Term-Project

●강의명 : 문제해결기법
●학수번호 : **13967005**
● 팀 : **Team-H**

Team introduction

- Cho Hyunseo: solve the problem & merge all codes
- Cha Junha: solve the problem
- Hong Heejin: solve the problem
- Ban Jaehyeon: solve the problem

Contribution percentage

- Cho Hyunseo: 30%
- Cha Junha: 25%
- Hong Heejin: 25%
- Ban Jaehyeon: 20%

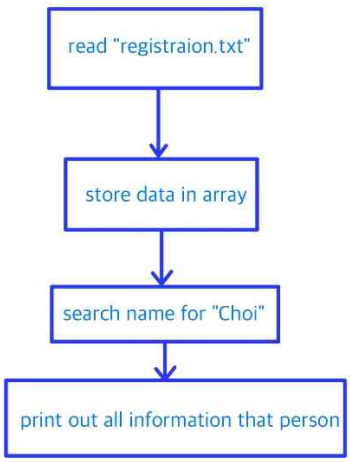
Achievement table

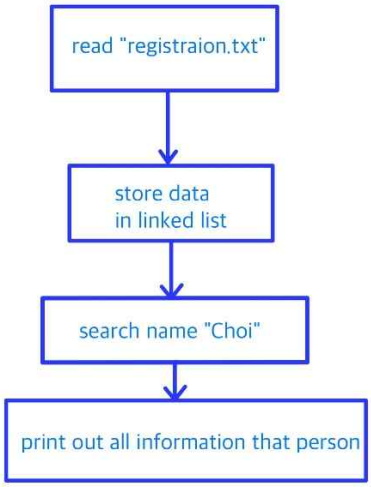
Problem	Members Involved	Achievment
P1	Hong Heejin	100%
P2	Hong Heejin	100%
P3	Cha Junha	100%
P4	Cha Junha	100%
P5	Cho Hyunseo	100%
P6	Ban Jaehyeon, Cho Hyunseo	100%
P7	Ban Jaehyeon	100%
P8	Cho Hyunseo	100%

<P1>

Step1. Read file and store data in array and linked list, search for "Choi" in both.

Step2. Print all information about the persons whose name is "Choi" by array and linked list.





Step3.

<P1-1>

1. Read "registraion.txt" file
2. Store data in structure array[0] to [29].
3. Search about name "Choi"
4. Print out all data matched name in array

<P1-2>

1. Read "registraion.txt" file
2. Store data in linked list line by line.
3. Search about name "Choi"
4. Print out all data matched name in linked list

Step4.

<P1-1>

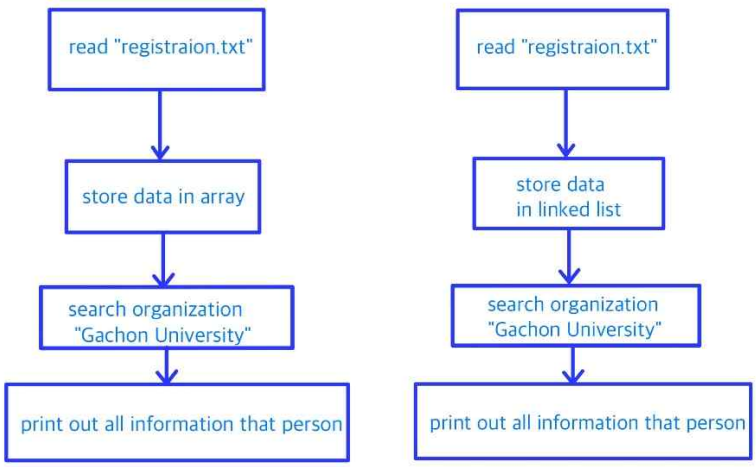
```
PERSONAL persons_array[]
search_name[] = "Choi";
*name_key = search_name
if (!read_file_array)
        return -1
```

for t= 0 to t<MAX_PERSON
PERSONAL* ptr = &persons_array[t];
if (!SearchNameInArr(ptr, name_key))
return -1
<P1-2>
PERSONAL* persons = NULL;
search_name[] = "Choi";
* name_key = search_name;
if (!read_file_linked)
return -1
if (!SearchNameInLL(persons, name_key))
return -1

<P2>

Step1. Read file and store data in array and linked list, search for "Gachon University" in both.

Step2. Print all information about the persons whose organization is "Gachon University" by array and linked list.



Step3.

<P2-1>

1. Read "registraion.txt" file
2. Store data in structure array[0] to [29].
3. Search about organization "Gachon University"

4. Print out all data matched organization in array

<P2-2>

1. Read “registraion.txt” file
2. Store data in linked list line by line.
3. Search about organization “Gachon University”
4. Print out all data matched organization in linked list

Step4.

<P2-1>

```
PERSONAL persons_array[]
search_org[] = "Gachon University";
* org_key = search_org;
if (!read_file_array)
        return -1
for t= 0 to t<MAX_PERSON
PERSONAL* ptr = &persons_array[t];
if (!SearchOrgInArr(ptr, org_key))
                return -1
```

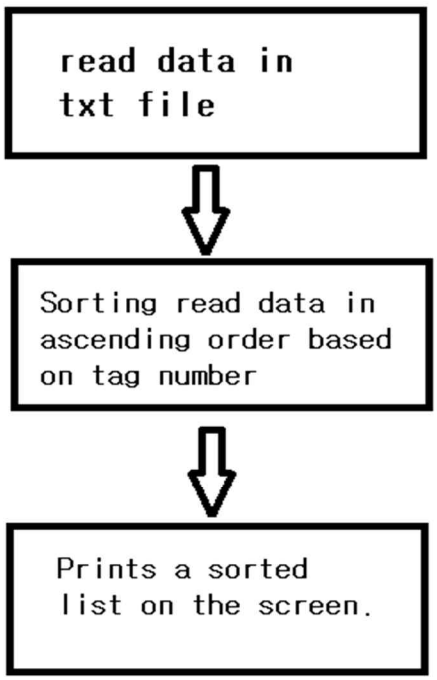
<P2-2>

```
PERSONAL* persons = NULL;
search_org[] = " Gachon University ";
*org_key = search_org;
if (!read_file_linked)
return -1
if (!SearchOrgInLL(persons, org_key))
return -1
```

P3

Step 1. Reading data into a txt file, sorting the data in ascending order based on tag number, and print it

Step 2. Prints 30 data sorted by tag number and print sort data.



Step 3.

3-1. Read a file data and copy persons[0~29] ...

3-2. The values in the persons array are sorted by bubble sort.

3-3. Print result of sorting data (all persons data)

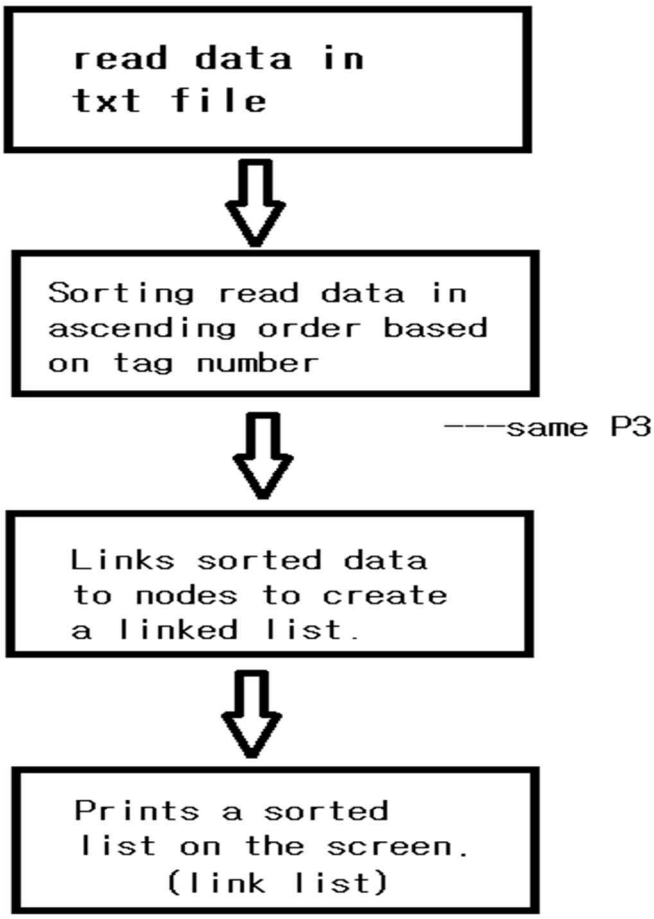
Step 4.

```
struct RegList_Data persons[30];
int num_persons = 0;
if ((read_File("registraion_data.txt", persons, &num_persons)) != true)          return false;
sort_Data_tagNumber(persons, num_persons);
print_Data(persons, num_persons);
```

P4

Step 1. Print the sorted values after concatenating them into a list through a link.

Step 2. Data sorted in ascending order are connected by nodes and links. and print it out.



Step 3.

3-1. Read a file data and copy persons[0~29] ...

3-2. The values in the persons array are sorted by bubble sort.

3-3. Link the sort data to create a linked list.

