Team-2

//Team introduction (list all members & roles)

Members	Roles
202135842 Cho Taewan	Setup, p7-2, p8-1, p8-2
202135833 Jeong Yeongjun	P1-1, P1-2, P2-1
202135730 Kim Byungkyu	P2-2, P4, P6-1, write report
202135814 Lee Jeonggyun	P5, P7-1, write report

// Contribution percentage (Kim: 25%, Park: 30% ...)

Cho Taewan : 50%

Jeong Yeongjun : 20%

Kim Byungkyu : 20%

Lee Jeonggyun : 10%

// Achievement table (self-evaluation)

Cho Taewan

Problem	Menbers Involved	Achievement
Setup	Cho Taewan	100
P1-1	Cho Taewan, Jeong Yeongjun	100
P1-2	Cho Taewan, Jeong Yeongjun	100
P2-1	Cho Taewan, Jeong Yeongjun	100
P2-2	Cho Taewan, Kim Byungkyu	100
P3	Cho Taewan, Jeong Yeongjun	100
P4	Cho Taewan, Kim Byungkyu, Jeong Yeongjun	100
P5	Cho Taewan, Lee Jeonggyun, Jeong Yeongjun	100
P6-1	Cho Taewan, Kim Byungkyu	100
P6-2	Cho Taewan, Kim Byungkyu	100
P7-1	Cho Taewan, Lee Jeonggyun, Kim Byungkyu	100
P7-2	Cho Taewan, Lee Jeonggyun, Jeong Yeongjun	100
P8-1	Cho Taewan, Lee Jeonggyun	100
P8-2	Cho Taewan	100

Jeong Yeongjun

Problem	Menbers Involved	Achievement
Setup	Cho Taewan	0
P1-1	Cho Taewan, Jeong Yeongjun	50
P1-2	Cho Taewan, Jeong Yeongjun	50
P2-1	Cho Taewan, Jeong Yeongjun	50
P2-2	Cho Taewan, Kim Byungkyu	0
P3	Cho Taewan, Jeong Yeongjun	70
P4	Cho Taewan, Kim Byungkyu, Jeong Yeongjun	70
P5	Cho Taewan, Lee Jeonggyun, Jeong Yeongjun	50
P6-1	Cho Taewan, Kim Byungkyu	0
P6-2	Cho Taewan, Kim Byungkyu	0
P7-1	Cho Taewan, Lee Jeonggyun, Kim Byungkyu	0
P7-2	Cho Taewan, Lee Jeonggyun, Jeong Yeongjun	30
P8-1	Cho Taewan, Lee Jeonggyun	0
P8-2	Cho Taewan	0

Kim Byungkyu

Problem	Menbers Involved	Achievement
Setup	Cho Taewan	0
P1-1	Cho Taewan, Jeong Yeongjun	0
P1-2	Cho Taewan, Jeong Yeongjun	0
P2-1	Cho Taewan, Jeong Yeongjun	0
P2-2	Cho Taewan, Kim Byungkyu	80
P3	Cho Taewan, Jeong Yeongjun	0
P4	Cho Taewan, Kim Byungkyu, Jeong Yeongjun	50
P5	Cho Taewan, Lee Jeonggyun, Jeong Yeongjun	0
P6-1	Cho Taewan, Kim Byungkyu	100
P6-2	Cho Taewan, Kim Byungkyu	90
P7-1	Cho Taewan, Lee Jeonggyun, Kim Byungkyu	90
P7-2	Cho Taewan, Lee Jeonggyun, Jeong Yeongjun	0
P8-1	Cho Taewan, Lee Jeonggyun	0
P8-2	Cho Taewan	0

Lee Jeonggyun

Problem	Menbers Involved	Achievement
Setup	Cho Taewan	0
P1-1	Cho Taewan, Jeong Yeongjun	0
P1-2	Cho Taewan, Jeong Yeongjun	0
P2-1	Cho Taewan, Jeong Yeongjun	0
P2-2	Cho Taewan, Kim Byungkyu	0
P3	Cho Taewan, Jeong Yeongjun	0
P4	Cho Taewan, Kim Byungkyu, Jeong Yeongjun	0
P5	Cho Taewan, Lee Jeonggyun, Jeong Yeongjun	10
P6-1	Cho Taewan, Kim Byungkyu	0
P6-2	Cho Taewan, Kim Byungkyu	0
P7-1	Cho Taewan, Lee Jeonggyun, Kim Byungkyu	10
P7-2	Cho Taewan, Lee Jeonggyun, Jeong Yeongjun	10
P8-1	Cho Taewan, Lee Jeonggyun	10
P8-2	Cho Taewan	0

// Problem solving steps 1-4

setup

step1.

