

Problem Solving Methods

[Team-1]

202235512 정은섭

202235143 최정원

202035510 김유현

202135750 김지현

202135762 김희준

Team introduction & Contribution percentage

Name	Role	Contribution Percentage
202235512 정은섭	P4-1 code/png, P6-1 code/png, P6-2 code/png, P7-1 code/png, P7-2 code/png, code-merging	25%
202235143 최정원	P3-1 code/png, P4-1 code/png, P7-1 code/png	25%
202035510 김유현	P1-1 code/png, P1-2 code/png, P2-1 code/png, P2-2 code/png, make ppt	25%
202135750 김지현		0%
202135762 김희준	P5-1 code/png, P8-1 code/png, P8-2 code/png	25%

Achievement Table

Problem	Members involved	Achievement
P1-1, P1-2	정은섭, 김유현	100%
P2-1, P2-2	김유현	100%
P3-1	최정원	100%
P4-1	정은섭, 최정원	100%
P5-1	김희준	100%
P6-1, P6-2	정은섭	80%
P7-1, P7-2	정은섭, 최정원	80%
P8-1, P8-2	김희준	100%

Problem Solving Steps – Step 1

- Step 1. Understand the problem

- 1) Find a person whose name contains "Choi"
- 2) Print info of a person who satisfies conditions
- 3) Find a person whose university is "Gachon University"
- 4) Print info of a person who satisfies conditions
- 5) Sort the data in the array in tag# order
- 6) Sort the data in the array by age using the selection sort and write it in the file.
- 7) Remove the data about "Choi"
- 8) Add the data about "Paik"
- 9) After extracting the most recent 10 data, write the data in 'P8-1.txt' after calculating the checksum based on the name (char[25]).
- 10) After copying P8-1.txt to P8-2.txt, calculate checksum based on name (char[25]) of the data in 'P8-2.txt'.
- 11) Compare whether the original checksum and copy checksum are the same.

Problem Solving Steps – Step 2

- Step 2. Outline a solution
 - Make up a search string "Choi", "Gachon University"
 - Search condition: Find "Choi", "Gachon University"
 - Search each array and linked list if the person data satisfy the search condition("Choi", "Gachon University")
 - Print the person data for each person who satisfy all the search condition("Choi", "Gachon-University")
 - Sort the data according to the tag number
 - Make the linked list with sorted data in tag
 - Read the recent array and sort the data by age using selection sort.
 - And store the sorted in PERSON sort_age[MAX].
 - Find "Choi" and canceled registration in array, linked list and sort the array.
 - Find tag number not exist in array and put the num for "Paik".
 - Read the most recent 10 data in the recent array and calculate checksum based on temp.name[].
 - Write the data in txt file by using Write_File() and Copy data in another file.
 - Calculate copy data's checksum by using checksum() and compare org_checkSum, cpy_checkSum.

Problem Solving Steps – Step 3

- Step 3. Form a program structure

- Read data("registration_txt") and divide the list with "/"
- Search the linked list for persons whose university is "Gachon-university"
- Print the linked list for persons whose university is "Gachon-university"
- Sort the data according to the tag number
- Make the linked list with sorted data in tag
- Arr_head[] containing the current data is stored in the sort_age[] array after selection sort.
- Using find() to find "Choi" and count and del() to delete the array data and swap() to change order.
- Using deleteNode() to delete node and cmp() to compare the name "Choi".
- Using find_tag_arr() and find_tag_list() to find tag number that not exist in array and linked list and put the number for "Paik".
- After storing the most recent 10 data in most_Recent[], calculate org_checksum.
- Write the data of most_Recent[] on P8-1.txt and copy P8-1.txt to P8-2.txt using Read_File() Write_File(). Then calculate cpy_checkSum and compare it with org_checkSum.

Problem Solving Steps – Step 4

- Step 4. Write a Pseudo code

```
step 4 - 메모장
파일 편집 보기
add to the list
if(NULL)
while()
return
read list
read file
use strtok to divide data with "/"
read addList
close file
print from the list
if("Gachon University")
print the persons data
```

```
//p3
for(i=0; i<29; i++){
for(j=0; j<29-i; j++){
if(data[j].tag > data[j+1].tag){
temp[j] = data[j];
data[j] = data[j+1];
data[j+1] = temp[j];
}
}
}
//p4
for(i = 0; i < arr count; i++) {
a(struct pointer) -> all member except next = arr[i] all member
a->next = NULL
add_list(head, a)
```

```
PS-1 pseudo code.txt - 메모장
파일 편집 보기
// PS-1 pseudo code
void selection_sort()
{
for (int i = 0; i < numberOfPersons; i++) {
Copy_Element();
}
for (index = 0; index < numberOfPersons - 1; index++) {
least = index;
for (int t = index + 1; t < numberOfPersons; t++) {
if (age[t] < age[least]) {
least = t;
}
}
if (index != least) {
temp = age[index];
age[index] = age[least];
age[least] = temp;
}
}
}
```

```
//p6-1
for(i=0; i<arr count; i++){
compare the arr and "Choi"
if (strstr(arr, "Choi") != 0)
arr[i] == NULL;
}
//p6-2
head = linked list
while (head != NULL) {
cnode = head;
if(strstr(cnode->name, "Choi")) != 0) break;
pnode = head;
cnode = cnode->next;
}
pnode->next = cnode->next;
}
```

```
//p7-1
loop for find tag not exist in data
head[] is arr list
for (i=30; i>0; i--) {
for (j=0; j<arr count; j++) {
if (i == head[j].tag) break;
}
if(j == arr count) return i;
}
arr[i+1] = Paik member
//p7-2
loop for find tag not exist in data
head = linked list
for (i=30; i>0; i--) {
a = head;
while (a != NULL) {
if(a->tag == i) break;
a = a->next;
}
if(a == NULL) return i;
}
put i in Paik's tag
and addlist Paik
```

```
PS-1, PS-2 pseudo code.txt - 메모장
파일 편집 보기
// PS-1, PS-2 pseudo code
int CheckSum()
{
int XOR = 0; // checksum
for (int i = 0; i < numberOfData; i++) {
Copy_Element();
int k = 0;
while (temp.name[k] != 'W0') {
XOR ^= temp.name[k];
k++;
}
}
return XOR;
}
```

P1-1 Code & Result screenshots

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <string.h>
5  #define MAX 35
6
7  typedef struct PERSON {
8      int tag;
9      char date[100];
10     char fee[100];
11     char name[25];
12     int age;
13     char univ[35];
14     char job[15];
15 } PERSON;
16
17 int read_data(struct PERSON head[], char* myFile) {
18     FILE* myinFile;
19     int i = 0;
20     myinFile = fopen(myFile, "r");
21     if (myinFile == NULL) {
22         printf("File could not be opened\n");
23     }
24
25     char line[256];
26     while ((fgets(line, 256, myinFile)) != NULL) {
27         char* tok;
28         tok = strtok(line, "/");
29         head[i].tag = atoi(tok);
30
31         tok = strtok(NULL, "/");
32         strcpy(head[i].date, tok);
33
34         tok = strtok(NULL, "/");
35         strcpy(head[i].fee, tok);
36
37         tok = strtok(NULL, "/");
38         strcpy(head[i].name, tok);
39
40         tok = strtok(NULL, "/");
41         head[i].age = atoi(tok);
42
43         tok = strtok(NULL, "/");
44         strcpy(head[i].univ, tok);
45     }
```