3-4. Print result of linked list sort data (all persons data)

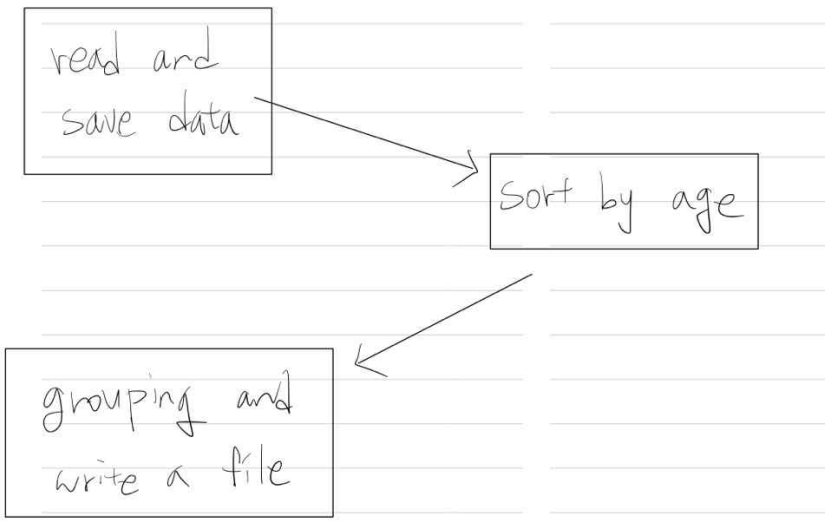
Step 4.

```
struct RegList_Data persons[30];
int num_persons = 0;
if ((read_File("registraion_data.txt", persons, &num_persons)) != true)return false;

sort_Data_tagNumber(persons, num_persons);

struct RegList_Data* head = malloc(sizeof(struct RegList_Data));
struct RegList_Data* node1 = malloc(sizeof(struct RegList_Data));

head->next = node1;
node1->next = NULL;
```

```
for (int i = 0; i < num_persons; i++)   change_linkedList(head, &persons[i], num_persons);
 printList(node1);
```

P5

Step 1. Read a file. Sort by age(Selection Sort). Grouped by age. Write a file.
Step 2. The result file's data must be grouped by age.



Step 3.

1. Read a file
2. Save data[0], data[1], data[2], …
3. Sorting data by age
4. Grouping and write a file

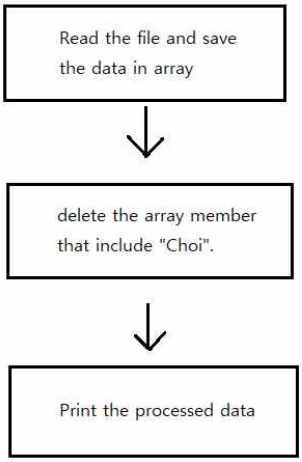
Step 4.

```
if (readFile == false)
        return -1
selectionSort(data, MAX)
if (writeFile = false)
        return -1
```

P6-1.

Step 1. Read a file and store the data in the array. Search “Choi” and delete data chunk that include “Choi”. Print the processed data.

Step 2. Print the data except about the people who have name “Choi”.



Step 3.

1.Read the data and store it in the array.

2.Delete all the member whose name is “Choi” in the array.

3.Print the data.

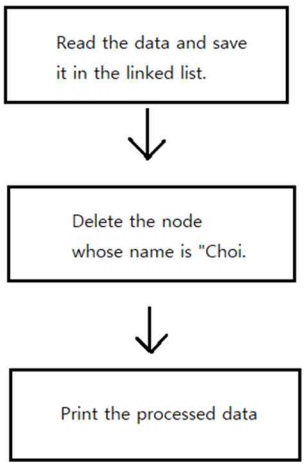
Step 4.

```
Struct Data data[30]
if(readfile==false)
return -1;
Storedata(file, data);
deleteChoi(file, data);
Print(file, data);
```

P6-2.

Step 1. Read a file and store the data in the linked list. Search “Choi” and delete node that include “Choi”. Print the processed data.

Step 2. Print the data except the node that name is “Choi”.



Step 3.

1.Read the data and store it in the linked list.

2.Delete all the node whose name is “Choi”.

3.Print the data.

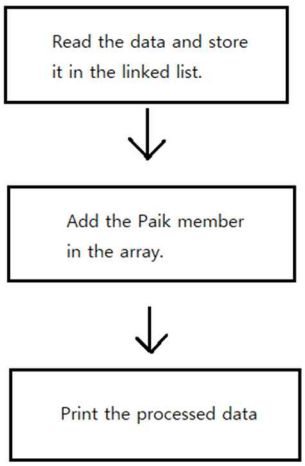
Step 4.

```
Struct Data={data; Data next;};
if(readfile==false)
return -1;
AddNode(file, head);
deleteChoi(file, head);
Print(file, head);
```

P 7-1.

Step 1. Read a file and store the data in the array. Add Paik(#/2020-11-30/yes/Gildong Paik/35/Gachon University/Student.) in the array. Print the processed data.

Step 2. Add the Paik in the array and print the data.



Step 3.

1.Read the data and store it in the array.

2.Add the Paik member in the array.

3.Print the data.

Step 4.

Struct Data data[30];

if(readfile==false)

return -1;

Storedata (file, data);

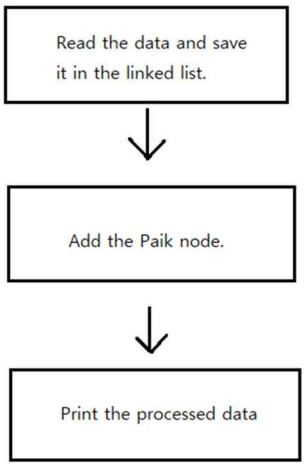
AddPaik(file, data);

Print(file, data);

P 7-2.

Step 1. Read a file and store the data in the linked list. Add Paik(#/2020-11-30/yes/Gildong Paik/35/Gachon University/Student.) node. Print the processed data.

Step 2. Add the Paik node and print the data.



Step 3.

1.Read the data and store it in the linked list.

2.Add the Paik node.

3.Print the data.

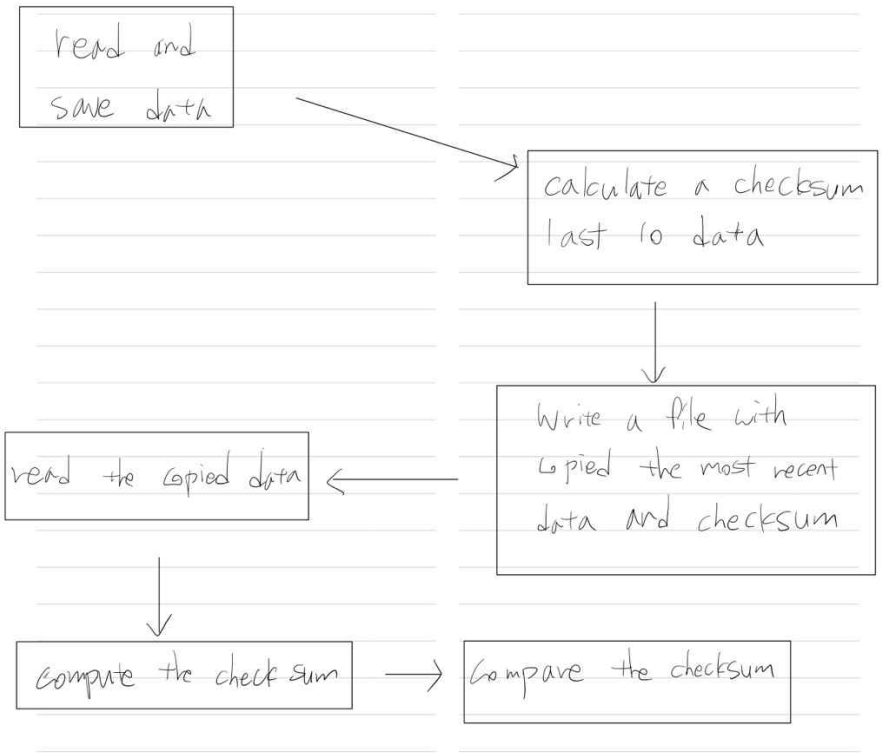
Step 4.

```
Struct Data ={data; Data next;};
if(readfile==false)
return -1;
Storedata (file, head);
AddPaik(file, head);
Print(file, head);
```

P8

Step 1. Read a file. Calculate a checksum. Write a file with copied the most recent data. Read a copied file. Compare a checksum.

Step 2. The original's checksum and copied's checksum must be same.



Step 3.

1. Read a file
2. Save data[0], data[1], data[2], …
3. Calculate a checksum of last 10 data
4. Write a file with last 10 data and original's checksum
5. Read a copied file
6. Calculate a copied's checksum
7. Compare the original's checksum and copied's

Step 4.

```
if (readFile == false)
        return -1
computeChecksum(original)
if (writeCopiedFile == false)
        retrun -1
if (readCopiedFile == false)
        return -1
computeChecksum(copied)
```

Compare a checksum

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>

#define MAX_NAME 25
#define MAX_ORGANIZATION 35
#define MAX_JOB 15
#define MAX 30

#define MAX_PERSON 30
#define MAX_NAME_LEN 25
#define MAX_ORG_LEN 35
#define MAX_JOB_LEN 15
#define NUM_TEST_SET 2

// p1
/**
 * @ Author: Hong Heejin
 * @ Create Time: 2021-11-19
 * @ Modified time: 2021-11-23
 * @ Description: term project P1
 */
struct PERSONAL
{
	int tag;
	char date[11];
	char fee_paid[4];
	char name[MAX_NAME_LEN];
	int age;
	char org[MAX_ORG_LEN];
	char job[MAX_JOB_LEN];
	struct PERSONAL* next;
};
void copy_element_1(const PERSONAL* src, PERSONAL* dest) {
	dest->tag = src->tag;
	strcpy(dest->date, src->date);
	strcpy(dest->fee_paid, src->fee_paid);
```

