Term project

Team 9

201835546 Ham geonwook

202035318 Kim wonjong

202035352 Ahn hyunjin

201834735 Lee jiheon

Software development Process:

Step 1: understand the problem

(1)setup

read file, store the data (1.array 2.linked list)

file open→array {data1, data2, data3}, store data number

file open→linked list; head->data1->data2->data3

-cases to consider

no data open, data exceeded(array)

(2) Search for "Choi"

Search from registration data and array structure

If person start with "Choi" found, print all information about him

- Search in the array **P1-1**
- Search in the linked list **P1-2**

Case to consider : no name start with "Choi", can't find file to open

(3) Search for all from Gachon University

Search from registration data and array structure

If Gachon University found, print all information about person include Gachon University

- Search in the array **P2-1**
- Search in the linked list **P2-2**

Case to consider : no case of Gachon University, can't find file to open

(4)Sort the data in the array in tag# order - **P3-1**

data1.tag = 2, data2.tag = 3, data3.tag = 1

{data1, data2, data3} → sort → {data3, data1, data2}

-cases to consider

no data, same tag

(5)Create a linked list using the sorted data - **P4-1**

{data3, data1, data2} → head->data3->data1->data2

-cases to consider

no data

(6) – **P5-1**

Sort the data in the array in age group order(using selection sort) & Write the sorted data to a text file.

Copy array and use bubble sorting, sort the data in the array.

->print at textfile 5-1.txt

Ex) data5.age = 70, data6.age = 39, data7.age = 38

{data5, data6, data7} → sort → {data7, data6, data5} → print at 5-1txt

-cases to consider-

No data, Same age

(7) – **P6-1**

All "Choi"s canceled registration. Remove the data from array. Print result

If data name include Choi, delete data. Save array. Print array.

Ex) {data1, data2, data3} → data2 include "Choi" → {data1,data3}

-cases to consider-

Choi is continuous, similar Cho

- P6-2

All "Choi"s canceled registration. Remove the data from the linked list. Print result

If data name include Choi, delete node. Save list. Print list.

Ex) {data1, data2, data3} → data2 include "Choi" → {data1,data3}

-cases to consider-

Choi is continuous, similar Cho

(8) – **P7-1**

One "Paik" registered late. Add the data to the array

1. Even after adding the data, need to keep the ascending order by age.
2. All affected data will be moved.
3. Print result

Paik's information is as follows.

- 100/2020-11-30/yes/Ildang Paik/22/Gachon University/engineer

ex) 100/2020-11-30/yes/Ildang Paik/22/Gachon University/engineer ← add data

22/2020-06-29/no/Tongbang Cho/29/Northwestern University/marketer

5/2020-06-12/yes/Chunyong Park/48/University of Cambridge/student

23/2020-06-15/yes/Seungmin Cho/71/Stanford University/professor

- **P7-2**

One "Paik" registered late. Add the data to the linked list

1. Even after adding the data, need to keep the ascending order by tag.
2. Print result

Paik's information is as follows.

- 100/2020-11-30/yes/Ildang Paik/22/Gachon University/engineer

ex) 5/2020-06-12/yes/Chunyong Park/48/University of Cambridge/student

22/2020-06-29/no/Tongbang Cho/29/Northwestern University/marketer

23/2020-06-15/yes/Seungmin Cho/71/Stanford University/professor

100/2020-11-30/yes/Ildang Paik/22/Gachon University/engineer ← add data

(9) **P8-1**

Copy the names of most recent data in the array for transmission to a remote computer.

1. Last 5 data in array(name)
2. Compute original data checksum (using bitwise ExclusiveOR)
3. Attach it to the copy.
4. Compute the checksum in the copied data
5. Compare it against the checksum in the original data.
6. Confirm that the two data are the same.

ex)	[original data]	[copied data]	[result]
checksum	55	55	Same(55=55)
name	Choi	Choi	
	Kim	Kim	
	…	…	

Step2: outline a solution

(1)setup

Fileopen

loop for file open

open one line and store the array

return array

loop for file open

open one line and store the linked list

return head

(2) Search for "Choi"

Read the data array - **P1-1**

Char name in registration data txt file

Loop for find "Choi"

Open file in reading mode

Search name "Choi"

If no file

Print can't open file

If find "Choi"

Get including data

Print all data include "Choi"

Close file

Read the data linked list head - **P1-2**

Loop for find "Choi"

Set head

If find "Choi"

Print all data including name "Choi"

Print linked list

(3) Search for all from Gachon University

Read the data array – **P2-1**

Char name in registration data txt file

Loop for find Gachon University

Open file in reading mode

Search name Gachon University

If no file

Print can't open file

If find Gachon University

Get including data

Print all data include Gachon University

Close file

Read the data linked list head – **P2-2**

Loop for find Gachon University

Set head

If find Gachon University

Print all data of including Gachon University

Print linked list

(4)Sort the data in the array in tag# order - **P3-1**

Read the data array

Loop for Sort

Sort the data

(5)Create a linked list using the sorted data - **P4-1**

Read the data array

Loop to make a linked list

Data.next = next data

Head.next = first data

Return head

(6) – **P5-1**

Read the data array

Loop for bubble sort

Sort the order by age order

Return array

Loop for write file

Print array at textfile 5-1.txt

Return head

(7) – **P6-1**

Read the data array

Loop for find "Choi"

If find {Move to next data}

Else {save at temp}

Copy temp to array

Return data_num

Print array

- **P6-2**

Read the data linked list head

Loop for find "Choi"