```
46     tok = strtok(NULL, "\n");
47     strcpy(head[i].job, tok);
48     i++;
49 }
50 fclose(myinFile);
51 return i;
52 }
53
54 void print_list(struct PERSON head[], int n) {
55     int i;
56
57     for (i = 0; i < n; i++) {
58         if (strstr(head[i].name, "Choi") != 0)
59             //if (strcmp(head[i].univ, "Gachon University") == 0)
60             {
61                 printf("%d/%s/%s/%s/%d/%s/%s\n", head[i].tag, head[i].date, head[i].fee, head[i].name, head[i].age, head[i].univ, head[i].job);
62             }
63     }
64 }
65
66 int main() {
67     char fileName[] = "registration_data.txt";
68     struct PERSON head[MAX];
69
70     int count = 0;
71
72     count = read_data(head, fileName);
73     print_list(head, count);
74
75     return 0;
76 }
77
78 }
```

Microsoft Visual Studio 디버그

11/2022-07-22/no/Kwangsui Choi/48/Seoul National University/marketer
15/2022-07-12/no/Tongbang Choi/26/Cornell University/engineer
1/2022-08-25/yes/Jihu Choi/74/Harvard University/engineer
30/2022-07-13/yes/Kyungmin Choi/44/Duke University/student
2/2022-08-22/no/Seungmin Choi/31/Gachon University/staff

C:\Users\morri\OneDrive\문서\대학(1-2)\문제해결기법(화7, 목12)\강의 자료\입 프로젝트\2-1\x64\Debug\2-1.exe(프로세스 2462 4개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용 하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

P1-2 Code & Result screenshots

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <string.h>
5  #define MAX 35
6
7  typedef struct PERSON{
8      int tag;
9      char date[100];
10     char fee[100];
11     char name[25];
12     int age;
13     char univ[35];
14     char job[15];
15     struct PERSON* next;
16 }PERSON;
17
18 struct PERSON *add_list(struct PERSON* head, struct PERSON* ptr){
19     struct PERSON* p;
20
21     if (head == NULL){
22         head = ptr;
23         return head;
24     }
25     p = head;
26     while (p->next) {
27         p = p->next;
28     }
29     p->next = ptr;
30
31     return head;
32 }
33
34 struct PERSON *read_data(struct PERSON* head, char* myFile){
35     FILE* myinFile;
36
37     myinFile = fopen(myFile, "r");
38     if (myinFile == NULL){
39         printf("File could not be opened\n");
40     }
41 }
```

```
42 char line[256];
43 while ((fgets(line, 256, myinFile)) != NULL) {
44     struct PERSON* ptr = (struct PERSON*)malloc(sizeof(struct PERSON));
45     ptr->next = NULL;
46     char* tok;
47
48     tok = strtok(line, "/");
49     ptr->tag = atoi(tok);
50
51     tok = strtok(NULL, "/");
52     strcpy(ptr->date, tok);
53
54     tok = strtok(NULL, "/");
55     strcpy(ptr->fee, tok);
56
57     tok = strtok(NULL, "/");
58     strcpy(ptr->name, tok);
59
60     tok = strtok(NULL, "/");
61     ptr->age = atoi(tok);
62
63     tok = strtok(NULL, "/");
64     strcpy(ptr->univ, tok);
65
66     tok = strtok(NULL, "\n");
67     strcpy(ptr->job, tok);
68     head = add_list(head, ptr);
69 }
70 fclose(myinFile);
71 return head;
72
73 void print_list(struct PERSON* head){
74     struct PERSON *person = head;
75     while (person){
76         if (strstr(person->name, "Choi") != 0 ){
77             //if (strcmp(person->univ, "Gachon University") == 0) {
78                 printf("Id/%s/%s/%s/%s/%s\n", person->tag, person->date, person->fee, person->name, person->age, person->univ, person->job);
79             }
80         person = person->next;
81     }
82 }
83
84 int main() {
85     char fileName[] = "registration_data.txt";
86     struct PERSON* head = NULL;
87     head = read_data(head, fileName);
88     print_list(head);
89     return 0;
90 }
```

```
29/2022-06-08/yes/Bailey Houghton/31/Gachon University/engineer
12/2022-07-22/no/Owen Martin/66/Gachon University/engineer
8/2022-06-04/no/Moises Barlow/57/Gachon University/engineer
14/2022-08-15/yes/Kwangsu Cho/48/Gachon University/executive
27/2022-08-24/no/Monner French/42/Gachon University/professor
17/2022-08-14/no/Chunyoung Chang/75/Gachon University/student
2/2022-08-22/no/Seungmin Choi/31/Gachon University/staff
13/2022-06-03/yes/Chinho Cho/68/Gachon University/student
20/2022-07-30/yes/Chinho Kim/52/Gachon University/engineer
```

C:\Users\morri\OneDrive\문서\대학(1-2)\문제해결기법(과7, 목12)\강의 자료\팀 프로젝트\2-1\x64\Debug\2-1.exe(프로세스 6872)가(가) 종료되었습니다(코드: 0x0).

디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.

이 점을 닫으려면 아무 키나 누르세요...

P2-1 Code & Result screenshots

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <string.h>
5  #define MAX 35
6
7  typedef struct PERSON {
8      int tag;
9      char date[100];
10     char fee[100];
11     char name[25];
12     int age;
13     char univ[35];
14     char job[15];
15 } PERSON;
16
17 int read_data(struct PERSON head[], char* myFile) {
18     FILE* myinFile;
19     int i = 0;
20     myinFile = fopen(myFile, "r");
21     if (myinFile == NULL) {
22         printf("File could not be opened\n");
23     }
24
25     char line[256];
26     while ((fgets(line, 256, myinFile)) != NULL) {
27         char* tok;
28         tok = strtok(line, "/");
29         head[i].tag = atoi(tok);
30
31         tok = strtok(NULL, "/");
32         strcpy(head[i].date, tok);
33
34         tok = strtok(NULL, "/");
35         strcpy(head[i].fee, tok);
36
37         tok = strtok(NULL, "/");
38         strcpy(head[i].name, tok);
39
40         tok = strtok(NULL, "/");
41         head[i].age = atoi(tok);
42
43         tok = strtok(NULL, "/");
44         strcpy(head[i].univ, tok);
45     }
```