-Read file and store in struct array and Linked-List

-Draw the problem & create examples

txt. file

```
2/Gildong
1/Taewan
```

->

struct array

number	name
2	Gildong
1	Taewan

and

Linked-list

number	name	next
2	Gildong	1004

1004

number	name	next
2	Gildong	NULL

Identify cases to consider

- array overload
- multiple data
- end of Linked-list must be NULL
- divide using '/'

step2.

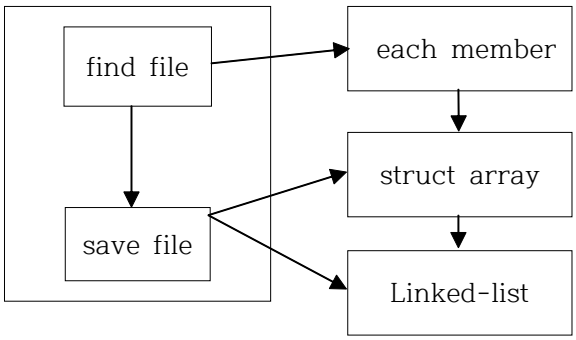
-outline a Solution

- find file
- make proper number of array
- divide into each member
- save each array in Linked-list
- NULL end of the Linked-list

step3.

-key codes : fopen, fgets, strtok, createNewNode

-flow graph



step4.

Form a Program Structure

declartions

main

fopen file

while loop(file != NULL)

while loop(token != NULL)

find&save data in array;

copy array in Linked-list;

Pseudo Code

```
fopen file("registration.txt", "r")
while(fgets(line, sizeof(line), myInFile) != NULL) {
        char* token = strtok(line, "/")
        while(token != NULL) {
                /* find members and save */
        }
/* copy array in Linked-list */
}
```

P1-1

step1.

-search the 'Choi' in struct array and print all information

-Draw the problem & create examples.

input output

number	name
3	'Choi'
1	'Kim'
2	'Choi'

-> 3/Choi
2/Choi

Identify cases to consider

The words include 'Choi' (ex : 'Chois')

No 'Choi'

step2.

-outline Solution

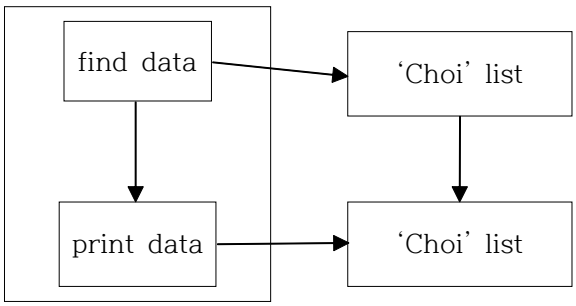
search array list whose name member is 'Choi'

print ‘Choi’’s all information

step3.

-key codes : strstr(compare)

-flow graph



step4.

-Form a Program Structure

declarations

main

char sub[5] = ‘Choi’;

for (0 through count of array)

if(including ‘Choi’)

print struct;

Write a Program Outline: Pseudo Code

```
char sub[5] = ‘Choi’;
struct ARRAY* ptr = temp1;
for (i=0; i<count; i++) {
        if strstr(ptr[i].name, sub) != NULL
                /* print struct array */
}
```

p1-2

step1.

-search the ‘Choi’ in linked-list and print all inforamtion

-Draw the problem & create examples.

Linked-list 1004 2008

number	name	next
3	Choi	1004

number	name	next
2	Gildong	2008

number	name	next
1	Choi	NULL

->

3/Choi 1/Choi

Identify cases to consider

The words include ‘Choi’ (ex : ‘Chois’)

No ‘Choi’

search in linked-list

step2.