Set head

If find {delete node}

Else {move to next node}

Return head

Print linked list

(8) – **P7-1**

Read the data array

Add new data at the end of the array

Loop for bubble sort

Sort ascending by age

Print array

- **P7-2**

Read the data linked list

Loop for linked list

Add a new node to the end of the linked list

Loop for bubble sort

Sort ascending by tag

Print linked list

(9) – **P8-1**

Read data array

Loop for array

Find data of last 5 names in array

Copy to new file

Compute original data checksum

attach copied file

Read copied file

Loop for file read

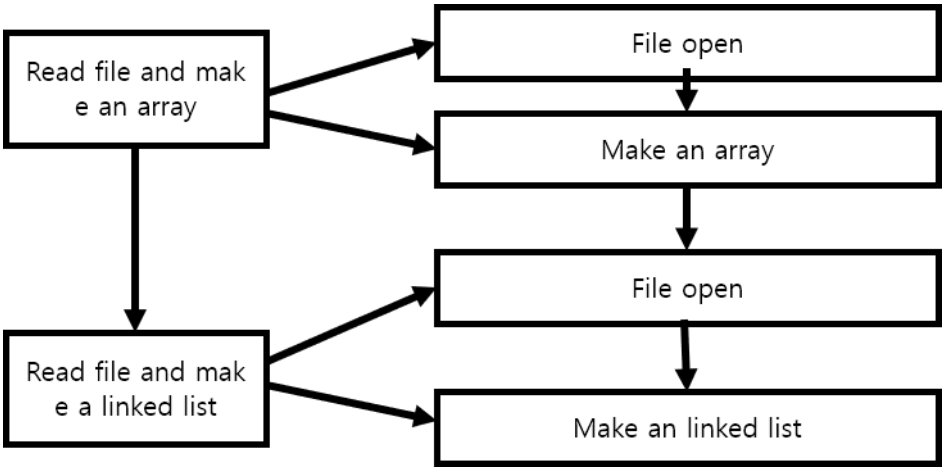
Compute the checksum in the copied data

Compare the original checksum and copied data checksum

Print compare result

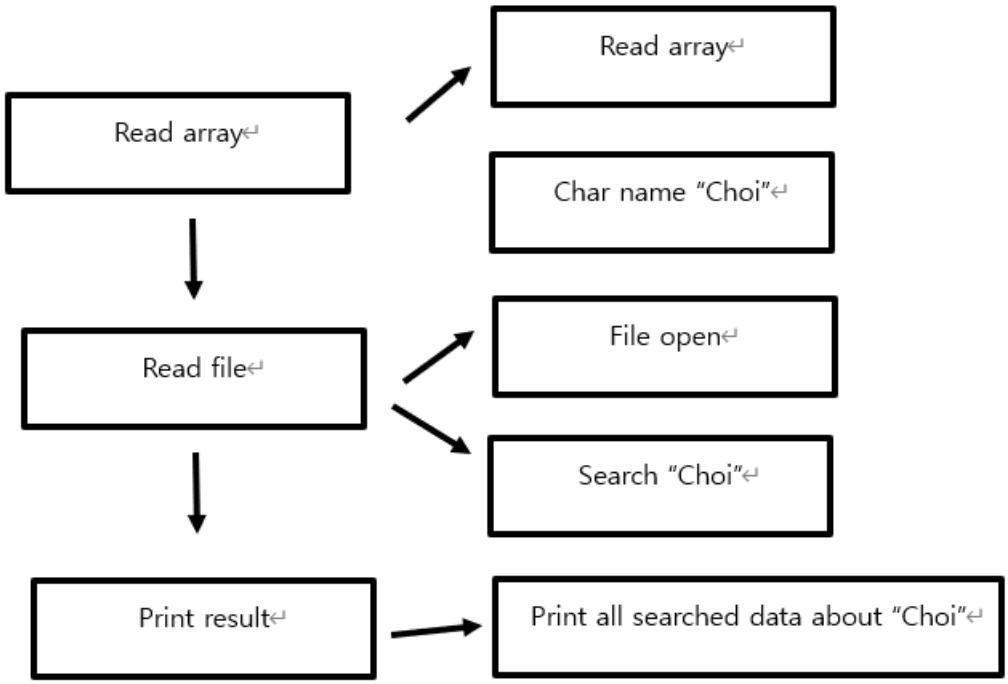
Step 3: form a program structure

(1)setup

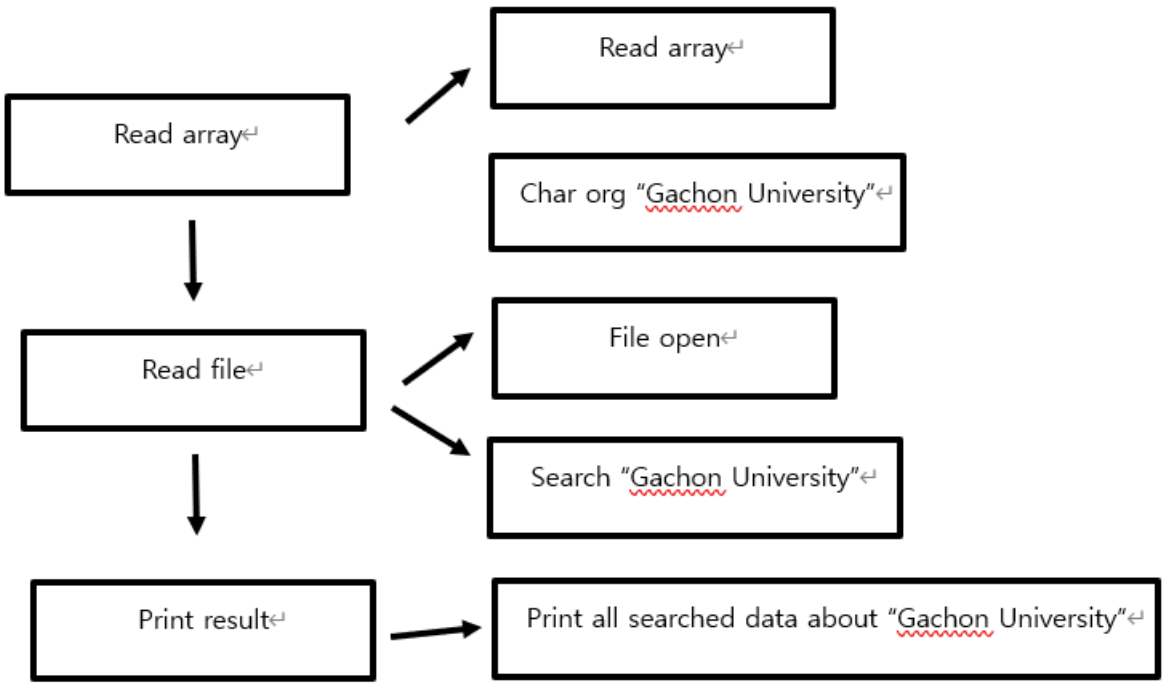


(2) Search for "Choi"

P1-1

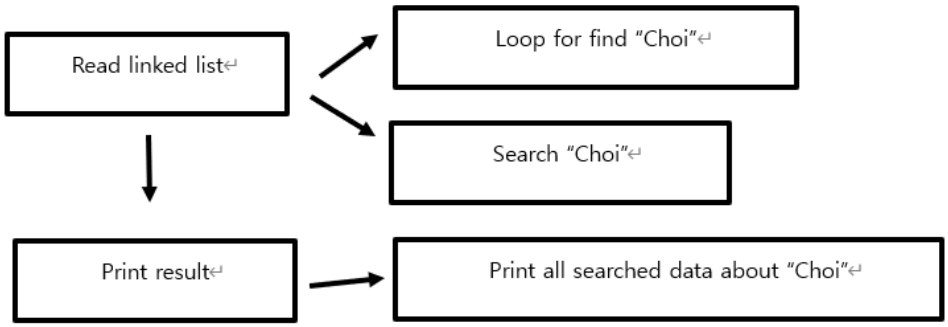

Read array
Read array
Char name "Choi"
Read file
File open
Search "Choi"
Print result
Print all searched data about "Choi"

P1-2

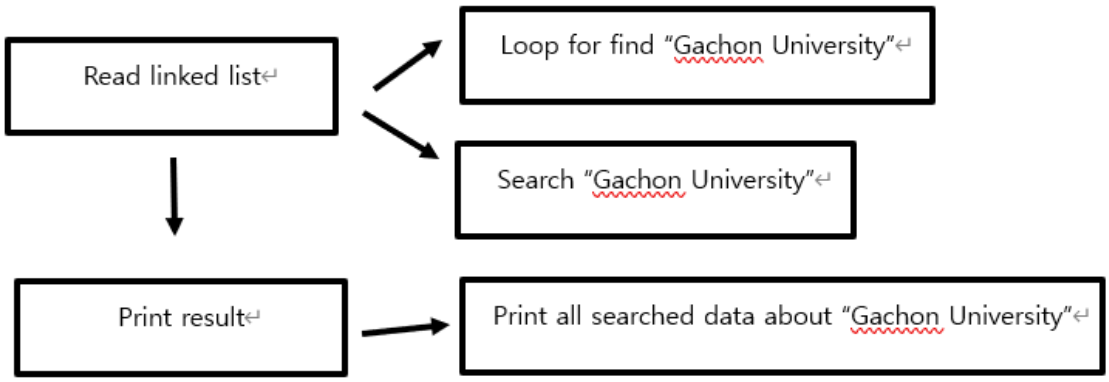

Read array
Read array
Char org "Gachon University"
Read file
File open
Search "Gachon University"
Print result
Print all searched data about "Gachon University"

