks"},

{3, "megaton", "megazone"},

{4, "dd", "vitec"},

{5, "banana", "monkeysoft"},

{6, "park123", "nudbana"},

};

int list\_len = sizeof(list) / sizeof(list[0]);

while (1) {

select\_menu(list, list\_len);

}

return 0;

}

algorithm.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

#include "struct.h"

void select\_menu(team\_info\* list, int list\_len)

{

int menu\_num;

printf("==== 메뉴를 선택해 주세요.====\n");

printf("==== (예시 : 1 Enter ) ====\n");

printf("==== 1. 팀아이디 순 정렬 =====\n");

printf("==== 2. 팀 이름 순 정렬 =====\n");

printf("==== 3. 회사이름 순 정렬 =====\n");

printf("==== 4. 종 료 =====\n");

printf("==============================\n");

scanf\_s("%d", &menu\_num);

printf("\n");

switch

(menu\_num)

{

case 1:

printf("팀 아이디 순 정렬입니다.\n");

sortByTeamId(list, list\_len);

break;

case 2:

printf("팀 이름 순 정렬입니다.\n");

sortByTeamName(list, list\_len);

break;

case 3:

printf("회사이름 순 정렬입니다.\n");

sortByCorpName(list, list\_len);

break;

case 4:

printf("프로그램을 종료합니다.\n");

exit(0);

break;

default:

break;

}

return;

}

void sortByTeamId(team\_info\* list, int list\_len)

{

int i, j;

team\_info tmp;

for (i = 0; i < list\_len; i++)

{

for (j = 0; j < list\_len - i - 1; j++)

{

if (list[j].team\_id > list[j + 1].team\_id)

{

tmp = list[j];

list[j] = list[j + 1];

list[j + 1] = tmp;

}

}

}

print\_list(list, list\_len);

}

void sortByTeamName(team\_info\* list, int list\_len)

{

int i, j;

team\_info tmp;

for (i = 0; i < list\_len; i++)

{

for (j = 0; j < list\_len - i - 1; j++)

{

if (strcmp(list[j].team\_name, list[j + 1].team\_name) > 0)

{

tmp = list[j];

list[j] = list[j + 1];

list[j + 1] = tmp;

}

}

}

print\_list(list, list\_len);

}

void sortByCorpName(team\_info\* list, int list\_len)

{

int i, j;

team\_info tmp;

for (i = 0; i < list\_len; i++)

{

for (j = 0; j < list\_len - i - 1; j++)

{

if (strcmp(list[j].corp\_name, list[j + 1].corp\_name) > 0)

{

tmp = list[j];

list[j] = list[j + 1];

list[j + 1] = tmp;

}

}

}

print\_list(list, list\_len);

}

void print\_list(team\_info\* list, int list\_len)

{

int i;

FILE\* fp;

fp = fopen("result.txt", "a+");

if (fp == NULL)

{

printf("파일을 만들지 못했습니다.\n");

return 1;

}

for (i = 0; i < list\_len; i++)

{

fprintf(fp, "%-5d %-20s\t %-20s\n", (list + i)->team\_id, (list + i)->team\_name, (list + i)->corp\_name);

}

fprintf(fp, "%s %s\n\n", \_\_DATE\_\_, \_\_TIME\_\_);

fflush(fp);

fclose(fp);

for (i = 0; i < list\_len; i++)

{

printf("%-5d %-20s\t %-20s\n", (list + i)->team\_id, (list + i)->team\_name, (list + i)->corp\_name);

}

printf("\n");

}