-outline Solution

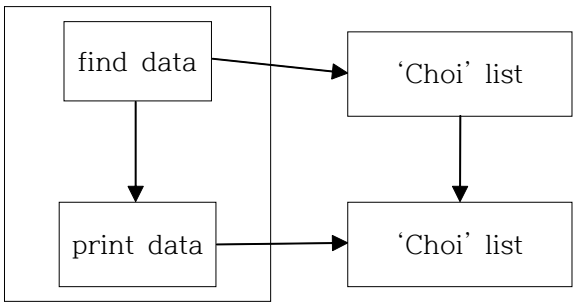
search linked-list whose name member is 'Choi'

print 'Choi''s all information

step3.

-key codes : strstr(compare)

-flow graph



step4.

-Form a Program Structure

declarations

main

```
char sub[5] = 'Choi';
while (linked-list->next != NULL) {
        if(including 'Choi')
                print struct;
}
```

Write a Program Outline: Pseudo Code

```
char sub[5] = 'Choi';
struct ARRAY* ptr = temp2;
while (ptr->next != NULL) {
        if strstr(ptr[i].name, sub) != NULL
                /* print struct array */
        /* to next linked-list */
}
```

p2-1

step1.

-Search for all from Gachon University in struct array

-Draw the problem & create examples

number	organization
3	Gachon University
1	Gachon University
2	Seoul University

-> 3/Gachon University
1/Gachon University

Identify cases to consider

The words include ‘Gachon University’ (ex : ‘Chois’)

No ‘Gachon University’

step2.

-outline Solution

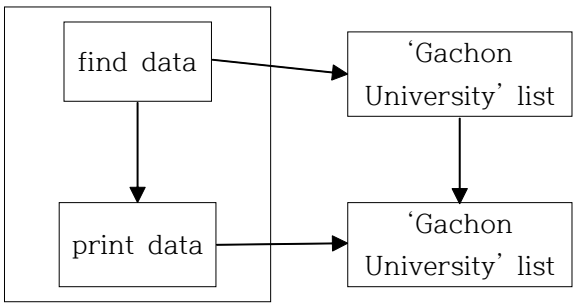
search array whose organization member is ‘Gachon University’

print ‘Gachon University’‘s all information

step3.

-key codes : strstr(compare)

-flow graph



step4.

-Form a Program Structure

declarations

main

```
char sub[18] = ‘Gachon University’;
for (0 through count of array)
        if(including ‘Gachon University’)
                print struct;
```

Write a Program Outline: Pseudo Code

```
char sub[18] = ‘Gachon University’;
struct ARRAY* ptr = temp1;
for (i=0; i<count; i++) {
        if strstr(ptr[i].name, sub) != NULL
                /* print struct array */
}
```

p2-2

step1.

-search the ‘Gachon University’ in linked-list and print all inforamtion

-Draw the problem & create examples.

Linked-list

number	organization	next
3	Gachon University	1004

1004

number	organization	next
2	Seoul University	2008

2008

number	organization	next
1	Gachon University	NULL

3/Gachon University
1/Gachon University
->

Identify cases to consider

The words include ‘Gachon University’

No ‘Gachon University’

search in linked-list

step2.

-outline Solution

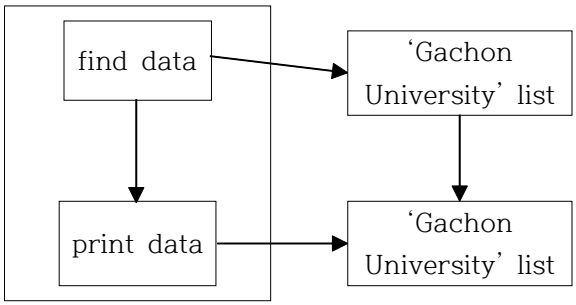
search linked-list whose name member is ‘Choi’

print ‘Choi’‘s all information

step3.

-key codes : strstr(compare)

-flow graph



step4.

-Form a Program Structure

declarations

main

```
char sub[18] = ‘Gachon University’;
while (linked-list->next != NULL) {
        if(including ‘Gachon University’)
                print struct;
}
```

Write a Program Outline: Pseudo Code

```
char sub[18] = ‘Gachon University’;
struct ARRAY* ptr = temp2;
while (ptr->next != NULL) {
        if strstr(ptr[i].name, sub) != NULL
                /* print struct array */
        /* to next linked-list */
}
```

p3

step1.