```
46     tok = strtok(NULL, "\n");
47     strcpy(head[i].job, tok);
48     i++;
49 }
50 fclose(myinFile);
51 return i;
52
53 void print_list(struct PERSON head[], int n) {
54     int i;
55     for (i = 0; i < n; i++) {
56         //if(strstr(head[i].name, "Choi") != 0)
57         if (strcmp(head[i].univ, "Gachon University") == 0)
58         {
59             printf("%d/%s/%s/%s/%s/%s\n", head[i].tag, head[i].date, head[i].fee, head[i].name, head[i].age, head[i].univ, head[i].job);
60         }
61     }
62 }
63
64 int main() {
65     char fileName[] = "registration_data.txt";
66     struct PERSON head[MAX];
67
68     int count = 0;
69
70     count = read_data(head, fileName);
71     print_list(head, count);
72
73     return 0;
74 }
```

Microsoft Visual Studio 디버그 콘솔

```
11/2022-07-22/no/Kwangsu Choi/48/Seoul National University/marketer
15/2022-07-12/no/Tongbang Choi/26/Cornell University/engineer
1/2022-08-25/yes/Jihu Choi/74/Harvard University/engineer
30/2022-07-13/yes/Kyungmin Choi/44/Duke University/student
2/2022-08-22/no/Seungmin Choi/31/Gachon University/staff
```

C:\Users\morri\OneDrive\문서\대학(1-2)\문제해결기법(외7, 목12)\강의 자료\팀 프로젝트\Team-project\x64\Debug\Team-project.exe(프로세스 6652개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...

P2-2 Code & Result screenshots

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #define MAX 35
6
7 typedef struct PERSON{
8     int tag;
9     char date[100];
10    char fee[100];
11    char name[25];
12    int age;
13    char univ[35];
14    char job[15];
15    struct PERSON* next;
16 }PERSON;
17
18 struct PERSON *add_list(struct PERSON* head, struct PERSON* ptr){
19     struct PERSON* p;
20
21     if (head == NULL){
22         head = ptr;
23         return head;
24     }
25     p = head;
26     while (p->next) {
27         p = p->next;
28     }
29     p->next = ptr;
30
31     return head;
32 }
33
34 struct PERSON *read_data(struct PERSON* head, char* myInFile){
35     FILE* myInFile;
36
37     myInFile = fopen(myFile, "r");
38     if (myInFile == NULL){
39         printf("File could not be opened\n");
40     }
41 }
```

```
Microsoft Visual Studio 디버그
29/2022-06-08/yes/Bailey Houghton/31/Gachon University/engineer
12/2022-07-22/no/Owen Martin/66/Gachon University/engineer
8/2022-06-04/no/Moises Barlow/57/Gachon University/engineer
14/2022-08-15/yes/Kwangsue Cho/48/Gachon University/executive
27/2022-08-24/no/Konner French/42/Gachon University/professor
17/2022-08-14/no/Chunyoung Chang/75/Gachon University/student
2/2022-08-22/no/Seungmin Choi/31/Gachon University/staff
13/2022-06-03/yes/Chinho Cho/68/Gachon University/student
28/2022-07-30/yes/Chinho Kim/52/Gachon University/engineer

C:\Users\morri\OneDrive\문서\대학(1-2)\문제해결기법(과7, 목12)\강의 자료\팀 프로젝트\Team-project\x64\Debug\Team-project
.exe(프로세스 26788개)에(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

char line[256];
while ((fgets(line, 256, myInFile)) != NULL) {
    struct PERSON* ptr = (struct PERSON*)malloc(sizeof(struct PERSON));
    ptr->next = NULL;
    char* tok;

    tok = strtok(line, "/");
    ptr->tag = atoi(tok);

    tok = strtok(NULL, "/");
    strcpy(ptr->date, tok);

    tok = strtok(NULL, "/");
    strcpy(ptr->fee, tok);

    tok = strtok(NULL, "/");
    strcpy(ptr->name, tok);

    tok = strtok(NULL, "/");
    ptr->age = atoi(tok);

    tok = strtok(NULL, "/");
    strcpy(ptr->univ, tok);

    tok = strtok(NULL, "\n");
    strcpy(ptr->job, tok);
    head = add_list(head, ptr);
}

fclose(myInFile);
return head;

void print_list(struct PERSON* head){
    struct PERSON *person = head;
    while (person){
        //if(strstr(person->name, "Choi") != 0){
        if (strcmp(person->univ, "Gachon University") == 0){
            printf("%d/%s/%s/%s/%d/%s/%s\n", person->tag, person->date, person->fee, person->name, person->age, person->univ, person->job);
        }
        person = person->next;
    }
}

int main() {
    char fileName[] = "registration_data.txt";
    struct PERSON* head = NULL;
    head = read_data(head, fileName);
    print_list(head);
    return 0;
}
```

P3-1 Code

```
Project33
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <string.h>
5
6  struct Data {
7      int tag;
8      int year;
9      int month;
10     int day;
11     char fee_paid[5];
12     char name[20];
13     int age;
14     char org[30];
15     char job[15];
16     struct Data* next;
17 };
18
19
20 void tag_sort(struct Data data[]) { //태그
21     FILE* inFile;
22     if ((inFile = fopen("registraion_data.txt", "r")) == NULL) {
23         printf("File Could Not Be Opened");
24         exit(1);
25     }
26     int i = 0;
27
28     FILE* tagFile = fopen("tagdata.txt", "w");
29
30     while ((fscanf(inFile, "%d/%d-%d-%d/%s/%s/%d/%s/%s", &data[i].tag, &data[i].year, &data[i].month, &data[i].day, data[i].fee_paid, data[i].name, &data[i].age, data[i].org, data[i].job)) != EOF) {
31         i++;
32     }
33
34     struct Data temp[30];
35     for (int i = 0; i < 29; i++)
36     {
37         for (int j = 0; j < 29 - i; j++)
38         {
39             if (data[j].tag > data[j + 1].tag)
40             {
41                 temp[i] = data[j];
42                 data[j] = data[j + 1];
43                 data[j + 1] = temp[i];
44             }
45         }
46     }
47
48     for (int i = 0; i < 30; i++)
49     {
50         fprintf(tagFile, "%d/%d-%d-%d/%s/%s/%d/%s/%s\n", data[i].tag, data[i].year, data[i].month, data[i].day, data[i].fee_paid, data[i].name, data[i].age, data[i].org, data[i].job);
51         printf("%d/%d-%d-%d/%s/%s/%d/%s/%s\n", data[i].tag, data[i].year, data[i].month, data[i].day, data[i].fee_paid, data[i].name, data[i].age, data[i].org, data[i].job);
52     }
53 }
```