```
    strcpy(dest->name, src->name);
    dest->age = src->age;
    strcpy(dest->org, src->org);
    strcpy(dest->job, src->job);
}
//print personal information
void PrintInfo_1(struct PERSONAL* persons) {
    printf("%d/%s/%s/%s/%d/%s/%s₩n", persons->tag, persons->date,
        persons->fee_paid, persons->name, persons->age,
        persons->org, persons->job);
}
//P1-1. Read file and store in array
bool read_File_1_array(const char* fname, PERSONAL persons[]) {
    char line[256];
    char* ptr;
    FILE* pFile;
    pFile = fopen(fname, "r");
    if (pFile == NULL) {
        printf("Error: Cannot Read File: %s₩n", fname);

        return false;
    }

    int num = 0;
    PERSONAL read_person;

    while (fgets(line, sizeof(line), pFile) != NULL)
    {
        ptr = strtok(line, "/");
        read_person.tag = atoi(ptr);
        ptr = strtok(NULL, "/");
        strcpy(read_person.date, ptr);
        ptr = strtok(NULL, "/");
        strcpy(read_person.fee_paid, ptr);
        ptr = strtok(NULL, "/");
        strcpy(read_person.name, ptr);
        ptr = strtok(NULL, "/");
        read_person.age = atoi(ptr);
        ptr = strtok(NULL, "/");
```

```
            strcpy(read_person.org, ptr);
            ptr = strtok(NULL, "/");
            strcpy(read_person.job, ptr);

            copy_element_1(&read_person, &persons[num]);
            num++;
        }

        fclose(pFile);
        return true;
}
//P1-2. Create new node
PERSONAL* createNode_1(PERSONAL src) {
        PERSONAL* dest = (PERSONAL*)malloc(sizeof(PERSONAL));
        dest->tag = src.tag;
        strcpy(dest->date, src.date);
        strcpy(dest->fee_paid, src.fee_paid);
        strcpy(dest->name, src.name);
        dest->age = src.age;
        strcpy(dest->org, src.org);
        strcpy(dest->job, src.job);
        dest->next = NULL;
        return dest;
}
//P1-2. Read file and store linked list
bool read_File_1_linked(const char* fname, PERSONAL** head) {
        char line[256];
        char* p;
        FILE* pFile;
        pFile = fopen(fname, "r");
        if (pFile == NULL) {
            printf("Error: Cannot Read File: %s₩n", fname);

            return false;
        }
        int num = 0;
        PERSONAL temp;

        while (fgets(line, sizeof(line), pFile) != NULL)
```

```
	{
		p = strtok(line, "/");
		temp.tag = atoi(p);
		p = strtok(NULL, "/");
		strcpy(temp.date, p);
		p = strtok(NULL, "/");
		strcpy(temp.fee_paid, p);
		p = strtok(NULL, "/");
		strcpy(temp.name, p);
		p = strtok(NULL, "/");
		temp.age = atoi(p);
		p = strtok(NULL, "/");
		strcpy(temp.org, p);
		p = strtok(NULL, "/");
		strcpy(temp.job, p);
		temp.next = NULL;

		PERSONAL* newNode = createNode_1(temp);
		if (*head == NULL) {

			*head = newNode;
		}
		else {
			PERSONAL* ptr = *head;
			//find last node loop
			while (ptr->next)
			{
				ptr = ptr->next;
			}
			//if last node, connect new node
			ptr->next = newNode;
		}
	}
	fclose(pFile);

	return true;
}
//P1-1. Search matched name in array and print all information that name
bool SearchNameInArr(PERSONAL* str, const char* search_name) {
```

```
	int cursor;
	int i = 0;
	for (cursor = 0; cursor < (signed)strlen(str->name); cursor++)
	{
		if (str->name[cursor] == search_name[0]) {
			bool found = true;
			for (int i = 1; i < (signed)strlen(search_name); i++) {

				if (str->name[cursor + i] != search_name[i])
				{
					found = false;
					break;
				}
			}
			// if found
			if (found)
			{
				PrintInfo_1(str);
				return true;
			}
		}
	}
	return false;
}
//P1-2. Search matched name in linked list
bool SearchNameInLL(struct PERSONAL* persons, const char* search_name) {
	int num = 0;
	int cursor;
	int i = 0;

	while (persons) {
		for (cursor = 0; cursor < (signed)strlen(persons->name); cursor++) {
			if (persons->name[cursor] == search_name[0]) {
				bool found = true;
				for (int i = 1; i < (signed)strlen(search_name); i++) {

					if (persons->name[cursor + i] != search_name[i])
					{
						found = false;
```

```
                        break;
                    }
                }
                // if found
                if (found)
                {
                    PrintInfo_1(persons);
                }
            }
        }
        persons = persons->next;
        cursor = 0;
    }
    return true;
}

// p2
/**
 * @ Author: Hong Heejin
 * @ Create Time: 2021-11-19
 * @ Modified time: 2021-11-23
 * @ Description: term project P2
 */
void copy_element_2(const PERSONAL* src, PERSONAL* dest) {
    dest->tag = src->tag;
    strcpy(dest->date, src->date);
    strcpy(dest->fee_paid, src->fee_paid);
    strcpy(dest->name, src->name);
    dest->age = src->age;
    strcpy(dest->org, src->org);
    strcpy(dest->job, src->job);
}
//P2-2. Create new node
PERSONAL* createNode_2(PERSONAL src) {
    PERSONAL* dest = (PERSONAL*)malloc(sizeof(PERSONAL));
    dest->tag = src.tag;
    strcpy(dest->date, src.date);
    strcpy(dest->fee_paid, src.fee_paid);
    strcpy(dest->name, src.name);
```

```
    dest->age = src.age;
    strcpy(dest->org, src.org);
    strcpy(dest->job, src.job);
    dest->next = NULL;
    return dest;
}
//P2-1. Read file and store in array
bool read_File_2_array(const char* fname, PERSONAL persons[]) {
    char line[256];
    char* ptr;
    FILE* pFile;
    pFile = fopen(fname, "r");
    if (pFile == NULL) {
        printf("Error: Cannot Read File: %s₩n", fname);
        return false;
    }

    int num = 0;
    PERSONAL read_person;

    while (fgets(line, sizeof(line), pFile) != NULL)
    {
        ptr = strtok(line, "/");
        read_person.tag = atoi(ptr);
        ptr = strtok(NULL, "/");
        strcpy(read_person.date, ptr);
        ptr = strtok(NULL, "/");
        strcpy(read_person.fee_paid, ptr);
        ptr = strtok(NULL, "/");
        strcpy(read_person.name, ptr);
        ptr = strtok(NULL, "/");
        read_person.age = atoi(ptr);
        ptr = strtok(NULL, "/");
        strcpy(read_person.org, ptr);
        ptr = strtok(NULL, "/");
        strcpy(read_person.job, ptr);

        copy_element_2(&read_person, &persons[num]);
        num++;
```

```
    }

    fclose(pFile);
    return true;
}
//P2-2. Read file and store linked list
bool read_File_2_linked(const char* fname, PERSONAL** head) {
    char line[256];
    char* p;
    FILE* pFile;
    pFile = fopen(fname, "r");
    if (pFile == NULL) {
        printf("Error: Cannot Read File: %s₩n", fname);

        return false;
    }
    int num = 0;
    PERSONAL temp;

    while (fgets(line, sizeof(line), pFile) != NULL)
    {
        p = strtok(line, "/");
        temp.tag = atoi(p);
        p = strtok(NULL, "/");
        strcpy(temp.date, p);
        p = strtok(NULL, "/");
        strcpy(temp.fee_paid, p);
        p = strtok(NULL, "/");
        strcpy(temp.name, p);
        p = strtok(NULL, "/");
        temp.age = atoi(p);
        p = strtok(NULL, "/");
        strcpy(temp.org, p);
        p = strtok(NULL, "/");
        strcpy(temp.job, p);
        temp.next = NULL;

        PERSONAL* newNode = createNode_2(temp);
        if (*head == NULL) {
```

```
            *head = newNode;
        }
        else {
            PERSONAL* ptr = *head;
            //find last node loop
            while (ptr->next)
            {
                ptr = ptr->next;
            }
            //if last node, connect new node
            ptr->next = newNode;
        }
    }
    fclose(pFile);

    return true;
}
//print personal information
void PrintInfo_2(struct PERSONAL* persons) {
    printf("%d/%s/%s/%s/%d/%s/%s₩n", persons->tag, persons->date,
        persons->fee_paid, persons->name, persons->age,
        persons->org, persons->job);
}
//P2-1. Search matched organization in array and print all information that one
bool SearchOrgInArr(PERSONAL* str, const char* search_org) {
    int i = 0;
    if (strcmp(str->org, search_org) == 0) {
        bool found = true;
        // if found
        if (found)
        {
            PrintInfo_2(str);
            return true;
        }
        else {
            return false;
        }
    }
```

```
        return false;
    }
    //P2-2. Search matched organzation in linked list and print all information that one
    bool SearchOrgInLL(struct PERSONAL* persons, const char* search_org) {
        int i = 0;

        while (persons) {

            if (strcmp(persons->org, search_org) == 0) {
                bool found = true;
                // if found
                if (found)
                {
                    PrintInfo_2(persons);
                }
            }
            persons = persons->next;
        }
        return true;
    }
    // p3
    //made by jun ha cha

    struct RegList_Data_1        //Creating a structure.
    {
            int tag;
            char date_Registered[20];
            char fee_paid[10];
            char name[25];
            int age;
            char organization[35];
            char job[15];
    };

    void copy_element_3(struct RegList_Data_1* dest, struct RegList_Data_1* src) // This function copies the transmitted data to persons
    {
            dest->tag = src->tag;
            strcpy(dest->date_Registered, src->date_Registered);
```