-Sort the dta in the array in tag# order

-Draw the problem & create examples

tag	name
3	'Choi'
1	'Kim'
2	'Choi'

->

tag	name
1	'Kim'
2	'Choi'
3	'Choi'

Identify cases to consider

step2

-outline Solution

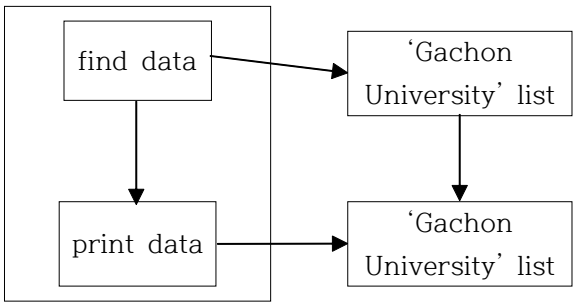
compare the tag member

if we need change the list number

step3.

-key codes : swap(function : change the number of array list)

-flow graph



step4.

-Form a Program Structure

declarations

main

```
for (i=0 through (count of array - 1))
        for ((count of array - 1) through I)
                find bigger one and change if we need;
```

Write a Program Outline: Pseudo Code

```
for (i=0; i<n-1; i++) {
        for (j=n-1; j>i; j--) {
                if head[j-1].tag > head[j].tag {
                        /* swap the list number */
                }
        }
}
```

p4

step1

-Create a linked list using the sorted data

-Draw the problem & create examples

array

tag	name
1	'Kim'
2	'Choi'
3	'Choi'

linked-list 1004 2008

->

tag	name	next
1	'Kim'	1004

tag	name	next
2	'Choi'	2008

tag	name	next
3	'Choi'	NULL

Identify cases to consider

make head then linkNode

step2

-outline Solution

Loop through the sorted array

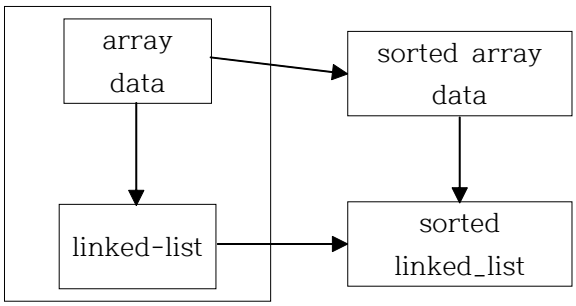
createNewNode

linkNode

step3.

-key codes : createNewNode, linkNode

-flow graph



step4.

-Form a Program Structure

declarations

main

for loop (0 through count of array)

copy array in linked-list;

linkNode;

Write a Program Outline: Pseudo Code

```
for (count=0; count<num; count++) {
        if (count == 0) {
```

```
                /* create head */
        }
        else {
                /* create Node*/
                /* linkNode */
        }
}
```

p5

step1.

-Write the sorted data to a text file

-Draw the problem and create examples

linked-list

tag	name	next
1	'Kim'	1004

tag	name	next
2	'Choi'	2008

tag	name	next
3	'Choi'	NULL

txt file

-> 1/'Kim'
2/'Choi'
3/'Choi'

step2

-outline Solution

open file 'w'

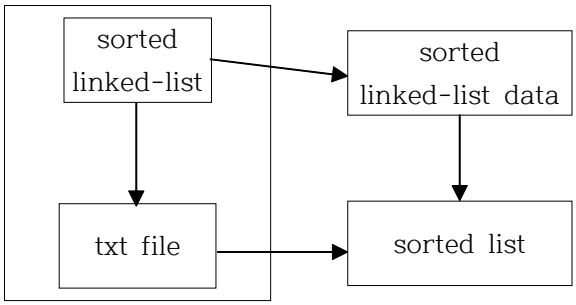
loop through the sorted linked-list

write them to txt file

step3.

-key codes : fopen(name, "w"), fprintf

-flow graph



step4.