P3-1 Code & Result screenshots

```
53     fclose(inFile);
54     fclose(tagFile);
55 }
56
57
58 void printdata(struct Data* head) {
59     struct Data* ptr = head;
60     while (ptr != NULL) {
61         printf("%d/%d-%02d-%02d/%s/%s/%d/%s/%s\n", ptr->tag, ptr->year, ptr->month, ptr->day, ptr->fee_paid, ptr->name, ptr->age, ptr->org, ptr->job);
62         ptr = ptr->next;
63     }
64 }
65
66 void main() {
67     struct Data data[30];
68     int count = 0;
69     tag_sort(data);
70
71     struct Data* head = NULL;
72     head = &data[0];
73     for (int i = 0; i < 30; i++) {
74         data[i].next = NULL;
75     }
76     for (int i = 0; i < 29; i++) {
77         data[i].next = &data[i + 1];
78     }
79     printdata(head);
80 }
```

Microsoft Visual Studio 디버그 콘솔

```
1/2022-08-25/yes/Jihu Choi/74/Harvard University/engineer
2/2022-08-22/no/Seungmin Choi/31/Gachon University/staff
3/2022-07-01/no/Chinho Park/53/Peking University/engineer
4/2022-07-03/no/Jihu Cho/71/Tsinghua University/engineer
5/2022-06-12/yes/Chunyong Park/48/University of Cambridge/student
6/2022-06-04/yes/Bobby Anderson/33/McGill University/engineer
7/2022-06-28/yes/Jihu Park/70/Australian National Universitystudent/studen
8/2022-06-04/no/Moises Barlow/57/Gachon University/engineer
9/2022-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
10/2022-06-06/yes/William Cohen/37/University of Cambridge/engineer
11/2022-07-22/no/Kwangsu Choi/48/Seoul National University/marketer
12/2022-07-22/no/Owen Martin/66/Gachon University/engineer
13/2022-06-03/yes/Chinho Cho/68/Gachon University/student
14/2022-08-15/yes/Kwangsu Cho/48/Gachon University/executive
15/2022-07-12/no/Tongbang Choi/26/Cornell University/engineer
16/2022-08-16/yes/Tongbang Kim/39/Tsinghua University/student
17/2022-08-14/no/Chunyong Chang/75/Gachon University/student
18/2022-06-14/no/Tongbang Park/32/New York University/engineer
19/2022-06-07/yes/Chunyong Kim/34/Harvard University/staff
20/2022-07-30/yes/Chinho Kim/52/Gachon University/engineer
21/2022-07-21/yes/Jude Smith/38/Cornell University/executive
22/2022-06-29/no/Tongbang Cho/29/Northwestern University/marketer
23/2022-06-15/yes/Seungmin Cho/71/Stanford University/professor
24/2022-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
25/2022-06-09/no/Archie Hunt/60/Fudan University/student
26/2022-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
27/2022-08-24/no/Konner French/42/Gachon University/professor
28/2022-08-27/no/Kwangsu Park/43/University of Pennsylvania/student
29/2022-06-08/yes/Bailey Houghton/31/Gachon University/engineer
30/2022-07-13/yes/Kyungmin Choi/44/Duke University/student
```

P4-1 Code

```
53     i++;
54 }
55 fclose(myinFile);
56
57 return i;
58 }
59
60
61 void tag_sort(struct PERSON data[]) {
62     struct PERSON temp[30];
63     for (int i = 0; i < 29; i++)
64     {
65         for (int j = 0; j < 29 - i; j++)
66         {
67             if (data[j].tag > data[j + 1].tag)
68             {
69                 temp[i] = data[j];
70                 data[j] = data[j + 1];
71                 data[j + 1] = temp[i];
72             }
73         }
74     }
75 }
76
77
78 int number_p = 0;
79 void linked_read_file(struct PERSON node[], int* num) {
80     number_p = 0;
```

```
80     number_p = 0;
81     FILE* IF = fopen("registraion.txt", "r");
82     char line[200];
83     while (fgets(line, sizeof(line), IF) != NULL) {
84         int count = 0;
85         char* ptr = strtok(line, "/");
86         while (ptr != NULL)
87         {
88             if (count == 0) {
89                 node[number_p].tag = atoi(ptr);
90                 count++;
91             }
92             else if (count == 1) {
93                 strcpy(node[number_p].date, ptr);
94                 count++;
95             }
96             else if (count == 2) {
97                 strcpy(node[number_p].fee, ptr);
98                 count++;
99             }
100             else if (count == 3) {
101                 strcpy(node[number_p].name, ptr);
102                 count++;
103             }
104             else if (count == 4) {
105                 node[number_p].age = atoi(ptr);
106                 count++;
107             }
108         }
109     }
```


P4-1 Codes & Result screenshots

```
107     }
108     else if (count == 5) {
109         strcpy(node[number_p].univ, ptr);
110         count++;
111     }
112     else if (count == 6) {
113         strcpy(node[number_p].job, ptr);
114         count++;
115     }
116     ptr = strtok(NULL, "/");
117 }
118 number_p++;
119 }
120 fclose(IF);
121 *num = number_p;
122 }
123
124 void main() {
125     int num = 0;
126     struct PERSON temp_n[300];
127     linked_read_file(temp_n, &num);
128     tag_sort(temp_n);
129     for (int i = 0; i < num; i++) {
130         printf("%d %s %s %s %d %s %s", temp_n[i].tag, temp_n[i].date, temp_n[i].fee, temp_n[i].name, temp_n[i].age, temp_n[i].univ, temp_n[i].job);
131     }
132 }
```

```
1/2022-08-25/yes/Jihu Choi/74/Harvard University/engineer
2/2022-08-22/no/Seungmin Choi/31/Gachon University/staff
3/2022-07-01/no/Chinho Park/53/Peking University/engineer
4/2022-07-08/no/Jihu Cho/71/Tsinghua University/engineer
5/2022-06-12/yes/Chunyoung Park/48/University of Cambridge/student
6/2022-06-04/yes/Bobby Anderson/33/McGill University/engineer
7/2022-06-28/yes/Jihu Park/70/Australian National Universitystudent/student
8/2022-06-04/no/Moises Barlow/57/Gachon University/engineer
9/2022-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
10/2022-06-06/yes/William Cohen/37/University of Cambridge/engineer
11/2022-07-22/no/Kwangsue Choi/48/Seoul National University/marketer
12/2022-07-22/no/Owen Martin/66/Gachon University/engineer
13/2022-06-03/yes/Chinho Cho/68/Gachon University/student
14/2022-08-15/yes/Kwangsue Choi/48/Gachon University/executive
15/2022-07-12/no/Tongbang Choi/26/Cornell University/engineer
16/2022-08-16/yes/Tongbang Kim/39/Tsinghua University/student
17/2022-08-14/no/Chunyoung Chang/75/Gachon University/student
18/2022-06-14/no/Tongbang Park/32/New York University/engineer
19/2022-06-07/yes/Chunyoung Kim/34/Harvard University/staff
20/2022-07-30/yes/Chinho Kim/52/Gachon University/engineer
21/2022-07-21/yes/Jude Smith/38/Cornell University/executive
22/2022-06-29/no/Tongbang Choi/29/Northwestern University/marketer
23/2022-06-15/yes/Seungmin Choi/71/Stanford University/professor
24/2022-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
25/2022-06-09/no/Archie Hunt/60/Fudan University/student
26/2022-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
27/2022-08-24/no/Konner French/42/Gachon University/professor
28/2022-08-27/no/Kwangsue Park/43/University of Pennsylvania/student
29/2022-06-08/yes/Bailey Houghton/31/Gachon University/engineer
30/2022-07-13/yes/Kyungmin Choi/44/Duke University/student
```