```
	strcpy(dest->fee_paid, src->fee_paid);
	strcpy(dest->name, src->name);
	dest->age = src->age;
	strcpy(dest->organization, src->organization);
	strcpy(dest->job, src->job);
}

bool read_File(const char* fname, struct RegList_Data_1* persons, int* num_persons) //This function reads data into a file.
{
	FILE* pFile;
	pFile = fopen(fname, "r");
	if (pFile == NULL)
	{
		printf("Error : Cannot read file : %s₩n", fname);
		*num_persons = 0;
		return false;
	}
	struct RegList_Data_1 reading_Data;
	int num = 0;
	while (fscanf(pFile, "%d/%[^/]/%[^/]/%[^/]/%d/%[^/]/%s", &reading_Data.tag, &reading_Data.date_Registered, &reading_Data.fee_paid, &reading_Data.name, &reading_Data.age, &reading_Data.organization, &reading_Data.job) == 7)
	{
		copy_element_3(&persons[num], &reading_Data); //Copy the data you read divided according to the format.
		num++;
	}
	fclose(pFile);
	*num_persons = num; //Save the number of data.
	return true;
}

void sort_Data_tagNumber(struct RegList_Data_1* persons, int num_persons) // This function sort data using 'bubble sort' based on tag number.
{
	struct RegList_Data_1 temp_Data;
	for (int k = 0; k < num_persons; k++)
	{
```

```
                for (int i = 0; i < num_persons - 1; i++)
                {
                        if (persons[i].tag > persons[i + 1].tag)// Read the information and send it back if there is a larger tag Number value.
                        {
                                temp_Data = persons[i];
                                persons[i] = persons[i + 1];
                                persons[i + 1] = temp_Data;

                        }
                }
        }
  }

  void print_Data(struct RegList_Data_1* persons, int num_persons) //Output sorting data.
  {
        for (int i = 0; i < num_persons; i++)
        {
                printf("%d/%s/%s/%s/%d/%s/%s₩n", persons[i].tag, persons[i].date_Registered, persons[i].fee_paid, persons[i].name, persons[i].age, persons[i].organization, persons[i].job);
        }
  }
  // p4
  //made by jun ha cha

  struct RegList_Data_2   //Creating a structure.
  {
        int tag;
        char date_Registered[20];
        char fee_paid[10];
        char name[25];
        int age;
        char organization[35];
        char job[15];
        struct RegList_Data_2* next;
  };

  void copy_element_4(struct RegList_Data_2* dest, struct RegList_Data_2* src) // This function copies the transmitted data to persons
```

```
{
	dest->tag = src->tag;
	strcpy(dest->date_Registered, src->date_Registered);
	strcpy(dest->fee_paid, src->fee_paid);
	strcpy(dest->name, src->name);
	dest->age = src->age;
	strcpy(dest->organization, src->organization);
	strcpy(dest->job, src->job);
}

bool read_File2(const char* fname, struct RegList_Data_2* persons, int* num_persons) //This function reads data into a file.
{
	FILE* pFile;
	pFile = fopen(fname, "r");
	if (pFile == NULL)
	{
		printf("Error : Cannot read file : %s₩n", fname);
		*num_persons = 0;
		return false;
	}
	struct RegList_Data_2 reading_Data;
	int num = 0;
	while (fscanf(pFile, "%d/%[^/]/%[^/]/%[^/]/%d/%[^/]/%s", &reading_Data.tag, &reading_Data.date_Registered, &reading_Data.fee_paid, &reading_Data.name, &reading_Data.age, &reading_Data.organization, &reading_Data.job) == 7)
	{
		copy_element_4(&persons[num], &reading_Data); //Copy the data you read divided according to the format.
		num++;
	}
	fclose(pFile);
	*num_persons = num; // Save the number of data.
	return true;
}

void sort_Data_tagNumber2(struct RegList_Data_2* persons, int num_persons)// This function sort data using 'bubble sort' based on tag number.
{
```

```
	struct RegList_Data_2 temp_Data;
	for (int k = 0; k < num_persons; k++)
	{
		for (int i = 0; i < num_persons - 1; i++)
		{
			if (persons[i].tag > persons[i + 1].tag)// Read the information and send it back if there is a larger tag Number value.
			{
				temp_Data = persons[i];
				persons[i] = persons[i + 1];
				persons[i + 1] = temp_Data;

			}
		}
	}
}

void linking(struct RegList_Data_2* new_node, struct RegList_Data_2* persons) //This function that copies all values as well.
{
	new_node->tag = persons->tag;
	strcpy(new_node->date_Registered, persons->date_Registered);
	strcpy(new_node->fee_paid, persons->fee_paid);
	strcpy(new_node->name, persons->name);
	new_node->age = persons->age;
	strcpy(new_node->organization, persons->organization);
	strcpy(new_node->job, persons->job);
}

int change_linkedList(struct RegList_Data_2* head, struct RegList_Data_2* persons, int* num_persons)//This function that create a linked list using the sorted data.
{
	struct RegList_Data_2* p = head->next, * prev = head;
	while (p) {
		if (p->tag > persons->tag)
		{
			break;
		}
		prev = p;
```

```
                p = p->next;
        }
        struct RegList_Data_2* new_node = (struct RegList_Data_2*)malloc(sizeof(struct RegList_Data_2));

        linking(new_node, persons); // Call a functuon that copies all values as well.

        prev->next = new_node;
        new_node->next = p;
        return 0;
}

void ScanList(struct RegList_Data_2* node1)    // This function print the data.
{
        printf("₩nTraversing the linked list..₩n");
        int cnt = 1;
        struct RegList_Data_2* k = node1->next;
        while (k != NULL) // Repeat until the 'next' field of the connecting linkedlist is not 'NULL'.
        {
                printf("node[%d] %d/%s/%s/%s/%d/%s/%s₩n", cnt, k->tag, k->date_Registered, k->fee_paid, k->name, k->age, k->organization, k->job);
                k = k->next;
                cnt++;
        }
}

// p5
/**
 * @ Author: Cho hyunseo
 * @ Create Time: 2021-11-15 15:32:00
 * @ Modified time: 2021-11-19 23:24:10
 * @ Description: term project p5
 */
struct DATA
{
    int tagNumber;
    char dateRegistered[15];
    char feePaid[5];
    char name[MAX_NAME];
```

```
    int age;
    char organization[MAX_ORGANIZATION];
    char job[MAX_JOB];
};

// function for swap
void swap(struct DATA* x, struct DATA* y)
{
    struct DATA tmp;
    tmp = *x;
    *x = *y;
    *y = tmp;
}

// function for selection sort
void selectionSort(struct DATA data[], int n)
{
    int i, j;
    for (i = 0; i < n - 1; i++)
    {
        int min = i;
        for (j = i + 1; j < n; j++)
        {
            if (data[j].age < data[min].age)
                min = j;
        }
        swap(&data[i], &data[min]);
    }
}

// function for save the data
void saveData(struct DATA data[], char tagNumber[], char dateRegistered[], char feePaid[], char name[], char age[], char organization[], char job[], int n)
{
    data[n].tagNumber = atoi(tagNumber);
    strcpy(data[n].dateRegistered, dateRegistered);
    strcpy(data[n].feePaid, feePaid);
    strcpy(data[n].name, name);
    data[n].age = atoi(age);
```

```
        strcpy(data[n].organization, organization);
        strcpy(data[n].job, job);
}

// function for read a file
bool readFile(char* fileName, struct DATA data[], int max)
{
        FILE* pFile;
        pFile = fopen(fileName, "r");
        if (pFile == NULL)
                return false;
        int n = 0;
        int i, j;
        while (1)
        {
                char tmp[100] = {
                        0,
                };

                fgets(tmp, sizeof(tmp), pFile);
                if (feof(pFile))
                        break;
                char tagNumber[5] = {
                        0,
                };
                char dateRegistered[15] = {
                        0,
                };
                char feePaid[5] = {
                        0,
                };
                char name[MAX_NAME] = {
                        0,
                };
                char age[5] = {
                        0,
                };
                char organization[MAX_ORGANIZATION] = {
                        0,
```

```
};
char job[MAX_JOB] = {
	0,
};
i = 0;
j = 0;
// read a string split by slash
while (tmp[i] != '/')
{
	tagNumber[j] = tmp[i];
	j++;
	i++;
}
i++;
j = 0;
while (tmp[i] != '/')
{
	dateRegistered[j] = tmp[i];
	j++;
	i++;
}
i++;
j = 0;
while (tmp[i] != '/')
{
	feePaid[j] = tmp[i];
	j++;
	i++;
}
i++;
j = 0;
while (tmp[i] != '/')
{
	name[j] = tmp[i];
	j++;
	i++;
}
i++;
j = 0;
```

```
            while (tmp[i] != '/')
            {
                age[j] = tmp[i];
                j++;
                i++;
            }
            i++;
            j = 0;
            while (tmp[i] != '/')
            {
                organization[j] = tmp[i];
                j++;
                i++;
            }
            i++;
            j = 0;
            while (tmp[i] != '₩n')
            {
                job[j] = tmp[i];
                j++;
                i++;
            }
            // save the read data
            saveData(data, tagNumber, dateRegistered, feePaid, name, age, organization, job, n);
            n++;
        }

        fclose(pFile);
        return true;
}

// function for write a file
bool writeFile(char* fileName, struct DATA data[], int max)
{
        FILE* pFile;
        pFile = fopen(fileName, "w");
        if (pFile == NULL)
        {
            printf("Something wrong₩n");
```