-Form a Program Structure
declarations

main
	open file(“w”)
	for loop (0 through count of sorted_linked_list)
		print on txt file;

Write a Program Outline: Pseudo Code
myOutFile = fopen(name,“w”);
struct NODE ptr = sorted_temp;
while (ptr != NULL) {
	/* print on txt file */
	ptr = ptr->next;
}

p6-1

step1.
-All “Choi” s canceled registration in struct array
-Draw the problem and create examples

tag	name
1	‘Kim’
2	‘Lee’
3	‘Choi’

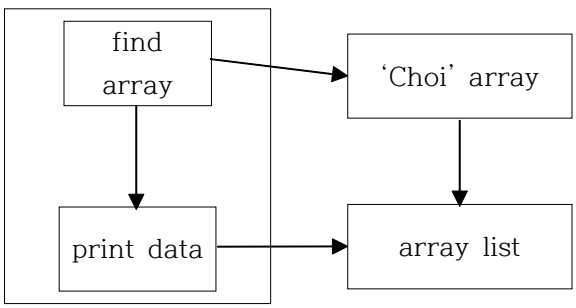
->

tag	name
1	‘Kim’
2	‘Lee’

1/‘Kim’
2/‘Lee’

step2
-outline Solution
	loop through array
		if name == ‘Choi’ delete the struct
	loop through after array
		print left struct array

step3.
-key codes : strstr
-flow graph



step4.
-Form a Program Structure

declarations

main
	for loop (0 through count of array)
		if (name == ‘Choi’)
			delete;
	for loop (0 through count of after array)
		print struct array

Write a Program Outline: Pseudo Code
```
for (i=0; i<count; i++) {
	if (strstr(ptr[i].name, del) {
		for (j = i; j<count; j++) {
			/* delete the struct */
		}
		count--;
	}
for (j=0; j<count; j++) {
	/* print struct array */
}
```

p6-2

step1.

-All “Choi” s canceled registration in linked-list

-Draw the problem and create examples

1004

tag	name	next
1	‘Kim’	1004

2008

tag	name	next
2	‘Lee’	2008

tag	name	next
3	‘Choi’	NULL

->

tag	name	next
1	‘Kim’	1004

tag	name	next
2	‘Lee’	NULL

1/‘Kim’
2/‘Lee’

step2

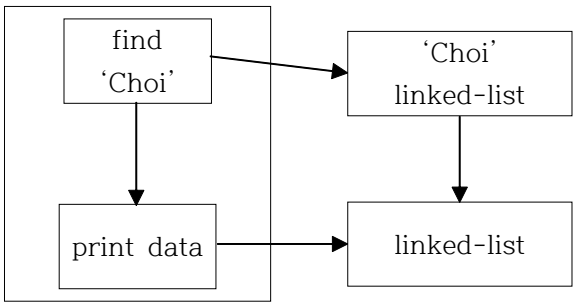
-outline Solution

	loop through array
		if name == ‘Choi’ delete the struct
	loop through after array
		print left struct array

step3.

-key codes : strstr

-flow graph



step4.

-Form a Program Structure

declarations

main

```
        while loop (0 through NULL)
                if (name == 'Choi')
                        delete;
        while loop (0 through NULL)
                print struct array
```

Write a Program Outline: Pseudo Code

```
while (ptr != NULL)
        if (strstr(ptr->next, del) != NULL) {
                count++;
        {
                if (ptr->next == NULL) {
                        /* NULL in the end */
                }
                else if (t == 0) {
                        /* if find "Choi" in first node delete it */
                }
                else {
                        /* if find "Choi" delete it */
                }
        }
        else {
                /* minus to tag numbeeer */
        }
        for (i=0; i<cnt_first; i++) {
                /* if find "Choi" in first node connect left thing */
        }
        while (head != NULL) {
                /* print_linked_list */
        }
}
```

p7-1

step1.

-Give Paik's tag# in consideration of the tag# of the registered persons.

(“tag#/2021-11-30/yes/Gildong Paik/35/Gachon University/Student”)
-Draw the problem and create examples

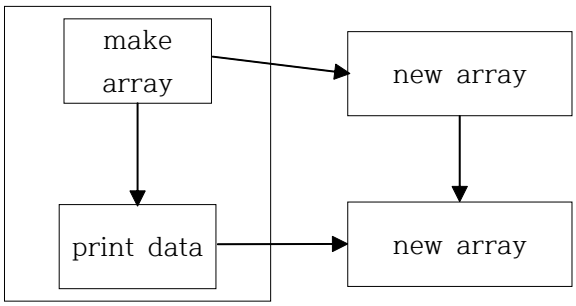
tag	name
1	‘Kim’
2	‘Lee’
3	‘Choi’

->

tag	name
1	‘Kim’
2	‘Lee’
3	‘Choi’
4	‘Paik’

step2
-outline Solution
create New array with “Paik”
for loop 0 through array
print struct array

step3.
-key codes : num + 1
-flow graph



step4.
main
create New array with Paik
for loop 0 through count of array with Paik
print struct array