P5-1 Code

```
void Copy_Element(const PERSON* src, PERSON* dest)
{
    dest->tag = src->tag;
    strcpy(dest->date, src->date);
    strcpy(dest->fee, src->fee);
    strcpy(dest->name, src->name);
    dest->age = src->age;
    strcpy(dest->univ, src->univ);
    strcpy(dest->job, src->job);
    dest->next = src->next;
}
```

```
void selection_sort(const PERSON data[], const int num_persons, PERSON sort_age[])
{
    int least, temp;
    for (int i = 0; i < num_persons; i++) {
        Copy_Element(&data[i], &sort_age[i]);
    }
    for (int index = 0; index < num_persons - 1; index++) {
        least = index;
        for (int t = index + 1; t < num_persons; t++) {
            if (sort_age[t].age < sort_age[least].age) {
                least = t;
            }
        }
        if (index != least) {
            temp = sort_age[index].age;
            sort_age[index].age = sort_age[least].age;
            sort_age[least].age = temp;
        }
    }
}
```

```
bool Write_File(const char* fname, PERSON sort_age[], int num_persons)
{
    FILE* pFile;
    pFile = fopen(fname, "w");

    if (pFile == NULL) {
        printf("ERROR: Cannot read the file!\n");
        return false;
    }

    for (int index = 0; index < num_persons; index++) {
        fprintf(pFile, "%d-%d-%d %s %s %s %d %d %d %s\n", sort_age[index].tag, sort_age[index].date, sort_age[index].fee, sort_age[index].name, sort_age[index].age, sort_age[index].univ, sort_age[index].job);
    }

    fclose(pFile);
    return true;
}
```

main function

```
printf("[ P5 ]\n");
printf("Output to P5-1.txt\n");

PERSON sort_age[MAX];

selection_sort(arr_head, count, sort_age);

if (!Write_File("P5-1.txt", sort_age, count)) {
    return -1;
}
```