```
        return false;
    }
    int i = 0;

    fprintf(pFile, "10's₩n");
    fprintf(pFile, "----------------------------------------------------------₩n");
    // grouping 10's
    while (i < max)
    {
        if (data[i].age >= 20)
            break;
        fprintf(pFile, "%d/%s/%s/%s/%d/%s/%s₩n", data[i].tagNumber, data[i].dateRegistered,
data[i].feePaid, data[i].name, data[i].age, data[i].organization, data[i].job);
        i++;
    }
    // grouping 20's
    fprintf(pFile, "----------------------------------------------------------₩n");
    fprintf(pFile, "20's₩n");
    fprintf(pFile, "----------------------------------------------------------₩n");
    while (i < max)
    {
        if (data[i].age >= 30)
            break;
        fprintf(pFile, "%d/%s/%s/%s/%d/%s/%s₩n", data[i].tagNumber, data[i].dateRegistered,
data[i].feePaid, data[i].name, data[i].age, data[i].organization, data[i].job);
        i++;
    }
    // grouping 30's
    fprintf(pFile, "----------------------------------------------------------₩n");
    fprintf(pFile, "30's₩n");
    fprintf(pFile, "----------------------------------------------------------₩n");
    while (i < max)
    {
        if (data[i].age >= 40)
            break;
        fprintf(pFile, "%d/%s/%s/%s/%d/%s/%s₩n", data[i].tagNumber, data[i].dateRegistered,
data[i].feePaid, data[i].name, data[i].age, data[i].organization, data[i].job);
        i++;
    }
```

```
        // grouping 40's
        fprintf(pFile, "-------------------------------------------------------------₩n");
        fprintf(pFile, "40's₩n");
        fprintf(pFile, "-------------------------------------------------------------₩n");
        while (i < max)
        {
            if (data[i].age >= 50)
                break;
            fprintf(pFile, "%d/%s/%s/%s/%d/%s/%s₩n", data[i].tagNumber, data[i].dateRegistered,
data[i].feePaid, data[i].name, data[i].age, data[i].organization, data[i].job);
            i++;
        }
        // grouping 50's
        fprintf(pFile, "-------------------------------------------------------------₩n");
        fprintf(pFile, "50's₩n");
        fprintf(pFile, "-------------------------------------------------------------₩n");
        while (i < max)
        {
            if (data[i].age >= 60)
                break;
            fprintf(pFile, "%d/%s/%s/%s/%d/%s/%s₩n", data[i].tagNumber, data[i].dateRegistered,
data[i].feePaid, data[i].name, data[i].age, data[i].organization, data[i].job);
            i++;
        }
        // grouping 60's
        fprintf(pFile, "-------------------------------------------------------------₩n");
        fprintf(pFile, "60's₩n");
        fprintf(pFile, "-------------------------------------------------------------₩n");
        while (i < max)
        {
            if (data[i].age >= 70)
                break;
            fprintf(pFile, "%d/%s/%s/%s/%d/%s/%s₩n", data[i].tagNumber, data[i].dateRegistered,
data[i].feePaid, data[i].name, data[i].age, data[i].organization, data[i].job);
            i++;
        }
        // grouping 70's
        fprintf(pFile, "-------------------------------------------------------------₩n");
        fprintf(pFile, "70's₩n");
```

```
        fprintf(pFile, "----------------------------------------------------------------₩n");
        while (i < max)
        {
            if (data[i].age >= 80)
                break;
            fprintf(pFile, "%d/%s/%s/%s/%d/%s/%s₩n", data[i].tagNumber, data[i].dateRegistered,
data[i].feePaid, data[i].name, data[i].age, data[i].organization, data[i].job);
            i++;
        }
        fprintf(pFile, "----------------------------------------------------------------₩n");
        fclose(pFile);
        return true;
    }

    // p6
    // Ban Jae Hyeon
    // p6-1
    struct {
        char row[100] = { NULL };
        char name[30] = { NULL };
    }data_6[31];

    // p6-2

    struct data_6_2 {
        char name[30] = { NULL };
        char row[100] = { NULL };
        data_6_2* next = NULL;
    };

    void Insert(data_6_2* first, char n[30], char r[100]) // add node
    {
        while (first->next) // until there is no next node
        {
            first = first->next; // to next node
        }
        data_6_2* new_node = (data_6_2*)malloc(sizeof(data_6_2)); // create node
        first->next = new_node; // connect the node
        strcpy(new_node->name, n); // save the value of node
```

```
        strcpy(new_node->row, r);;
        new_node->next = NULL;
    }

    void Print(data_6_2* first) // print function
    {
        while (first)
        {
            printf("%s", first->row);
            first = first->next;
        }
    }

    void Choierase(data_6_2* head)// erase choi
    {
        data_6_2* prev = NULL;
        while (1)
        {
            int a = 0, exit = 0;
            while (head->name[a] != '₩0') // check the name
            {
                if (head->name[a] == 'C' && head->name[a + 1] == 'h' && head->name[a + 2]
== 'o' && head->name[a + 3] == 'i') // check 'Choi'
                {
                    prev->next = head->next; // reconnect the node
                    free(head); // erase node
                    head = prev->next;
                    exit = 1;
                    break;
                }
                a++;
            }
            if (head->next == NULL)
            {
                break;
            }
            if (exit == 0)
            {
                prev = head;
```

```
                head = head->next;
            }
        }
}
// p7
// Ban Jae Hyeon

struct Data_7
{
    int tag;
    char date[20];
    char fee[10];
    char name[25];
    int age;
    char organ[35];
    char job[15];
    Data_7* next;
};

void Paikadd(Data_7* head)// add Paik
{
    while (head->next)
    {
        head = head->next;
    }
    head->next = (Data_7*)malloc(sizeof(Data_7)); // create node
    head = head->next;
    head->tag = 31;// enter tag
    head->next = NULL;
    strcpy(head->date, "2020-11-30");
    strcpy(head->fee, "yes");
    strcpy(head->name, "Gildong Paik");
    head->age = 35;
    strcpy(head->organ, "Gachon University");
    strcpy(head->job, "Student");
}

void Print_7(Data_7* head)// print
{
```

```
        while (head)
        {
            printf("%d/%s/%s/%s/%d/%s/%s₩n", head->tag, head->date, head->fee, head->name,
head->age, head->organ, head->job);
            head = head->next;
        }
    }
    // p8
    /**
     * @ Author: Cho hyunseo
     * @ Create Time: 2021-11-16 11:18:27
     * @ Modified time: 2021-11-19 23:27:36
     * @ Description: term project p8
     */
     // function for compute the checksum
    int computeChecksum(char* name)
    {
        char ch;
        int checksum = 0;
        int i = 0;
        while (name[i] != '₩0')
        {
            ch = name[i];
            if (checksum == 0)
            {
                checksum = ch;
            }
            else
            {
                checksum = checksum ^ ch;
            }
            i++;
        }

        return checksum;
    }


    // function for read a file
```

```
bool readFile2(char* fileName, struct DATA data[], int max)
{
	FILE* pFile;
	pFile = fopen(fileName, "r");
	if (pFile == NULL)
		return false;
	int n = 0;
	int i, j;
	while (n < MAX)
	{
		char tmp[100] = {
			0,
		};

		fgets(tmp, sizeof(tmp), pFile);
		if (feof(pFile))
			break;
		if (n >= max)
			break;
		char tagNumber[5] = {
			0,
		};
		char dateRegistered[15] = {
			0,
		};
		char feePaid[5] = {
			0,
		};
		char name[MAX_NAME] = {
			0,
		};
		char age[5] = {
			0,
		};
		char organization[MAX_ORGANIZATION] = {
			0,
		};
		char job[MAX_JOB] = {
			0,
```

```
};
i = 0;
j = 0;
// read a string split by slash
while (tmp[i] != '/')
{
	tagNumber[j] = tmp[i];
	j++;
	i++;
}
i++;
j = 0;
while (tmp[i] != '/')
{
	dateRegistered[j] = tmp[i];
	j++;
	i++;
}
i++;
j = 0;
while (tmp[i] != '/')
{
	feePaid[j] = tmp[i];
	j++;
	i++;
}
i++;
j = 0;
while (tmp[i] != '/')
{
	name[j] = tmp[i];
	j++;
	i++;
}
i++;
j = 0;
while (tmp[i] != '/')
{
	age[j] = tmp[i];
```

```
            j++;
            i++;
        }
        i++;
        j = 0;
        while (tmp[i] != '/')
        {
            organization[j] = tmp[i];
            j++;
            i++;
        }
        i++;
        j = 0;
        while (tmp[i] != '₩n')
        {
            job[j] = tmp[i];
            j++;
            i++;
        }
        // save the read data
        saveData(data, tagNumber, dateRegistered, feePaid, name, age, organization, job, n);
        n++;
    }

    fclose(pFile);
    return true;
}

// function for write a file
bool writeFile2(char* fileName, struct DATA data[], int min, int max, int checksum)
{
    FILE* pFile;
    pFile = fopen(fileName, "w");
    if (pFile == NULL)
    {
        printf("Something wrong₩n");
        return false;
    }
    int i = min;
```