(3) Search for all from Gachon University

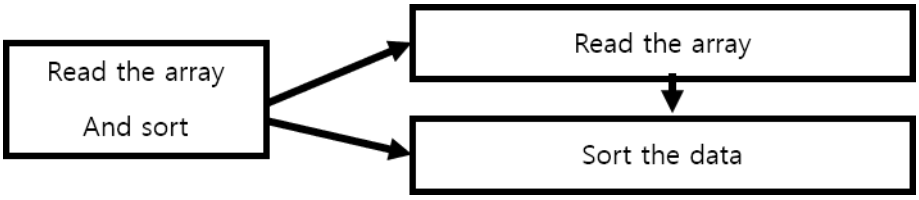
P2-1



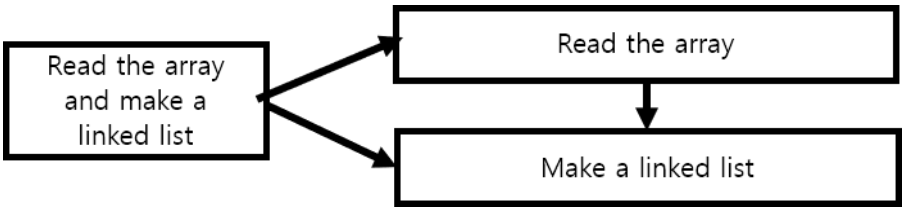
P2-2



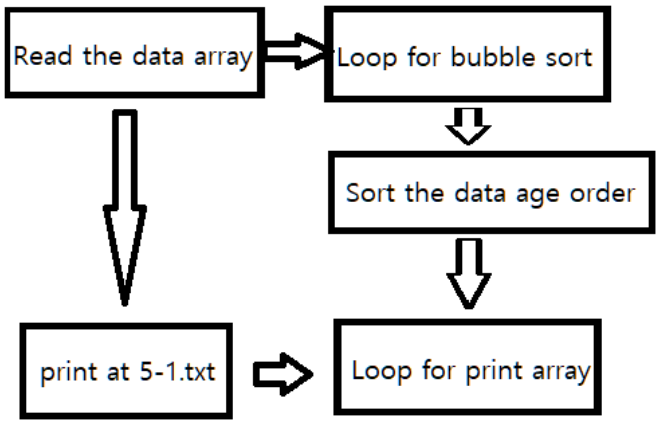
(4)Sort the data in the array in tag# order - **P3-1**



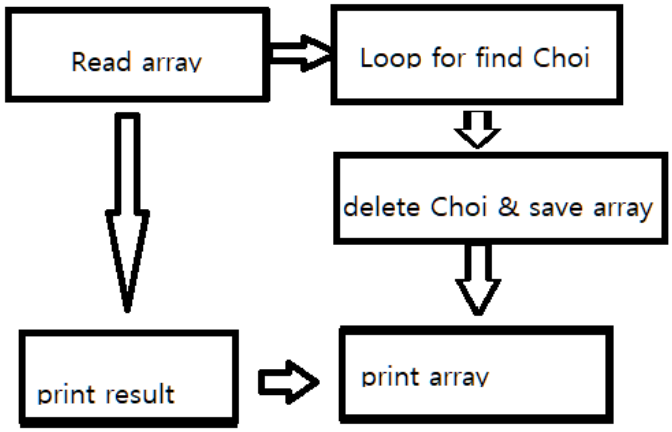
(5)Create a linked list using the sorted data - **P4-1**