P5-1 Result screenshots

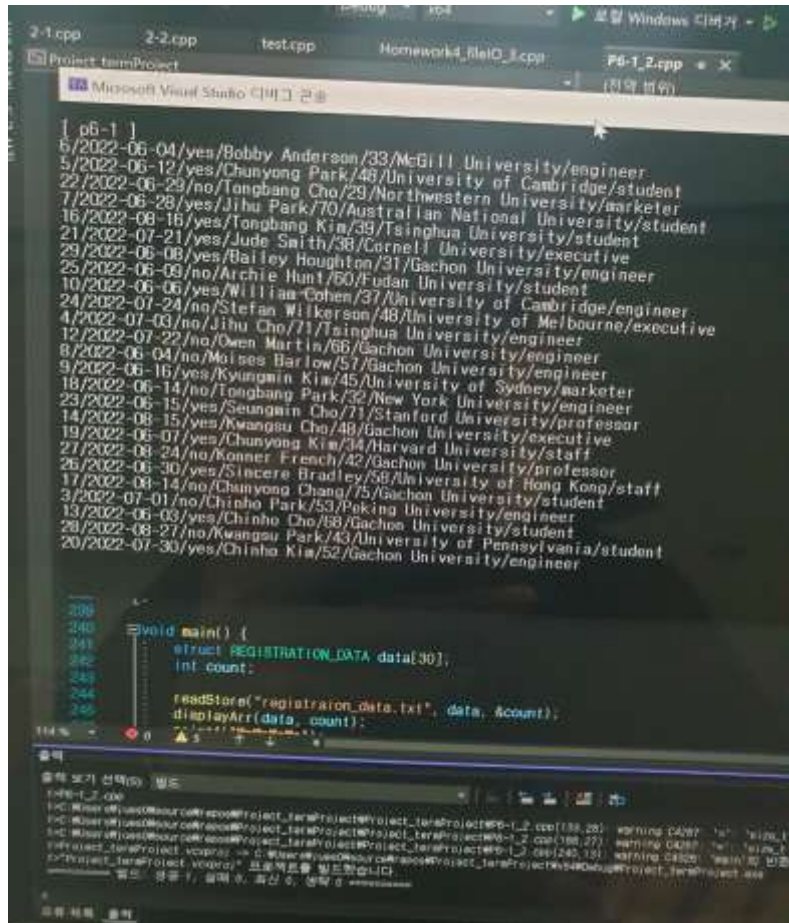
P5-1.txt - 메모장

파일 편집 보기

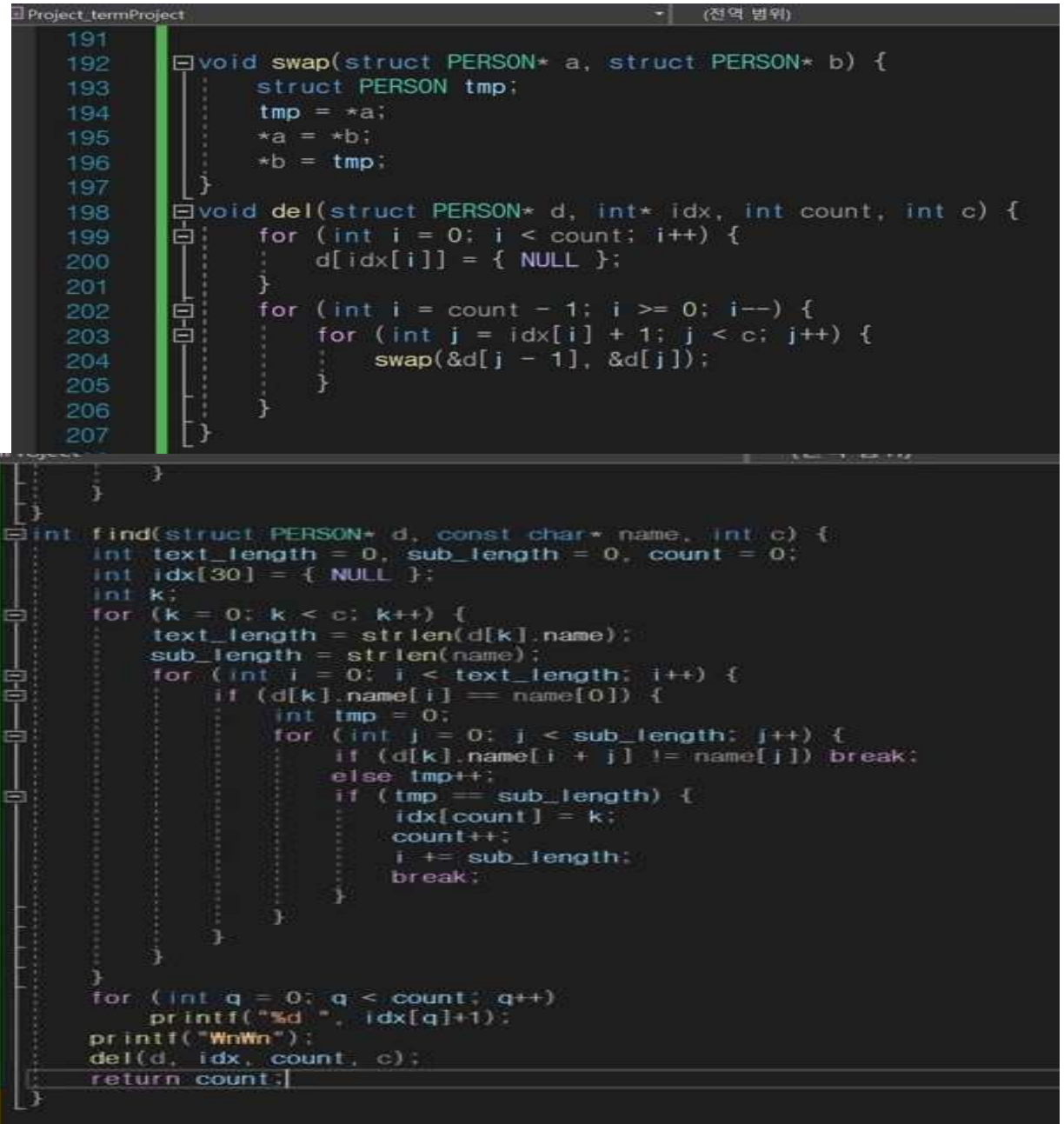
```
1/2022-08-25/yes/Jihu Choi/26/Harvard University/engineer
2/2022-08-22/no/Seungmin Choi/29/Gachon University/staff
3/2022-07-01/no/Chinho Park/31/Peking University/engineer
4/2022-07-03/no/Jihu Cho/31/Tsinghua University/engineer
5/2022-06-12/yes/Chunyong Park/32/University of Cambridge/student
6/2022-06-04/yes/Bobby Anderson/33/McGill University/engineer
7/2022-06-28/yes/Jihu Park/34/Australian National University/student
8/2022-06-04/no/Moises Barlow/37/Gachon University/engineer
9/2022-06-16/yes/Kyungmin Kim/38/University of Sydney/marketer
10/2022-06-06/yes/William Cohen/39/University of Cambridge/engineer
11/2022-07-22/no/Kwangsu Choi/42/Seoul National University/marketer
12/2022-07-22/no/Owen Martin/43/Gachon University/engineer
13/2022-06-03/yes/Chinho Cho/44/Gachon University/student
14/2022-08-15/yes/Kwangsu Cho/45/Gachon University/executive
15/2022-07-12/no/Tongbang Choi/48/Cornell University/engineer
16/2022-08-16/yes/Tongbang Kim/48/Tsinghua University/student
17/2022-08-14/no/Chunyong Chang/48/Gachon University/student
18/2022-06-14/no/Tongbang Park/48/New York University/engineer
19/2022-06-07/yes/Chunyong Kim/52/Harvard University/staff
20/2022-07-30/yes/Chinho Kim/53/Gachon University/engineer
21/2022-07-21/yes/Jude Smith/57/Cornell University/executive
22/2022-06-29/no/Tongbang Cho/58/Northwestern University/marketer
23/2022-06-15/yes/Seungmin Cho/60/Stanford University/professor
24/2022-07-24/no/Stefan Wilkerson/66/University of Melbourne/executive
25/2022-06-09/no/Archie Hunt/68/Fudan University/student
26/2022-06-30/yes/Sincere Bradley/70/University of Hong Kong/staff
27/2022-08-24/no/Konner French/71/Gachon University/professor
28/2022-08-27/no/Kwangsu Park/71/University of Pennsylvania/student
29/2022-06-08/yes/Bailey Houghton/74/Gachon University/engineer
30/2022-07-13/yes/Kyungmin Choi/75/Duke University/student
```

```
-----
[ P5 ]
Output to P5-1.txt
-----
```

P6-1 Code & Result screenshots



```
6/2022-06-04/yes/Bobby Anderson/33/McGill University/engineer
5/2022-06-12/yes/Chunyang Park/48/University of Cambridge/student
22/2022-06-29/no/Tongbang Cho/29/Northwestern University/marketer
7/2022-06-28/yes/Jihu Park/70/Australian National University/student
16/2022-08-16/yes/Tongbang Kim/39/Tsinghua University/student
21/2022-07-21/yes/Jude Smith/38/Cornell University/executive
29/2022-06-08/yes/Bailey Houghton/31/Gachon University/engineer
25/2022-06-09/no/Archie Hunt/60/Fudan University/student
10/2022-06-06/yes/William Cohen/37/University of Cambridge/engineer
24/2022-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
4/2022-07-03/no/Jihu Cho/71/Tsinghua University/engineer
12/2022-07-22/no/Owen Martin/66/Gachon University/engineer
8/2022-06-04/no/Moises Barlow/51/Gachon University/engineer
9/2022-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
18/2022-06-14/no/Tongbang Park/32/New York University/engineer
23/2022-06-15/yes/Seungmin Cho/71/Stanford University/professor
14/2022-08-15/yes/Kwangsue Cho/48/Gachon University/professor
19/2022-06-07/yes/Chunyang Kim/34/Harvard University/executive
27/2022-08-24/no/Konner French/42/Gachon University/staff
26/2022-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
3/2022-04-14/no/Chunyang Chang/75/Gachon University/student
13/2022-06-03/yes/Chinho Park/53/Peking University/engineer
28/2022-08-27/no/Kwangsue Park/43/University of Pennsylvania/student
20/2022-07-30/yes/Chinho Kim/52/Gachon University/engineer
```



```
void swap(struct PERSON* a, struct PERSON* b) {
    struct PERSON tmp;
    tmp = *a;
    *a = *b;
    *b = tmp;
}

void del(struct PERSON* d, int* idx, int count, int c) {
    for (int i = 0; i < count; i++) {
        d[idx[i]] = { NULL };
    }
    for (int i = count - 1; i >= 0; i--) {
        for (int j = idx[i] + 1; j < c; j++) {
            swap(&d[j - 1], &d[j]);
        }
    }
}

int find(struct PERSON* d, const char* name, int c) {
    int text_length = 0, sub_length = 0, count = 0;
    int idx[30] = { NULL };
    int k;
    for (k = 0; k < c; k++) {
        text_length = strlen(d[k].name);
        sub_length = strlen(name);
        for (int i = 0; i < text_length; i++) {
            if (d[k].name[i] == name[0]) {
                int tmp = 0;
                for (int j = 0; j < sub_length; j++) {
                    if (d[k].name[i + j] != name[j]) break;
                    else tmp++;
                }
                if (tmp == sub_length) {
                    idx[count] = k;
                    count++;
                    i += sub_length;
                    break;
                }
            }
        }
    }
    for (int q = 0; q < count; q++)
        printf("%d ", idx[q]+1);
    printf("\n\n");
    del(d, idx, count, c);
    return count;
}
```