```
        while (i < max)
        {
            fprintf(pFile, "%d/%s/%s/%s/%d/%s/%s₩n", data[i].tagNumber, data[i].dateRegistered,
data[i].feePaid, data[i].name, data[i].age, data[i].organization, data[i].job);
            i++;
        }
        fprintf(pFile, "%d₩n", checksum);
        fclose(pFile);
        return true;
    }
    int main(void)
    {
        printf("₩n₩n-------------------------p1-1----------------------------₩n₩n");
          // p1
        PERSONAL persons_array[MAX_PERSON];
        PERSONAL* persons1 = NULL;

        char search_name[MAX_NAME_LEN] = "Choi";
        char* name_key = search_name;
        //read file in array
        if (!read_File_1_array("registraion_data.txt", persons_array))
        {
            return -1;
        }
        //P1-1. Search for ��Choi�� in the array
        printf("Information--₩"%s₩"₩n", search_name);
        for (int t = 0; t < MAX_PERSON; t++)
        {
            PERSONAL* ptr = &persons_array[t];

            if (!SearchNameInArr(ptr, name_key)) {
                return -1;
            }
        }
        if (!read_File_1_linked("registraion_data.txt", &persons1))
        {
            return -1;
        }
        printf("₩n₩n-------------------------p1-2----------------------------₩n₩n");
```

```
//P1-2. Print all information   who named "Choi" in linked list
printf("Information--₩"%s₩"₩n", search_name);
if (!SearchNameInLL(persons1, name_key)) {
	return -1;
}
  // p2
printf("₩n₩n-------------------------p2-1----------------------------₩n₩n");
PERSONAL persons_array_2[MAX_PERSON];
PERSONAL* persons2 = NULL;
char search_org[MAX_ORG_LEN] = "Gachon University";
char* org_key = search_org;
//P2-1. Read file
if (!read_File_2_array("registraion_data.txt", persons_array_2)) {
	return -1;
}
//P2-1. Print all information whose organization is "Gachon University" using array
printf("Matched information--₩"%s₩"₩n", search_org);
for (int t = 0; t < MAX_PERSON; t++)
{
	PERSONAL* ptr = &persons_array_2[t];

	if (!SearchOrgInArr(ptr, org_key)) {
		return -1;
	}
}
printf("₩n₩n-------------------------p2-2----------------------------₩n₩n");
//P2-2. Read file
if (!read_File_2_linked("registraion_data.txt", &persons2))
{
	return -1;
}
//P2-2. Print all information   whose organization is "Gachon University" using linked list
printf("Matched information--₩"%s₩"₩n", search_org);
if (!SearchOrgInLL(persons2, search_org)) {
	return -1;
}
  // p3
printf("₩n₩n-------------------------p3----------------------------₩n₩n");
  struct RegList_Data_1 persons[30];
```

```
	int num_persons = 0;
	if ((read_File("registraion_data.txt", persons, &num_persons)) != true) //Call a function that reads information inside the txt file.
	{
		return false;
	}
	sort_Data_tagNumber(persons, num_persons);   // Call a function that sorts the read data.
	print_Data(persons, num_persons); // Calls a function that outputs sorted data.
	// p4
	printf("₩n₩n-------------------------p4----------------------------₩n₩n");
	struct RegList_Data_2 persons_2[30];
	int num_persons2 = 0;
	if ((read_File2("registraion_data.txt", persons_2, &num_persons2)) != true) //Call a function that reads information inside the txt file.
	{
		return false;
	}
	sort_Data_tagNumber2(persons_2, num_persons2); // Call a function that sorts the read data.

	struct RegList_Data_2* head = (struct RegList_Data_2*)malloc(sizeof(struct RegList_Data_2));
	struct RegList_Data_2* node1 = (struct RegList_Data_2*)malloc(sizeof(struct RegList_Data_2));

	head->next = node1;
	node1->next = NULL;

	for (int i = 0; i < 30; i++)
	{
		change_linkedList(head, &persons_2[i], &num_persons2);   // Call a function that create a linked list using the sorted data.
	}

	ScanList(node1); // Call a function that print a linked list using the sorted data
	// p5
	printf("₩n₩n-------------------------p5----------------------------₩n₩n");
	struct DATA data[MAX];
	char fileName[MAX] = {
	    0,
	};
```

```
// set the file name
strcpy(fileName, "registraion_data.txt");
if (!readFile(fileName, data, MAX))
	return -1;
// sorting
selectionSort(data, MAX);
char writeFileName[MAX] = {
	0,
};
// set the file name
strcpy(writeFileName, "P5-1.txt");
if (!writeFile(writeFileName, data, MAX))
	return -1;
   // p6

printf("₩n₩n-------------------------p6-1----------------------------₩n₩n");
int i = 0;
int a = 0, k = 0, j = 0;
char alpha;
FILE* regist;
regist = fopen("registraion_data.txt", "r");
while (1)
{
	alpha = fgetc(regist);
	data_6[i].row[j] = alpha;
	j++;
	if (alpha == EOF)
	{
		break;
	}
	else if (alpha == '₩n')
	{
		a = 0;
		j = 0;
		i++;
	}
	else if (alpha == '/')
	{
		if (a == 3)
```

```
                {
                    data_6[i].name[k] = '₩0';
                }
                k = 0;
                a++;
                continue;
            }
            if (a == 3)
            {
                data_6[i].name[k] = alpha;
                k++;
            }
        }
        i = 0;
        while (1)
        {
            a = 0;
            while (data_6[i].name[a] != '₩0')
            {
                if (data_6[i].name[a] == 'C' && data_6[i].name[a + 1] == 'h' && data_6[i].name[a +
2] == 'o' && data_6[i].name[a + 3] == 'i')
                {
                    for (k = i; k < 30; k++)
                    {
                        strcpy(data_6[k].row, data_6[k + 1].row);
                        strcpy(data_6[k].name, data_6[k + 1].name);
                    }
                    continue;
                }
                a++;
            }
            i++;
            if (data_6[i].name[0] == '₩0')
                break;
        }
        for (i = 0; i < 30; i++)
        {
            printf("%s", data_6[i].row);
        }
```

```
fclose(regist);
printf("₩n₩n-------------------------p6-2----------------------------₩n₩n");
char alpha_2, n[30] = { NULL }, line[100] = { NULL };
data_6_2 first;
data_6_2* head_6 = &first;
FILE* regist_2;
regist_2 = fopen("registraion_data.txt", "r");
i = 0;
a = 0, k = 0, j = 0;
while (1) // for save
{
	alpha_2 = fgetc(regist_2); // get
	line[j] = alpha_2; // save
	j++;
	if (alpha_2 == EOF) // end condition
	{
		break;
	}
	else if (alpha_2 == '₩n') // to the next line
	{
		line[j] = '₩0';
		a = 0;
		j = 0;
		if (i == 0)// make head
		{
			strcpy(first.row, line);
			strcpy(first.name, n);
		}
		else
		{
			Insert(head_6, n, line);// input
		}
		i++;
	}
	else if (alpha_2 == '/')// check slash
	{
		if (a == 3)// if finish entering name
		{
			n[k] = '₩0'; // enter ₩0
```

```
            }
            k = 0;
            a++;
            continue;
        }
        if (a == 3) // entering the name
        {
            n[k] = alpha_2;
            k++;
        }
    }
    i = 0;
    Choierase(head_6); // erase Choi

    Print(head_6); // print
    fclose(regist_2);
      // p7
    printf("₩n₩n-------------------------p7-1----------------------------₩n₩n");
    struct Data
    {
        int tag;
        char date[20];
        char fee[10];
        char name[25];
        int age;
        char organ[35];
        char job[15];
    }man[31];

    i = 0;
    FILE* regist_7;
    regist_7 = fopen("registraion_data.txt", "r");
    for (i = 0; i < 30; i++)
    {
        fscanf(regist_7, "%d/%[^/]/%[^/]/%[^/]/%d/%[^/]/%s", &man[i].tag, &man[i].date,
&man[i].fee, &man[i].name, &man[i].age, &man[i].organ, &man[i].job);

    }
    //add Paik
```

```
	man[30].tag = 31;
	strcpy(man[30].date, "2020-11-30");
	strcpy(man[30].fee, "yes");
	strcpy(man[30].name, "Gildong Paik");
	man[30].age = 35;
	strcpy(man[30].organ, "Gachon University");
	strcpy(man[30].job, "Student");

	for (int i = 0; i < 31; i++)// print
	{
		printf("%d/%s/%s/%s/%d/%s/%s₩n", man[i].tag, man[i].date, man[i].fee, man[i].name, man[i].age, man[i].organ, man[i].job);
	}
	fclose(regist_7);

	printf("₩n₩n-------------------------p7-2----------------------------₩n₩n");
	Data_7 first_7;
	Data_7* head_7 = &first_7;
	Data_7* now = head_7;
	first.next = NULL;
	FILE* regist_7_2;
	regist_7_2 = fopen("registraion_data.txt", "r");
	Data_7* new_node;
	fscanf(regist_7_2, "%d/%[^/]/%[^/]/%[^/]/%d/%[^/]/%s", &now->tag, &now->date, &now->fee, &now->name, &now->age, &now->organ, &now->job);
	while (1)
	{
		new_node = (Data_7*)malloc(sizeof(Data_7)); // create node
		if (fscanf(regist_7_2, "%d/%[^/]/%[^/]/%[^/]/%d/%[^/]/%s", &new_node->tag, &new_node->date, &new_node->fee, &new_node->name, &new_node->age, &new_node->organ, &new_node->job) != 7)
		{
			break;
		}
		now->next = new_node;// connect node
		now = now->next; // to the next node
		now->next = NULL;
	}
	fclose(regist_7_2);
```

```
now = head_7;
Paikadd(head_7); // add Paik
Print_7(head_7);// print

   // p8
printf("₩n₩n-------------------------p8----------------------------₩n₩n");
int checksum = 0;
int copiedChecksum = 0;
struct DATA data2[MAX];
struct DATA copiedData[MAX];
char fileName2[MAX] = {
      0,
};
strcpy(fileName2, "registraion_data.txt");
if (!readFile2(fileName2, data2, MAX))
      return -1;