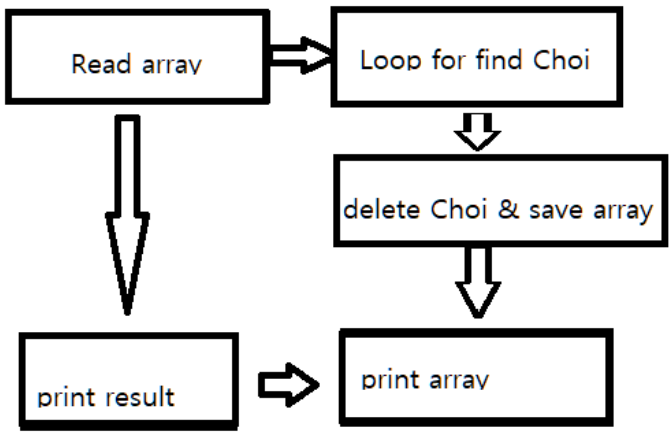
(6) – **P5-1**



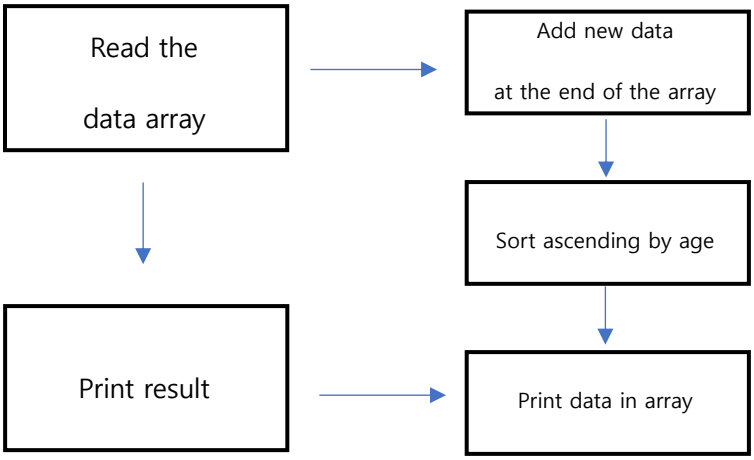
(7) – **P6-1**



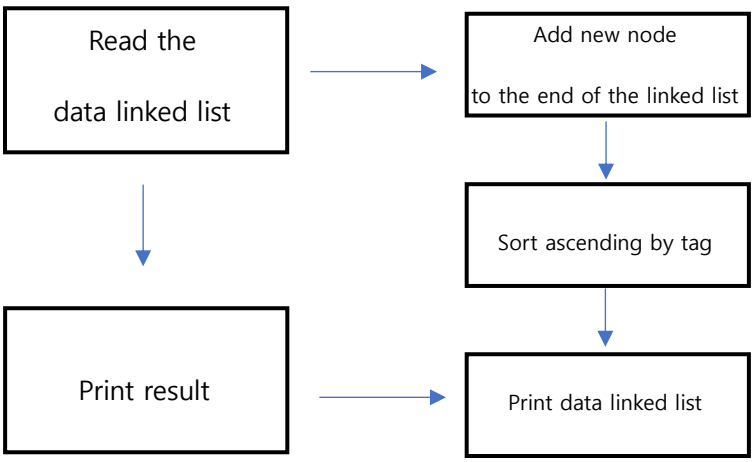
- **P6-2**



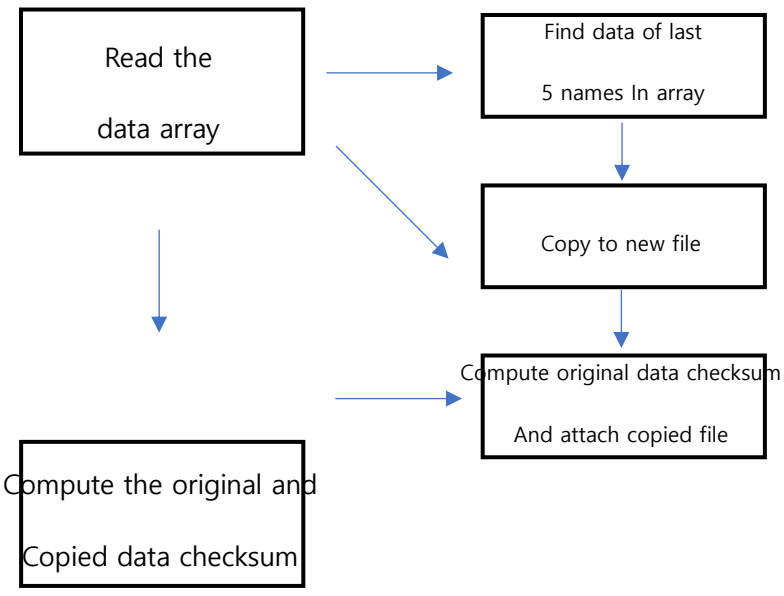
(8) – **P7-1**

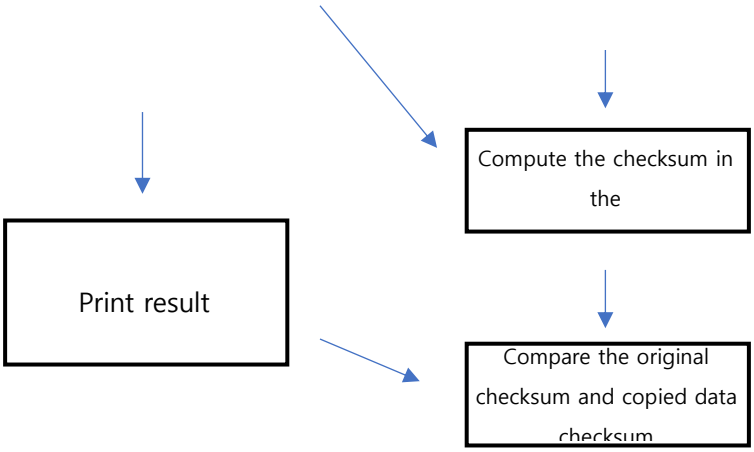

Read the
data array
Add new data
at the end of the array
Sort ascending by age
Print result
Print data in array

- **P7-2**


Read the
data linked list
Add new node
to the end of the linked list
Sort ascending by tag
Print result
Print data linked list

(9) – **P8-1**


Read the
data array
Find data of last
5 names In array
Copy to new file
Compute original data checksum
And attach copied file
Compute the original and
Copied data checksum



Step 4: write a program outline (pseudo code)

(1)setup

```
array_open(array)
{
        file = fopen("registraion_data.txt", "r");
        while (fscanf(regi_file, array))
                i++;
        data_number = i;

        fclose(regi_file);

        return data_number;
}

Linked_list_open()

{
        head;
        prev_node = head;

        regi_file = fopen("registraion_data.txt", "r");
        node;
        while (fscanf(regi_file, node)
                {
                        prev_node->next = node;
                        prev_node = node;
```

```
                    node = new_node;
                }
                prev_node->next = NULL; //last node
        }
        fclose(regi_file);

        return head;
}
```

(2) Search for "Choi"

P1-1

```
Array_findname(Data *regi_data){

FILE* myfile;

Char name="Choi";

Char array;

Char fname="registration_data,txt";

myfile = fopen(fname, "r");

        if (no file){

                printf("Can't find %s", fname);}

        for (int i=0; i<30; i++){

                fgets(buffer,300,myfile);

                if (find string name){

                        printf("All searched data in array");}

        }

        fclose(myfile);

}
```

P1-2

```
Search_Choi(Data *head){

        Data *node=head->next, *prev=head;
```

```
        While(node){
                If((node->name Choi) != NULL){
                        Printf(All searched data in linked list);}
                Prev=node;
                Node=node->next;
        }
}
```

(3) Search for all from Gachon University

P2-1

```
Array_findorg (Data *regi_data){
FILE* myfile;
Char org="Gachon University";
Char array;
Char fname="registration_data,txt";
myfile = fopen(fname, "r");
        if (no file){
                printf("Can't find %s", fname);}
        for (int i=0; i<30; i++){
                fgets(buffer,300,myfile);
                if (find string org){
                        printf("All searched data in array");}
        }
        fclose(myfile);
}
```

P2-2

```
Search_Gachon(Data *head){
        Data *node=head->next, *prev=head;
        While(node){
                If((node->org Gachon University) != NULL){
                        Printf(All searched data in linked list);}
                Prev=node;
                Node=node->next;
        }
}
```

(4)Sort the data in the array in tag# order **- P3-1**

```
sort_tag(array, sorted_array, data_number)
{
        for (int i = 0; i < data_number; i++)
                sorted_array[i] = array[i];

        for (int i = 0; i < data_number - 1; i++)
        {
                for (int j = 0; j < data_num - 1 - i; j++)
                {
                        if (sorted_array [j].tag > sorted_array [j + 1].tag)
                        {
                                temp = sorted_array [j];
                                sorted_array [j] = sorted_array [j + 1];
                                sorted_array [j + 1] = temp;
                        }
                }
        }
}
```

(5)Create a linked list using the sorted data **- P4-1**

```
Make_linked_list(sorted_array, data_number)
```

```
{
        sorted_head;
        node, prev_node;

        node = sorted_array[0];
        sorted_head ->next = node; prev_node = node;

        for (int i = 1; i < data_number; i++)
        {
                node = new_node; node = sorted_array[i];
                prev_node->next = node; prev_node = node;
        }
        node->next = NULL; //last node

        return sorted_head;
}
```

(6) – **P5-1**

```
//bring array and data_num

ageOrder(array, data_num)
{
        //sort data age order with bubble sort
        array temp;
        for loop i
        {       for loop j
                {
                        if (array[j].age > array[j + 1].age)
                        {
                                temp = array[j];
                                array[j] = array[j + 1];
                                array[j + 1] = temp;
                        }
                }
        }
        for loop
        {
                temp = array[k + 1];
                array[k] = temp;
        }
                //write data to 5-1.txt
        age_file = fopen("5-1.txt", "w");
        {
                for loop
                {
                        fprintf array at 5-1.txt
```

```
                }
}       fclose(age_file);
}
```