P6-2 Code & Result screenshots

```
Microsoft Visual Studio 17.1.3 2022
[ p6-2 ]
6/2022-06-04/yes/Bobby Anderson/33/McGill University/engineer
5/2022-06-12/yes/Chunyang Park/48/University of Cambridge/student
22/2022-06-29/no/Tongbang Cho/29/Northwestern University/marketer
7/2022-06-28/yes/Jihu Park/70/Australian National University/student
16/2022-06-16/yes/Tongbang Kim/39/Tsinghua University/student
21/2022-07-21/yes/Jude Smith/38/Cornell University/student
29/2022-06-08/yes/Bailey Houghton/31/Gachon University/executive
25/2022-06-09/no/Archie Hunt/60/Fudan University/student
10/2022-06-06/yes/William Cohen/37/University of Cambridge/engineer
24/2022-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
4/2022-07-03/no/Jihu Cho/71/Tsinghua University/engineer
12/2022-07-22/no/Owen Martin/66/Gachon University/engineer
8/2022-06-04/no/Moises Barlow/57/Gachon University/engineer
9/2022-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
18/2022-06-14/no/Tongbang Park/32/New York University/engineer
23/2022-06-15/yes/Seungmin Cho/71/Stanford University/professor
14/2022-06-15/yes/Kwangsu Cho/40/Gachon University/executive
19/2022-06-07/yes/Chunyang Kim/34/Harvard University/staff
27/2022-06-24/no/Konner French/42/Gachon University/professor
26/2022-06-30/yes/Sicere Bradley/58/University of Hong Kong/staff
17/2022-06-14/no/Chunyang Chang/75/Gachon University/student
3/2022-07-01/no/Chinho Park/53/Peking University/engineer
13/2022-06-03/yes/Chinho Cho/68/Gachon University/student
28/2022-06-27/no/Kwangsu Park/43/University of Pennsylvania/student
20/2022-07-30/yes/Chinho Kim/52/Gachon University/engineer

C:\Users\junes\source\repos\Project_ters\Project_ters\Debug\Project_ters\project.
239
240 void main() {
241     struct REGISTRATION_DATA data[30];
242     int count;
243
244     readStore("registration_data.txt", data, &count);
245     displayArr(data, count);
246 }
```

```
246 int cmp(struct PERSON* d, const char* name) {
247     int text_length = 0, sub_length = 0, count = 0;
248
249     text_length = strlen(d->name);
250     sub_length = strlen(name);
251     for (int i = 0; i < text_length; i++) {
252         if (d->name[i] == name[0]) {
253             int tmp = 0;
254             for (int j = 0; j < sub_length; j++) {
255                 if (d->name[i + j] != name[j])
256                     break;
257                 else tmp++;
258             }
259             if (tmp == sub_length) {
260                 count++;
261                 i += sub_length;
262                 break;
263             }
264         }
265     }
266     return count;
267 }
268 void deleteNode(struct PERSON* head, const char* name) {
269     struct PERSON* cNode = head;
270     struct PERSON* pNode = NULL;
271     while (cNode != NULL) {
272         if (cmp(cNode, name) == 1) break;
273         pNode = cNode;
274         cNode = cNode->next;
275     }
276     pNode->next = cNode->next;
277 }
317
318 int del_cnt = find(arr_head, "Choi", count);
319 printf("[ p6-1 ]\n");
320 print_arr(arr_head, count - del_cnt, &p);
321
322 printf("[ p6-2 ]\n");
323 for (int j = 0; j < del_cnt; j++)
324     deleteNode(head, "Choi");
325 print_list(head, &p);
326
327
328
329
```


P7-1, P7-2 Code

```
pra.cpp* 2-1.cpp 2-2.cpp test.cpp Homework4_fileIO_3.cpp P6-1_2.cpp
Project_termProject (전역 범위)
266
267 void p7_1(struct PERSON* ptr, int num) {
268     int n = 30;
269     ptr[num].tag = n;
270     strcpy(ptr[num].date, "2022-11-30");
271     strcpy(ptr[num].fee, "yes");
272     strcpy(ptr[num].name, "Gildong Paik");
273     ptr[num].age = 35;
274     strcpy(ptr[num].univ, "Gachon University");
275     strcpy(ptr[num].job, "Student");
276 }
277
278
279 void p7_2(struct PERSON* head, int num) {
280     struct PERSON* ptr = (struct PERSON*)malloc(sizeof(struct PERSON));
281     ptr->tag = 30;
282     strcpy(ptr->date, "2022-11-03");
283     strcpy(ptr->fee, "yes");
284     strcpy(ptr->name, "Gildong Paik");
285     ptr->age = 35;
286     strcpy(ptr->univ, "Gachon University");
287     strcpy(ptr->job, "Student");
288     ptr->next = NULL;
289     head = add_list(head, ptr);
290 }
291
292
```