// calculate original's checksum
for (i = MAX - 10; i < MAX; i++)
{
      checksum ^= computeChecksum(data2[i].name);
}

char writeFile2Name[MAX] = {
      0,
};
strcpy(writeFile2Name, "P8-1.txt");
if (!writeFile2(writeFile2Name, data2, MAX - 10, MAX, checksum))
      return -1;

char copiedFileName[MAX] = {
      0,
};
strcpy(copiedFileName, "P8-1.txt");
if (!readFile2(copiedFileName, copiedData, 10))
      return -1;

// calculate copied's checksum
for (i = 0; i < 10; i++)
```

```
    {
        copiedChecksum ^= computeChecksum(copiedData[i].name);
    }
    // compare two checksum
    printf("Original checksum: %d₩n", checksum);
    printf("Copied checksum: %d₩n", copiedChecksum);
    printf("Is same?: ");
    if (checksum == copiedChecksum)
        printf("Yes₩n");
    else
        printf("No₩n");


    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

```
------------------------p1-1-----------------------------

Information--"Choi"
11/2020-07-22/no/Kwangsu Choi/48/Seoul National University/marketer

15/2020-07-12/no/Tongbang Choi/26/Cornell University/engineer

1/2020-08-25/yes/Jihu Choi/74/Harvard University/engineer

30/2020-07-13/yes/Kyungmin Choi/44/Duke University/student

2/2020-08-22/no/Seungmin Choi/31/Gachon University/staff
```

```
------------------------p1-2-----------------------------

Information--"Choi"
11/2020-07-22/no/Kwangsu Choi/48/Seoul National University/marketer

15/2020-07-12/no/Tongbang Choi/26/Cornell University/engineer

1/2020-08-25/yes/Jihu Choi/74/Harvard University/engineer

30/2020-07-13/yes/Kyungmin Choi/44/Duke University/student

2/2020-08-22/no/Seungmin Choi/31/Gachon University/staff
```

```
-------------------------p2-1-----------------------------

Matched information--"Gachon University"
29/2020-06-08/yes/Bailey Houghton/31/Gachon University/engineer

12/2020-07-22/no/Owen Martin/66/Gachon University/engineer

8/2020-06-04/no/Moises Barlow/57/Gachon University/engineer

14/2020-08-15/yes/Kwangsu Cho/48/Gachon University/executive

27/2020-08-24/no/Konner French/42/Gachon University/professor

17/2020-08-14/no/Chunyong Chang/75/Gachon University/student

2/2020-08-22/no/Seungmin Choi/31/Gachon University/staff

13/2020-06-03/yes/Chinho Cho/68/Gachon University/student

20/2020-07-30/yes/Chinho Kim/52/Gachon University/engineer


-------------------------p2-2-----------------------------

Matched information--"Gachon University"
29/2020-06-08/yes/Bailey Houghton/31/Gachon University/engineer

12/2020-07-22/no/Owen Martin/66/Gachon University/engineer

8/2020-06-04/no/Moises Barlow/57/Gachon University/engineer

14/2020-08-15/yes/Kwangsu Cho/48/Gachon University/executive

27/2020-08-24/no/Konner French/42/Gachon University/professor

17/2020-08-14/no/Chunyong Chang/75/Gachon University/student

2/2020-08-22/no/Seungmin Choi/31/Gachon University/staff

13/2020-06-03/yes/Chinho Cho/68/Gachon University/student

20/2020-07-30/yes/Chinho Kim/52/Gachon University/engineer
```

```
-------------------------p3-----------------------------

1/2020-08-25/yes/Jihu Choi/74/Harvard University/engineer
2/2020-08-22/no/Seungmin Choi/31/Gachon University/staff
3/2020-07-01/no/Chinho Park/53/Peking University/engineer
4/2020-07-03/no/Jihu Cho/71/Tsinghua University/engineer
5/2020-06-12/yes/Chunyong Park/48/University of Cambridge/student
6/2020-06-04/yes/Bobby Anderson/33/McGill University/engineer
7/2020-06-28/yes/Jihu Park/70/Australian National University/student
8/2020-06-04/no/Moises Barlow/57/Gachon University/engineer
9/2020-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
10/2020-06-06/yes/William Cohen/37/University of Cambridge/engineer
11/2020-07-22/no/Kwangsu Choi/48/Seoul National University/marketer
12/2020-07-22/no/Owen Martin/66/Gachon University/engineer
13/2020-06-03/yes/Chinho Cho/68/Gachon University/student
14/2020-08-15/yes/Kwangsu Cho/48/Gachon University/executive
15/2020-07-12/no/Tongbang Choi/26/Cornell University/engineer
16/2020-08-16/yes/Tongbang Kim/39/Tsinghua University/student
17/2020-08-14/no/Chunyong Chang/75/Gachon University/student
18/2020-06-14/no/Tongbang Park/32/New York University/engineer
19/2020-06-07/yes/Chunyong Kim/34/Harvard University/staff
20/2020-07-30/yes/Chinho Kim/52/Gachon University/engineer
21/2020-07-21/yes/Jude Smith/38/Cornell University/executive
22/2020-06-29/no/Tongbang Cho/29/Northwestern University/marketer
23/2020-06-15/yes/Seungmin Cho/71/Stanford University/professor
24/2020-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
25/2020-06-09/no/Archie Hunt/60/Fudan University/student
26/2020-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
27/2020-08-24/no/Konner French/42/Gachon University/professor
28/2020-08-27/no/Kwangsu Park/43/University of Pennsylvania/student
29/2020-06-08/yes/Bailey Houghton/31/Gachon University/engineer
30/2020-07-13/yes/Kyungmin Choi/44/Duke University/student


-------------------------p4-----------------------------

Traversing the linked list..
node[1] 1/2020-08-25/yes/Jihu Choi/74/Harvard University/engineer
node[2] 2/2020-08-22/no/Seungmin Choi/31/Gachon University/staff
node[3] 3/2020-07-01/no/Chinho Park/53/Peking University/engineer
node[4] 4/2020-07-03/no/Jihu Cho/71/Tsinghua University/engineer
node[5] 5/2020-06-12/yes/Chunyong Park/48/University of Cambridge/student
node[6] 6/2020-06-04/yes/Bobby Anderson/33/McGill University/engineer
node[7] 7/2020-06-28/yes/Jihu Park/70/Australian National University/student
node[8] 8/2020-06-04/no/Moises Barlow/57/Gachon University/engineer
node[9] 9/2020-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
node[10] 10/2020-06-06/yes/William Cohen/37/University of Cambridge/engineer
node[11] 11/2020-07-22/no/Kwangsu Choi/48/Seoul National University/marketer
node[12] 12/2020-07-22/no/Owen Martin/66/Gachon University/engineer
node[13] 13/2020-06-03/yes/Chinho Cho/68/Gachon University/student
node[14] 14/2020-08-15/yes/Kwangsu Cho/48/Gachon University/executive
node[15] 15/2020-07-12/no/Tongbang Choi/26/Cornell University/engineer
node[16] 16/2020-08-16/yes/Tongbang Kim/39/Tsinghua University/student
node[17] 17/2020-08-14/no/Chunyong Chang/75/Gachon University/student
node[18] 18/2020-06-14/no/Tongbang Park/32/New York University/engineer
node[19] 19/2020-06-07/yes/Chunyong Kim/34/Harvard University/staff
node[20] 20/2020-07-30/yes/Chinho Kim/52/Gachon University/engineer
node[21] 21/2020-07-21/yes/Jude Smith/38/Cornell University/executive
node[22] 22/2020-06-29/no/Tongbang Cho/29/Northwestern University/marketer
node[23] 23/2020-06-15/yes/Seungmin Cho/71/Stanford University/professor
node[24] 24/2020-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
node[25] 25/2020-06-09/no/Archie Hunt/60/Fudan University/student
node[26] 26/2020-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
node[27] 27/2020-08-24/no/Konner French/42/Gachon University/professor
node[28] 28/2020-08-27/no/Kwangsu Park/43/University of Pennsylvania/student
node[29] 29/2020-06-08/yes/Bailey Houghton/31/Gachon University/engineer
node[30] 30/2020-07-13/yes/Kyungmin Choi/44/Duke University/student
```

P5-1 - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

10's

20's

15/2020-07-12/no/Tongbang Choi/26/Cornell University/engineer
22/2020-06-29/no/Tongbang Cho/29/Northwestern University/marketer
---
30's

29/2020-06-08/yes/Bailey Houghton/31/Gachon University/engineer
2/2020-08-22/no/Seungmin Choi/31/Gachon University/staff
18/2020-06-14/no/Tongbang Park/32/New York University/engineer
6/2020-06-04/yes/Bobby Anderson/33/McGill University/engineer
19/2020-06-07/yes/Chunyong Kim/34/Harvard University/staff
10/2020-06-06/yes/William Cohen/37/University of Cambridge/engineer
21/2020-07-21/yes/Jude Smith/38/Cornell University/executive
16/2020-08-16/yes/Tongbang Kim/39/Tsinghua University/student
---
40's

27/2020-08-24/no/Konner French/42/Gachon University/professor
28/2020-08-27/no/Kwangsu Park/43/University of Pennsylvania/student
30/2020-07-13/yes/Kyungmin Choi/44/Duke University/student
9/2020-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
14/2020-08-15/yes/Kwangsu Cho/48/Gachon University/executive
11/2020-07-22/no/Kwangsu Choi/48/Seoul National University/marketer
24/2020-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
5/2020-06-12/yes/Chunyong Park/48/University of Cambridge/student
---
50's