(7) – **P6-1**

```
int choi(Data* array, int data_num)
{
        Data temp[100];
        int i=0, j=0;

        for loop //Loop for find Choi
        {
                char* ptr = strstr(array[i].name,"Choi");

                if (ptr!= NULL)
                {
                        i++;
                        if (strstr(array[i].name, "Choi" )!= NULL) // continue Choi
                        {
                                i++;
                                temp[j] = array[i];
                                j++;
                        }
                        else                    // save at temp
                        {
                                temp[j] = array[i];
                                j++;
                        }
                }
                else                        // save at temp
                {
                        temp[j] = array[i];
                        j++;
                }
        }
        data_num = j;                // revise data_num

        for loop // copy temp to array
        {
                array[j] = temp[j];
        }

        return data_num;
}
```

- **P6-2**

```
Data* choiNode(Data* head)
{
        //set ptr & prev
        Data* ptr = head->next, * prev = head,*temp;

        while (ptr) {
```

```
                if (strstr(ptr->name, "Choi") != NULL)  // Find Choi
                {
                if (strstr((ptr->next)->name, "Choi") != NULL) // if next node is "Choi" again
                        {
                                prev->next = ((ptr->next)->next); //delete two node
                                prev = ptr;
                                ptr = ptr->next;
                        }
                        else
                        {
                                prev->next = ptr->next; /* node deleted */
                                prev = ptr;
                                ptr = ptr->next;
                        }
                }
                else
                {
                        prev = ptr;
                        ptr = ptr->next;
                }
        }

        return head;
}
```

(8) – **P7-1**

```
void add_data_to_array(Data *regi_data, int *data_num, int tag, char *date, char *fee_paid, char *name, int
age, char *organization, char *job)
{
   int new_data_num = *data_num;

   regi_data[new_data_num].tag = tag;
   strcpy(regi_data[new_data_num].date, date);
   strcpy(regi_data[new_data_num].fee_paid, fee_paid);
   strcpy(regi_data[new_data_num].name, name);
   regi_data[new_data_num].age = age;
   strcpy(regi_data[new_data_num].organization, organization);
   strcpy(regi_data[new_data_num].job, job);

   *data_num += 1;
   new_data_num = *data_num;

   Data temp;

   //   Bubble sort by age
   for (int i = 0; i < new_data_num; i++)
   {
      for (int j = 0; j < new_data_num - 1; j++)
      {
         if (regi_data[j].age > regi_data[j + 1].age)
         {
            temp = regi_data[j];
            regi_data[j] = regi_data[j + 1];
            regi_data[j + 1] = temp;
```

```
        }
      }
    }

    /* Print array */
}
```

- **P7-2**

```
void add_data_to_linked(Data *regi_head, int tag, char *date, char *fee_paid, char *name, int age, char
*organization, char *job)
{
    // Create node to be added
    Data *new_node = (Data *)malloc(sizeof(Data));
    new_node->tag = tag;
    strcpy(new_node->date, date);
    strcpy(new_node->fee_paid, fee_paid);
    strcpy(new_node->name, name);
    new_node->age = age;
    strcpy(new_node->organization, organization);
    strcpy(new_node->job, job);
    new_node->next = NULL;

    Data *ptr, *phead, *temp;

    // Insert new node into last node
    ptr = regi_head->next;
    while (ptr->next != NULL)
    {
        ptr = ptr->next;
    }
    ptr->next = new_node;

    //    Bubble sort by tag
    phead = regi_head;
    ptr = phead;
    while (ptr->next != NULL)
    {
        if (ptr->tag > ptr->next->tag)
        {
            temp = ptr->next;
            ptr->next = ptr->next->next;
            temp->next = phead;
            phead = ptr = temp;
            continue;
        }
        ptr = ptr->next;
    }

    // Print added data and sorted linked list
}
```

(9) – **P8-1**

```
void copy_is_vaild(Data *regi_data, int data_num)
```

```
{
    FILE *copyFile = fopen("8-1.txt", "w");
    char checksum;

    // Create checksum insert location on first line.
    fprintf(copyFile, "%s\n", "00");

    // 5 most recent data names in array
    for (int i = data_num - 1; i > data_num - 6; i--)
    {
        fprintf(copyFile, "%s\n", regi_data[i].name);

        int j = 0;
        while (regi_data[i].name[j] != '\0')
        {
            // using bitwise ExclusiveOR
            checksum ^= regi_data[i].name[j];
            j++;
        }
    }
    // Move the file pointer position to the start point. And Overwrite checksum value.
    fseek(copyFile, 0, SEEK_SET);
    fprintf(copyFile, "%d\n", checksum);

    fclose(copyFile);

    compare_checksum();
}

void compare_checksum()
{
    FILE *readFile = fopen("8-1.txt", "r");
    char str[25];
    char read_checksum = '\0', calculate_checksum = '\0';

    int line = 1;
    while (fgets(str, 25, readFile) != NULL)
    {
        // Since there is a checksum on the first line, it counts by excluding the first line.
        if (line != 1)
        {
            int i = 0;
            // Currently, str contains '\n', so exclude it.
            while (str[i] != '\n')
            {
                // using bitwise ExclusiveOR
                calculate_checksum ^= str[i];
                i++;
            }
        }
        else
        {
            int temp = atoi(str);
            read_checksum += temp;
        }
        line++;
```

```
    }
    fclose(readFile);

    if (read_checksum == calculate_checksum)
    {
            /* Print Same */
    }
    else
    {
            /* Print not equal */
    }
}
```