P7-1, P7-2 Result screenshots

```
Microsoft Visual Studio 디버그 콘솔
[ p7-1 ]
3/2022-07-01/no/Chinho Park/53/Peking University/engineer
4/2022-07-03/no/Jihu Cho/71/Tsinghua University/engineer
5/2022-06-12/yes/Chunyang Park/48/University of Cambridge/student
6/2022-06-04/yes/Bobby Anderson/33/McGill University/engineer
7/2022-06-28/yes/Jihu Park/70/Australian National University/engineer
8/2022-06-04/no/Moises Barlow/57/Gachon University/engineer
9/2022-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
10/2022-06-06/yes/William Cohen/37/University of Cambridge/engineer
12/2022-07-22/no/Owen Martin/66/Gachon University/engineer
13/2022-06-03/yes/Chinho Cho/68/Gachon University/student
14/2022-08-15/yes/Kwangsu Cho/48/Gachon University/executive
16/2022-08-16/yes/Tongbang Kim/39/Tsinghua University/student
17/2022-08-14/no/Chunyang Chang/75/Gachon University/student
18/2022-06-14/no/Tongbang Park/32/New York University/engineer
19/2022-06-07/yes/Chunyang Kim/34/Harvard University/staff
20/2022-07-30/yes/Chinho Kim/52/Gachon University/engineer
21/2022-07-21/yes/Jude Smith/38/Cornell University/executive
22/2022-06-29/no/Tongbang Cho/29/Northwestern University/marketer
23/2022-06-15/yes/Seungmin Cho/71/Stanford University/professor
24/2022-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
25/2022-06-09/no/Archie Hunt/60/Fudan University/student
26/2022-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
27/2022-08-24/no/Konner French/42/Gachon University/professor
28/2022-08-21/no/Kwangsu Park/43/University of Pennsylvania/student
29/2022-06-08/yes/Bailey Houghton/31/Gachon University/engineer
30/2022-11-30/yes/Gildong Paik/35/Gachon University/Student
[ p7-2 ]
6/2022-06-04/yes/Bobby Anderson/33/McGill University/engineer
strcpy(ptr[num].name, "Gildong Paik");
```

```
Microsoft Visual Studio 디버그 콘솔
[ p7-2 ]
6/2022-06-04/yes/Bobby Anderson/33/McGill University/engineer
5/2022-06-12/yes/Chunyang Park/48/University of Cambridge/student
22/2022-06-29/no/Tongbang Cho/29/Northwestern University/marketer
7/2022-06-28/yes/Jihu Park/70/Australian National University/student
16/2022-08-16/yes/Tongbang Kim/39/Tsinghua University/student
21/2022-07-21/yes/Jude Smith/38/Cornell University/executive
29/2022-06-08/yes/Bailey Houghton/31/Gachon University/engineer
25/2022-06-09/no/Archie Hunt/60/Fudan University/student
10/2022-06-06/yes/William Cohen/37/University of Cambridge/engineer
24/2022-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
4/2022-07-03/no/Jihu Cho/71/Tsinghua University/engineer
12/2022-07-22/no/Owen Martin/66/Gachon University/engineer
8/2022-06-04/no/Moises Barlow/57/Gachon University/engineer
9/2022-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
18/2022-06-14/no/Tongbang Park/32/New York University/engineer
23/2022-06-15/yes/Seungmin Cho/71/Stanford University/professor
14/2022-08-15/yes/Kwangsu Cho/48/Gachon University/executive
19/2022-06-07/yes/Chunyang Kim/34/Harvard University/staff
21/2022-08-24/no/Konner French/42/Gachon University/professor
26/2022-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
17/2022-08-14/no/Chunyang Chang/75/Gachon University/student
3/2022-07-01/no/Chinho Park/53/Peking University/engineer
13/2022-06-03/yes/Chinho Cho/68/Gachon University/student
28/2022-08-21/no/Kwangsu Park/43/University of Pennsylvania/student
20/2022-07-30/yes/Chinho Kim/52/Gachon University/engineer
30/2022-11-03/yes/Gildong Paik/35/Gachon University/Student
C:\Users\jues\source\repos\Project_term\Project\Debug\Project.exe(프로
(코드: 0x0).
```

P8-1 & P8-2 Code

```
int CheckSum(const PERSON input[])
{
    int XOR = 0;

    PERSON temp;
    for (int i = 0; i < 10; i++) {
        Copy_Element(&input[i], &temp);

        int k = 0;
        while (temp.name[k] != '\0') {
            XOR ^= temp.name[k];
            k += 1;
        }
    }

    return XOR;
}
```

```
void p8(const PERSON* original, const int numberOfPerson)
{
    PERSON mostRecent[10];
    int startIdx = numberOfPerson - 10;

    int mostRecentNum = 0;
    for (int idx = startIdx; idx < numberOfPerson; idx++)
    {
        Copy_Element(&original[idx], &mostRecent[idx - startIdx]);
        mostRecentNum += 1;
    }

    int org_checkSum = CheckSum(mostRecent);
    Write_File("P8-1.txt", mostRecent, mostRecentNum);

    PERSON afterWrite[10];
    int afterPersons = 0;
    Read_File("P8-1.txt", afterWrite, &afterPersons);
    Write_File("P8-2.txt", afterWrite, afterPersons);

    PERSON copiedData[10];
    int copiedPersons = 0;
    Read_File("P8-2.txt", copiedData, &copiedPersons);
    int cpy_checkSum = CheckSum(copiedData);

    if (org_checkSum == cpy_checkSum)
    {
        printf("-----Copy recent data and Exclusive OR Checksum-----\n");
        printf("[Org_Checksum]: %d\n", org_checkSum);
        printf("[Cpy_Checksum]: %d\n", cpy_checkSum);
        printf("\nCopy Complete !!\n");
    }
    else
    {
        printf("-----Copy recent data and Exclusive OR Checksum-----\n");
        printf("[Org_Checksum]: %d\n", org_checkSum);
        printf("[Cpy_Checksum]: %d\n", cpy_checkSum);
        printf("\nCopy Fail...\n");
    }
}
```

P8-1 & P8-2 Code

```
bool Read_File(const char* fname, PERSON* person, int* num_persons)
{
    FILE* pFile;
    pFile = fopen(fname, "r");

    if (pFile == NULL)
    {
        printf("ERROR: Cannot read the file !!\n");
        *num_persons = 0;
        return false;
    }

    PERSON read_data;
    while (fscanf(pFile, "%d/%d/%d/%d/%d/%d/%d/%d", &read_data.tag, &read_data.date, &read_data.fee, &read_data.name, &read_data.age, &read_data.univ, &read_data.job) == 7)
    {
        Copy_Element(&read_data, &person[(*num_persons)++]);
    }


    fclose(pFile);
    return true;
}
```

main function

```
printf("[ p8-1 && p8-2 ]\n\n");
p8(arr_head, count);


return 0;
}
```

P8-1 & P8-2 Result screenshots

 P8-1.txt - 메모장

파일 편집 보기

21/2022-07-21/yes/Jude Smith/38/Cornell University/executive
22/2022-06-29/no/Tongbang Cho/29/Northwestern University/marketer
23/2022-06-15/yes/Seungmin Cho/71/Stanford University/professor
24/2022-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
25/2022-06-09/no/Archie Hunt/60/Fudan University/student
26/2022-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
27/2022-08-24/no/Konner French/42/Gachon University/professor
28/2022-08-27/no/Kwangsu Park/43/University of Pennsylvania/student
29/2022-06-08/yes/Bailey Houghton/31/Gachon University/engineer
30/2022-11-30/yes/Gildong Paik/35/Gachon University/Student

 P8-2.txt - 메모장

파일 편집 보기

21/2022-07-21/yes/Jude Smith/38/Cornell University/executive
22/2022-06-29/no/Tongbang Cho/29/Northwestern University/marketer
23/2022-06-15/yes/Seungmin Cho/71/Stanford University/professor
24/2022-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
25/2022-06-09/no/Archie Hunt/60/Fudan University/student
26/2022-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
27/2022-08-24/no/Konner French/42/Gachon University/professor
28/2022-08-27/no/Kwangsu Park/43/University of Pennsylvania/student
29/2022-06-08/yes/Bailey Houghton/31/Gachon University/engineer
30/2022-11-30/yes/Gildong Paik/35/Gachon University/Student

```
[ p8-1 && p8-2 ]
```

```
-----Copy recent data and Exclusive OR Checksum-----
```

```
[Org_Checksum]: 21
```

```
[Cpy_Checksum]: 21
```

```
Copy Complete !!
```