20/2020-07-30/yes/Chinho Kim/52/Gachon University/engineer
3/2020-07-01/no/Chinho Park/53/Peking University/engineer
8/2020-06-04/no/Moises Barlow/57/Gachon University/engineer
26/2020-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
---
60's

25/2020-06-09/no/Archie Hunt/60/Fudan University/student
12/2020-07-22/no/Owen Martin/66/Gachon University/engineer
13/2020-06-03/yes/Chinho Cho/68/Gachon University/student

70's

7/2020-06-28/yes/Jihu Park/70/Australian National University/student
23/2020-06-15/yes/Seungmin Cho/71/Stanford University/professor
4/2020-07-03/no/Jihu Cho/71/Tsinghua University/engineer
1/2020-08-25/yes/Jihu Choi/74/Harvard University/engineer
17/2020-08-14/no/Chunyong Chang/75/Gachon University/student

```
-------------------------p6-1-----------------------------

6/2020-06-04/yes/Bobby Anderson/33/McGill University/engineer
5/2020-06-12/yes/Chunyong Park/48/University of Cambridge/student
22/2020-06-29/no/Tongbang Cho/29/Northwestern University/marketer
7/2020-06-28/yes/Jihu Park/70/Australian National University/student
16/2020-08-16/yes/Tongbang Kim/39/Tsinghua University/student
21/2020-07-21/yes/Jude Smith/38/Cornell University/executive
29/2020-06-08/yes/Bailey Houghton/31/Gachon University/engineer
25/2020-06-09/no/Archie Hunt/60/Fudan University/student
10/2020-06-06/yes/William Cohen/37/University of Cambridge/engineer
24/2020-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
4/2020-07-03/no/Jihu Cho/71/Tsinghua University/engineer
12/2020-07-22/no/Owen Martin/66/Gachon University/engineer
8/2020-06-04/no/Moises Barlow/57/Gachon University/engineer
9/2020-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
18/2020-06-14/no/Tongbang Park/32/New York University/engineer
23/2020-06-15/yes/Seungmin Cho/71/Stanford University/professor
14/2020-08-15/yes/Kwangsu Cho/48/Gachon University/executive
19/2020-06-07/yes/Chunyong Kim/34/Harvard University/staff
27/2020-08-24/no/Konner French/42/Gachon University/professor
26/2020-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
17/2020-08-14/no/Chunyong Chang/75/Gachon University/student
3/2020-07-01/no/Chinho Park/53/Peking University/engineer
13/2020-06-03/yes/Chinho Cho/68/Gachon University/student
28/2020-08-27/no/Kwangsu Park/43/University of Pennsylvania/student
20/2020-07-30/yes/Chinho Kim/52/Gachon University/engineer


-------------------------p6-2-----------------------------

6/2020-06-04/yes/Bobby Anderson/33/McGill University/engineer
5/2020-06-12/yes/Chunyong Park/48/University of Cambridge/student
22/2020-06-29/no/Tongbang Cho/29/Northwestern University/marketer
7/2020-06-28/yes/Jihu Park/70/Australian National University/student
16/2020-08-16/yes/Tongbang Kim/39/Tsinghua University/student
21/2020-07-21/yes/Jude Smith/38/Cornell University/executive
29/2020-06-08/yes/Bailey Houghton/31/Gachon University/engineer
25/2020-06-09/no/Archie Hunt/60/Fudan University/student
10/2020-06-06/yes/William Cohen/37/University of Cambridge/engineer
24/2020-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
4/2020-07-03/no/Jihu Cho/71/Tsinghua University/engineer
12/2020-07-22/no/Owen Martin/66/Gachon University/engineer
8/2020-06-04/no/Moises Barlow/57/Gachon University/engineer
9/2020-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
18/2020-06-14/no/Tongbang Park/32/New York University/engineer
23/2020-06-15/yes/Seungmin Cho/71/Stanford University/professor
14/2020-08-15/yes/Kwangsu Cho/48/Gachon University/executive
19/2020-06-07/yes/Chunyong Kim/34/Harvard University/staff
27/2020-08-24/no/Konner French/42/Gachon University/professor
26/2020-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
17/2020-08-14/no/Chunyong Chang/75/Gachon University/student
3/2020-07-01/no/Chinho Park/53/Peking University/engineer
13/2020-06-03/yes/Chinho Cho/68/Gachon University/student
28/2020-08-27/no/Kwangsu Park/43/University of Pennsylvania/student
20/2020-07-30/yes/Chinho Kim/52/Gachon University/engineer
```

```
------------------------p7-1----------------------------

6/2020-06-04/yes/Bobby Anderson/33/McGill University/engineer
5/2020-06-12/yes/Chunyong Park/48/University of Cambridge/student
11/2020-07-22/no/Kwangsu Choi/48/Seoul National University/marketer
22/2020-06-29/no/Tongbang Cho/29/Northwestern University/marketer
7/2020-06-28/yes/Jihu Park/70/Australian National University/student
16/2020-08-16/yes/Tongbang Kim/39/Tsinghua University/student
21/2020-07-21/yes/Jude Smith/38/Cornell University/executive
29/2020-06-08/yes/Bailey Houghton/31/Gachon University/engineer
25/2020-06-09/no/Archie Hunt/60/Fudan University/student
10/2020-06-06/yes/William Cohen/37/University of Cambridge/engineer
24/2020-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
4/2020-07-03/no/Jihu Cho/71/Tsinghua University/engineer
12/2020-07-22/no/Owen Martin/66/Gachon University/engineer
8/2020-06-04/no/Moises Barlow/57/Gachon University/engineer
9/2020-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
18/2020-06-14/no/Tongbang Park/32/New York University/engineer
23/2020-06-15/yes/Seungmin Cho/71/Stanford University/professor
14/2020-08-15/yes/Kwangsu Cho/48/Gachon University/executive
19/2020-06-07/yes/Chunyong Kim/34/Harvard University/staff
15/2020-07-12/no/Tongbang Choi/26/Cornell University/engineer
27/2020-08-24/no/Konner French/42/Gachon University/professor
26/2020-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
17/2020-08-14/no/Chunyong Chang/75/Gachon University/student
1/2020-08-25/yes/Jihu Choi/74/Harvard University/engineer
3/2020-07-01/no/Chinho Park/53/Peking University/engineer
30/2020-07-13/yes/Kyungmin Choi/44/Duke University/student
2/2020-08-22/no/Seungmin Choi/31/Gachon University/staff
13/2020-06-03/yes/Chinho Cho/68/Gachon University/student
28/2020-08-27/no/Kwangsu Park/43/University of Pennsylvania/student
20/2020-07-30/yes/Chinho Kim/52/Gachon University/engineer
31/2020-11-30/yes/Gildong Paik/35/Gachon University/Student



------------------------p7-2----------------------------

6/2020-06-04/yes/Bobby Anderson/33/McGill University/engineer
5/2020-06-12/yes/Chunyong Park/48/University of Cambridge/student
11/2020-07-22/no/Kwangsu Choi/48/Seoul National University/marketer
22/2020-06-29/no/Tongbang Cho/29/Northwestern University/marketer
7/2020-06-28/yes/Jihu Park/70/Australian National University/student
16/2020-08-16/yes/Tongbang Kim/39/Tsinghua University/student
21/2020-07-21/yes/Jude Smith/38/Cornell University/executive
29/2020-06-08/yes/Bailey Houghton/31/Gachon University/engineer
25/2020-06-09/no/Archie Hunt/60/Fudan University/student
10/2020-06-06/yes/William Cohen/37/University of Cambridge/engineer
24/2020-07-24/no/Stefan Wilkerson/48/University of Melbourne/executive
4/2020-07-03/no/Jihu Cho/71/Tsinghua University/engineer
12/2020-07-22/no/Owen Martin/66/Gachon University/engineer
8/2020-06-04/no/Moises Barlow/57/Gachon University/engineer
9/2020-06-16/yes/Kyungmin Kim/45/University of Sydney/marketer
18/2020-06-14/no/Tongbang Park/32/New York University/engineer
23/2020-06-15/yes/Seungmin Cho/71/Stanford University/professor
14/2020-08-15/yes/Kwangsu Cho/48/Gachon University/executive
19/2020-06-07/yes/Chunyong Kim/34/Harvard University/staff
15/2020-07-12/no/Tongbang Choi/26/Cornell University/engineer
27/2020-08-24/no/Konner French/42/Gachon University/professor
26/2020-06-30/yes/Sincere Bradley/58/University of Hong Kong/staff
17/2020-08-14/no/Chunyong Chang/75/Gachon University/student
1/2020-08-25/yes/Jihu Choi/74/Harvard University/engineer
3/2020-07-01/no/Chinho Park/53/Peking University/engineer
30/2020-07-13/yes/Kyungmin Choi/44/Duke University/student
2/2020-08-22/no/Seungmin Choi/31/Gachon University/staff
13/2020-06-03/yes/Chinho Cho/68/Gachon University/student
28/2020-08-27/no/Kwangsu Park/43/University of Pennsylvania/student
20/2020-07-30/yes/Chinho Kim/52/Gachon University/engineer
31/2020-11-30/yes/Gildong Paik/35/Gachon University/Student
```

```
------------------------p8-----------------------------

Original checksum: 14
Copied checksum: 14
Is same?: Yes

C:₩Users₩user₩Desktop₩term_project_Team_H₩term_project_Team_H₩Debug₩term_project_Team_H.exe(프로세스 9124개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```