Source code :

```
/*
setup code(array_open, print_array, linked_open, print_linked) - 202035318 Kim wonjong
*/
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

//#define DATA_NUM 30

typedef struct DATA
{
        int tag;
        char date[20];
        char fee_paid[10];
        char name[25];
        int age;
        char organization[30];
        char job[15];
        struct DATA* next;
} Data;

//-----------------------------
//Function prototype
//-----------------------------

int array_open(Data*);
void print_array(Data*, int);              // setup
Data* linked_open();
void print_linked(Data*);

void array_findname(Data*); //1-1
void array_findorg(Data*);  //2-1
Data* search_Choi(Data*);   //1-2
Data* search_Gachon(Data*); //2-2

void sort_tag(Data*, Data*, int);
Data* tag_linked(Data*, int);

void ageOrder(Data*, int);
int choi(Data*, int);
Data* choiNode(Data*);

void add_data_to_array(Data*, int*, int, char*, char*, char*, int, char*, char*);
void add_data_to_linked(Data*, int, char*, char*, char*, int, char*, char*);
void copy_is_vaild(Data*, int);
void compare_checksum();

//-----------------------------
//MAIN
//-----------------------------
```

```
int main()
{
	Data regi_data[50];
	Data* regi_head;
	int data_num;

	Data tsort_data[50]; //tsort_data -> sorted data in tag order
	Data* tsort_head;		//tsort linked list's head

	data_num = array_open(regi_data);
	print_array(regi_data, data_num);

	printf("₩n");

	regi_head = linked_open();
	print_linked(regi_head->next);

	//P1-1, 202035352 안현진
	printf("₩n===P1-1===₩n");
	array_findname(regi_data);

	//P1-2, 202035352 안현진
	printf("₩n===P1-2===₩n");
	search_Choi(regi_head);

	//P2-1, 202035352 안현진
	printf("₩n===P2-1===₩n");
	array_findorg(regi_data);

	//P2-2, 202035352 안현진
	printf("₩n===P2-2===₩n");
	search_Gachon(regi_head);

	//P3-1, 202035318 Kim wonjong
	printf("₩n===P3-1===₩n");
	sort_tag(regi_data, tsort_data, data_num);
	print_array(tsort_data, data_num);

	//P4-1, 202035318 Kim wonjong
	printf("₩n===P4-1===₩n");
	tsort_head = tag_linked(tsort_data, data_num);
	print_linked(tsort_head->next);

	//P5-1, 201835546 함건욱
	ageOrder(regi_data, data_num);

	//P6-1, 201835546 함건욱
	printf("₩n===P6-1===₩n");
	data_num = choi(regi_data, data_num);
	printf("Choi canceled registration₩n");
	print_array(regi_data, data_num);

	//P6-2, 201835546 함건욱
	printf("₩n===P6-2===₩n");
```

```
	regi_head = choiNode(regi_head);
	printf("Choi cencled list₩n");
	print_linked(regi_head->next);

	// P7-1, 201834735 LeeJiheon
	add_data_to_array(regi_data, &data_num, 100, "2020-11-30", "yes", "Ildang Paik", 22,
"Gachon University", "engineer");

	// P7-2, 201834735 LeeJiheon
	add_data_to_linked(regi_head, 100, "2020-11-30", "yes", "Ildang Paik", 22, "Gachon
University", "engineer");

	// P8-1, 201834735 LeeJiheon
	copy_is_vaild(regi_data, data_num);

	return 0;
}

//-----------------------------
//Read registraion_data.txt and store the data in a struct array
//-----------------------------

int array_open(Data* regi_data)
{
	int i = 0, data_num = 0;
	FILE* regi_file;

	regi_file = fopen("registraion_data.txt", "r");
	if (regi_file == NULL)
		printf("file open error₩n");
	else
	{
		while (fscanf(regi_file,
"%d %*[/] %[^/] %*[/] %[^/] %*[/] %[^/] %*[/] %d %*[/] %[^/] %*[/] %s", &regi_data[i].tag,
regi_data[i].date, regi_data[i].fee_paid, regi_data[i].name, &regi_data[i].age,
regi_data[i].organization, regi_data[i].job) == 7)
			i++;
	}
	data_num = i;

	fclose(regi_file);

	return data_num;
}

//-----------------------------
//Print array
//-----------------------------

void print_array(Data* regi_data, int data_num)
{
	printf("===array===₩n");

	for (int i = 0; i < data_num; i++)
		printf("%d %s %s %s %d %s %s₩n", regi_data[i].tag, regi_data[i].date,
```

```
regi_data[i].fee_paid, regi_data[i].name, regi_data[i].age, regi_data[i].organization,
regi_data[i].job);
        printf("======₩n");
}

//----------------------------
//Store the data in a linked list
//----------------------------

Data* linked_open()
{
        int i = 0;
        FILE* regi_file;
        Data* node;
        Data* head;
        Data* prev_node;

        head = (Data*)malloc(sizeof(Data));
        prev_node = head;

        regi_file = fopen("registraion_data.txt", "r");
        if (regi_file == NULL)
                printf("file open error₩n");
        else
        {
                node = (Data*)malloc(sizeof(Data));
                while (fscanf(regi_file,
"%d %*[/] %[^/] %*[/] %[^/] %*[/] %[^/] %*[/] %d %*[/] %[^/] %*[/] %s", &node->tag, node->date,
node->fee_paid, node->name, &node->age, node->organization, node->job) == 7)
                {
                        prev_node->next = node;
                        prev_node = node;
                        node = (Data*)malloc(sizeof(Data));
                }
                free(node);                                                     //free
trash node
                prev_node->next = NULL; //last node
        }

        fclose(regi_file);

        return head;
}

//----------------------------
//Print linked list
//----------------------------

void print_linked(Data* ptr)
{
        printf("===linked list===₩n");
        while (ptr != NULL)
        {
                printf("%d %s %s %s %d %s %s₩n", ptr->tag, ptr->date, ptr->fee_paid, ptr-
>name, ptr->age, ptr->organization, ptr->job);
```

```
                ptr = ptr->next;
        }
        printf("======₩n");
}

//----------------------------
//P1-1 / 202035352 안현진
// Search for "Choi" in the array
// if found, print all info of the persons
//----------------------------
void array_findname(Data* regi_data) {
        FILE* myfile;
        char name[20] = "Choi";
        char buffer[300];
        char buffer_2[300];
        int line_num = 0;
        int count = 29;
        char org[35];
        char fname[30] = "registraion_data.txt";
        printf("----------------search array---------------₩n");
        printf("name to search: Choi");

        myfile = fopen(fname, "r");
        if (myfile == NULL) {
                fprintf(stderr, "Can't open the file%s ₩n", fname);
                exit(1);
        }
        printf("₩n");
        for (int i = 0; i < 30; i++) {
                fgets(buffer, 300, myfile);
                if (strstr(buffer, name))//function to find string
                {
                        printf("%d %s %s %s %d %s %s₩n", regi_data[i].tag, regi_data[i].date,
regi_data[i].fee_paid, regi_data[i].name, regi_data[i].age, regi_data[i].organization,
regi_data[i].job);
                }
        }
        fclose(myfile);

}//FINISH

//----------------------------
//P1-2 / 202035352 안현진
// Search for "Choi" in the linked list
// if found, print all info of the persons
//----------------------------
Data* search_Choi(Data* head)
{

        Data* node = head->next, * prev = head; //set node & prev

        while (node)
        {
                if (strstr(node->name, "Choi") != NULL)
                {
```

```
                        printf("%d %s %s %s %d %s %s₩n", node->tag, node->date, node->fee_paid, node->name, node->age, node->organization, node->job);

                }
                prev = node;
                node = node->next;
        }
}//finish

//-----------------------------
//P2-1 / 202035352 안현진
// Search for all from Gachon University in the array
// if found, print all info of the persons
//-----------------------------
void array_findorg(Data* regi_data) {
        FILE* myfile;
        char name[20];
        char buffer[300];
        char buffer_2[300];
        int line_num = 0;
        int count = 29;
        char org[35] = "Gachon University";
        char fname[30] = "registraion_data.txt";
        printf("----------------search array---------------₩n");
        printf("org to search: Gachon University");

        myfile = fopen(fname, "r");
        if (myfile == NULL) {
                fprintf(stderr, "Can't open the file%s ₩n", fname);
                exit(1);
        }
        printf("₩n");
        for (int i = 0; i < 30; i++) {
                fgets(buffer_2, 300, myfile);
                if (strstr(buffer_2, org))//function to find string
                {
                        printf("%d %s %s %s %d %s %s₩n", regi_data[i].tag, regi_data[i].date, regi_data[i].fee_paid, regi_data[i].name, regi_data[i].age, regi_data[i].organization, regi_data[i].job);
                }
        }
        fclose(myfile);

}//FINISH

//-----------------------------
//P2-2 / 202035352 안현진
// Search for all from Gachon University in the linked list
// if found, print all info of the persons
//-----------------------------
Data* search_Gachon(Data* head)
{

        Data* node = head->next, * prev = head; //set node & prev
```

```
        while (node)
        {
                if (strstr(node->organization, "Gachon University") != NULL)
                {
                        printf("%d %s %s %s %d %s %s₩n", node->tag, node->date, node-
>fee_paid, node->name, node->age, node->organization, node->job);

                }
                prev = node;
                node = node->next;
        }
}//finish

//-----------------------------
//P3-1 / 202035318 - Kim wonjong
//Sort the data in the array in tag# order
//-----------------------------

void sort_tag(Data* regi_data, Data* tsort_data, int data_num)
{
        Data temp;

        //tsort initialization
        for (int i = 0; i < data_num; i++)
                tsort_data[i] = regi_data[i];

        //Sort the data
        for (int i = 0; i < data_num - 1; i++)
        {
                for (int j = 0; j < data_num - 1 - i; j++)
                {
                        if (tsort_data[j].tag > tsort_data[j + 1].tag)
                        {
                                temp = tsort_data[j];
                                tsort_data[j] = tsort_data[j + 1];
                                tsort_data[j + 1] = temp;
                        }
                }
        }
}

//-----------------------------
//P4-1 / 202035318 - Kim wonjong
//Create a linked list using the sorted data
//-----------------------------

Data* tag_linked(Data* tsort_data, int data_num)
{
        Data* tsort_head;
        Data* node, * prev_node;

        //node and head allocation
        tsort_head = (Data*)malloc(sizeof(Data));
        node = (Data*)malloc(sizeof(Data));
        *node = tsort_data[0];
```

```
        tsort_head->next = node;
        prev_node = node;

        //list -> linked list
        for (int i = 1; i < data_num; i++)
        {
                node = (Data*)malloc(sizeof(Data));
                *node = tsort_data[i];
                prev_node->next = node;
                prev_node = node;
        }
        node->next = NULL; //last node

        return tsort_head;
}

//----------------------------------------
//P5-1 / 201835546 함건욱
//Sort the data in the array in age group order (using selection sort - self-study)
//** “age group”  means 10, 20, 30, ….
//Write the sorted data to a text file. (Mark P5 - 1 in code, submit textfile 5 - 1.txt)
//----------------------------------------

void ageOrder(Data* array, int data_num) //bring array and data_num
{
        Data temp;

        for (int i = 0; i < data_num; i++) //sort data age order
        {
                for (int j = 0; j < data_num - i; j++)
                {
                        if (array[j].age > array[j + 1].age)
                        {
                                temp = array[j];
                                array[j] = array[j + 1];
                                array[j + 1] = temp;
                        }
                }
        }

        for (int k = 0; k < data_num; k++)
        {
                temp = array[k + 1];
                array[k] = temp;
        }

        FILE* age_file; // FIle header

        age_file = fopen("5-1.txt", "w"); //write data to 5-1.txt
        {
                fprintf(age_file, "===age order array===₩n");
                for (int i = 0; i < data_num; i++)
                {
                        fprintf(age_file, "%d %s %s %s %d %s %s₩n", array[i].tag,
array[i].date, array[i].fee_paid, array[i].name, array[i].age, array[i].organization,
```

```
array[i].job);
                }
                fprintf(age_file, "======₩n");
        }

        fclose(age_file);
}

//----------------------------------------
//P6-1 / 201835546 함건욱
//All  “Choi” s canceled registration. Remove the data from the array
//----------------------------------------

int choi(Data* array, int data_num)
{
        Data temp[100];
        int i = 0, j = 0;

        for (i = 0; i < data_num; i++)
        {
                char* ptr = strstr(array[i].name, "Choi");

                if (ptr != NULL)
                {
                        i++;
                        if (strstr(array[i].name, "Choi") != NULL)
                        {
                                i++;
                                temp[j] = array[i];
                                j++;
                        }
                        else
                        {
                                temp[j] = array[i];
                                j++;
                        }
                }
                else
                {
                        temp[j] = array[i];
                        j++;
                }
        }
        data_num = j;

        for (j = 0; j < data_num; j++)
        {
                array[j] = temp[j];
        }

        return data_num;
}

//----------------------------------------
//P6-2 / 201835546 함건욱
```

```
//All  “Choi” s canceled registration. Remove the data from the linked list
//----------------------------------------

Data* choiNode(Data* head)
{

	Data* ptr = head->next, * prev = head, * temp; //set ptr & prev

	while (ptr)
	{
		if (strstr(ptr->name, "Choi") != NULL)
		{
			if (strstr((ptr->next)->name, "Choi") != NULL) // if nex next node is
"Choi" again
			{
				prev->next = ((ptr->next)->next); //delete two node
				prev = ptr;
				ptr = ptr->next;
			}
			else
			{
				prev->next = ptr->next; /* node deleted */
				prev = ptr;
				ptr = ptr->next;
			}
		}
		else
		{
			prev = ptr;
			ptr = ptr->next;
		}
	}

	prev = head; //sort tag order
	ptr = prev;
	while (ptr->next != NULL)
	{
		if (ptr->tag > ptr->next->tag)
		{
			temp = ptr->next;
			ptr->next = ptr->next->next;
			temp->next = prev;
			prev = ptr = temp;
			continue;
		}
		ptr = ptr->next;
	}

	return head;
}

// ----------------------------------------
// P7-1 / 201834735 LeeJiheon
// One  “Paik”  registered late. Add the data to the array(sorted order)
// ----------------------------------------
```

```
void add_data_to_array(Data* regi_data, int* data_num, int tag, char* date, char* fee_paid,
char* name, int age, char* organization, char* job)
{
	printf("₩n===P7-1===₩n");

	// Arrays start at 0, so 1 is not added
	int new_data_num = *data_num;

	regi_data[new_data_num].tag = tag;
	strcpy(regi_data[new_data_num].date, date);
	strcpy(regi_data[new_data_num].fee_paid, fee_paid);
	strcpy(regi_data[new_data_num].name, name);
	regi_data[new_data_num].age = age;
	strcpy(regi_data[new_data_num].organization, organization);
	strcpy(regi_data[new_data_num].job, job);

	*data_num += 1;
	new_data_num = *data_num;

	Data temp;

	//  Bubble sort by age
	for (int i = 0; i < new_data_num; i++)
	{
		for (int j = 0; j < new_data_num - 1; j++)
		{
			if (regi_data[j].age > regi_data[j + 1].age)
			{
				temp = regi_data[j];
				regi_data[j] = regi_data[j + 1];
				regi_data[j + 1] = temp;
			}
		}
	}

	for (int i = 0; i < new_data_num; i++)
	{
		printf("%d %s %s %s %d %s %s", regi_data[i].tag, regi_data[i].date,
regi_data[i].fee_paid, regi_data[i].name, regi_data[i].age, regi_data[i].organization,
regi_data[i].job);
		// Data mark added
		if (!strcmp(regi_data[i].name, name) && regi_data[i].tag == tag)
		{
			printf(" <===== added data");
		}
		printf("₩n");
	}
	printf("==========₩n");
}

// ----------------------------------------
// P7-2 / 201834735 LeeJiheon
// One  “Paik”  registered late. Add the data to the linked list(sorted order)
// ----------------------------------------
```

```
void add_data_to_linked(Data* regi_head, int tag, char* date, char* fee_paid, char* name, int
age, char* organization, char* job)
{
        printf("₩n===P7-2===₩n");

        // Create node to be added
        Data* new_node = (Data*)malloc(sizeof(Data));
        new_node->tag = tag;
        strcpy(new_node->date, date);
        strcpy(new_node->fee_paid, fee_paid);
        strcpy(new_node->name, name);
        new_node->age = age;
        strcpy(new_node->organization, organization);
        strcpy(new_node->job, job);
        new_node->next = NULL;

        Data* ptr, * phead, * temp;

        // Insert new node into last node
        ptr = regi_head->next;
        while (ptr->next != NULL)
        {
                ptr = ptr->next;
        }
        ptr->next = new_node;

        //  Bubble sort by tag
        phead = regi_head;
        ptr = phead;
        while (ptr->next != NULL)
        {
                if (ptr->tag > ptr->next->tag)
                {
                        temp = ptr->next;
                        ptr->next = ptr->next->next;
                        temp->next = phead;
                        phead = ptr = temp;
                        continue;
                }
                ptr = ptr->next;
        }

        // Print added data and sorted linked list
        ptr = regi_head->next;
        while (ptr != NULL)
        {
                printf("%d %s %s %s %d %s %s", ptr->tag, ptr->date, ptr->fee_paid, ptr->name,
ptr->age, ptr->organization, ptr->job);
                // Data mark added
                if (!strcmp(ptr->name, new_node->name) && ptr->tag == new_node->tag)
                {
                        printf(" <===== added data");
                }
                printf("₩n");
```

```
                ptr = ptr->next;
        }
        printf("==========₩n");
}

// ----------------------------------------
// P8-1 / 201834735 LeeJiheon
// Copy and compare
// ----------------------------------------

void copy_is_vaild(Data* regi_data, int data_num)
{
        printf("₩n===P8-1===₩n");

        // Currently, the data in the array are sorted by age in 7-1.

        FILE* copyFile = fopen("8-1.txt", "w");
        char checksum = '₩0';

        // Create checksum insert location on first line.
        fprintf(copyFile, "%s₩n", "00");

        // 5 most recent data names in array
        // Currently data_num is a number containing the last null character(= '₩0'), so -1.
        for (int i = data_num - 1; i > data_num - 6; i--)
        {
                fprintf(copyFile, "%s₩n", regi_data[i].name);

                int j = 0;
                while (regi_data[i].name[j] != '₩0')
                {
                        // using bitwise ExclusiveOR
                        checksum ^= regi_data[i].name[j];
                        j++;
                }
        }
        // Move the file pointer position to the start point. And Overwrite checksum value.
        fseek(copyFile, 0, SEEK_SET);
        fprintf(copyFile, "%d₩n", checksum);

        fclose(copyFile);

        compare_checksum();
}

void compare_checksum()
{
        FILE* readFile = fopen("8-1.txt", "r");
        char str[25];
        char read_checksum = '₩0', calculate_checksum = '₩0';

        int line = 1;
        while (fgets(str, 25, readFile) != NULL)
        {
                // Since there is a checksum on the first line, it counts by excluding the
```

```
first line.
                if (line != 1)
                {
                        int i = 0;
                        // Currently, str contains '₩n', so exclude it.
                        while (str[i] != '₩n')
                        {
                                // using bitwise ExclusiveOR
                                calculate_checksum ^= str[i];
                                i++;
                        }
                }
                else
                {
                        int temp = atoi(str);
                        read_checksum += temp;
                }
                line++;
        }
        fclose(readFile);

        printf("Checksum attached to the copy : %d₩n", read_checksum);
        printf("Checksum calculated from copied data : %d₩n₩n", calculate_checksum);

        if (read_checksum == calculate_checksum)
        {
                printf("The result of comparing %d(attached) and %d(calculated) is the
same.₩n", read_checksum, calculate_checksum);
        }
        else
        {
                printf("The result of comparing %d(attached) and %d(calculated), they are not
equal.₩n", read_checksum, calculate_checksum);
        }
        printf("==========₩n");
}
```

Result screen capture

P1-1

```
===P1-1===
----------------search array---------------
name to search: 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

```
===P1-2===
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

```
===P2-1===
----------------search array---------------
org to search: 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

```
===P2-2===
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

```
===P3-1===
===array===
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 Universitystudent 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
======
```

P4-1

```
===P4-1===
===linked list===
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 Universitystudent 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
======
```

P6-1

```
===P6-1===
Choi canceled registration
===array===
22 2020-06-29 no Tongbang Cho 29 Northwestern University marketer
29 2020-06-08 yes Bailey Houghton 31 Gachon University engineer
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
27 2020-08-24 no Konner French 42 Gachon University professor
28 2020-08-27 no Kwangsu Park 43 University of Pennsylvania student
9 2020-06-16 yes Kyungmin Kim 45 University of Sydney marketer
5 2020-06-12 yes Chunyong Park 48 University of Cambridge student
24 2020-07-24 no Stefan Wilkerson 48 University of Melbourne executive
14 2020-08-15 yes Kwangsu Cho 48 Gachon University executive
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
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
7 2020-06-28 yes Jihu Park 70 Australian National Universitystudent student
4 2020-07-03 no Jihu Cho 71 Tsinghua University engineer
23 2020-06-15 yes Seungmin Cho 71 Stanford University professor
17 2020-08-14 no Chunyong Chang 75 Gachon University student
======
```

P6-2

```
===P6-2===
Choi cencled list
===linked list===
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 Universitystudent 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
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
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
======
```

P7-1

```
===P7-1===
100 2020-11-30 yes Ildang Paik 22 Gachon University engineer <===== added data
22 2020-06-29 no Tongbang Cho 29 Northwestern University marketer
29 2020-06-08 yes Bailey Houghton 31 Gachon University engineer
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
27 2020-08-24 no Konner French 42 Gachon University professor
28 2020-08-27 no Kwangsu Park 43 University of Pennsylvania student
9 2020-06-16 yes Kyungmin Kim 45 University of Sydney marketer
5 2020-06-12 yes Chunyong Park 48 University of Cambridge student
24 2020-07-24 no Stefan Wilkerson 48 University of Melbourne executive
14 2020-08-15 yes Kwangsu Cho 48 Gachon University executive
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
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
7 2020-06-28 yes Jihu Park 70 Australian National Universitystudent student
4 2020-07-03 no Jihu Cho 71 Tsinghua University engineer
23 2020-06-15 yes Seungmin Cho 71 Stanford University professor
17 2020-08-14 no Chunyong Chang 75 Gachon University student
==========
```

P7-2

```
===P7-2===
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 Universitystudent 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
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
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
100 2020-11-30 yes Ildang Paik 22 Gachon University engineer <===== added data
==========
```

P8-1

```
===P8-1===
Checksum attached to the copy : 43
Checksum calculated from copied data : 43

The result of comparing 43(attached) and 43(calculated) is the same.
==========
```

Contribution percentage

202035318 – Kim wonjong (25%)

Setup, P3-1, P4-1 problem solving steps, source code

202035352 – Ahn hyunjin (25%)

P1-1, P1-2, P2-1, P2-2 problem solving steps, source code

201834735 – Lee jiheon (25%)

P7-1, P7-2, P8-1 problem solving steps, source code

201835546 – Ham geonwook (25%)

P5-1, P6-1, P6-2 problem solving steps, source code