Write a Program Outline: Pseudo Code

```
/* create new struct array(tag = num+1)
for (j=0; j<num+1; j++) {
        /* print struct array */
}
```

p7-2

step1.
-add data in sorted linked-list at last
(“tag#/2021-11-30/yes/Gildong Paik/35/Gachon University/Student”)
-Draw the problem and create examples

tag	name	next
1	‘Kim’	1004

tag	name	next
2	‘Lee’	2008

tag	name	next
3	‘Choi’	NULL

->

1004

2008

tag	name	next
1	‘Kim’	1004

tag	name	next
2	‘Lee’	2008

tag	name	next
3	‘Choi’	3012

3012

tag	name	next
4	‘Paik’	NULL

step2

-outline Solution

create New Node ‘Paik’

while 1

link New Node;

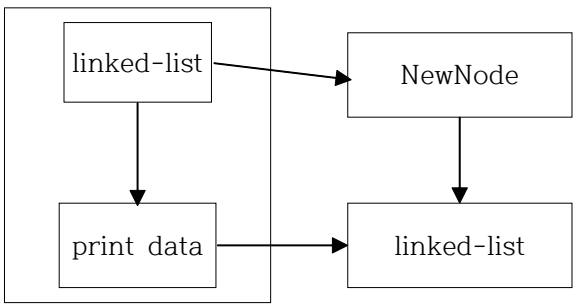
while loop until NULL

print linked list;

step3.

-key codes : num + 1

-flow graph



step4.

main

create New struct ‘Paik’

while 1

createNewNode (tag = num+1);

linkNode;

while loop until Node == NULL

print_linked_list;

Write a Program Outline: Pseudo Code

```
/* create new struct */
while (1) {
        /* createNewNode which tag is num+1 */
        /* linkNode in the end of Node*/
}
```

```
        while (temp2 != NULL) {
                /* print linked list */
                temp2 = temp2 -> next;
        }
num++;
```

p8-1

step1.

-Copy the most recent data in the array then make “p8-1.txt” and checksum.

step2.

1. Make p8-1.txt that is stored in struct array.
2. Copy the p8-2.txt that is copyed of p8-1.txt file.
3. Compare p8-1.txt's checksum and p8-2.txt's checksum.
4. If that checksum vaule is same, copy was successful.

step3.

- compute each word's ascii code as xor.
- And compare the each chcksum.

step4.

- process of make a checksum

```
for (int i = first word ; i <= last word; i++) {
        for (int j = fisrt alpabet; j < word's len ); j++) {
            if (j == 0) // if first alpabet
                name = each alpabet;
            else
                name += each alpabet;
        }
        if (i == cnt - 9) // if first word
            check_sum = name;
        else
            check_sum ^= name;
    }
```

// Codes & Result screenshots for each problem (step 5)

//P1-1

```
void p1_1(struct ARRAY* temp1, int num) {
    /*  search the 'Choi' in struct array and print all information  */
    printf("\nP1-1\n[search the 'Choi' in struct array and print all information]\n\n");
    struct ARRAY* ptr = temp1;
    char sub[5] = "Choi";
    for (int i = 0; i <num; i++)
        if (strstr(ptr[i].name, sub) != NULL)
            print_struct_array(ptr, i);
}
```

```
P1-1
[search the 'Choi' in struct array and print all information]

11 2021-07-22 no Kwangsu Choi 48 Seoul National University marketer
15 2021-07-12 no Tongbang Choi 26 Cornell University engineer
1 2021-08-25 yes Jihu Choi 74 Harvard University engineer
30 2021-07-13 yes Kyungmin Choi 44 Duke University student
2 2021-08-22 no Seungmin Choi 31 Gachon University staff
```

//P1-2

```
void p1_2(struct NODE* temp2) {
    /*  search the 'Choi' in linked-list and print all information  */
    printf("\nP1-2\n[search the 'Choi' in linked-list and print all information]\n\n");
    struct NODE* ptr = temp2;
    char sub[5] = "Choi";
    while (ptr->next != NULL) {
        if (strstr(ptr->name, sub) != NULL)
            print_linked_list(ptr);
        ptr = ptr->next;
    }
}
```

```
P1-2
[search the 'Choi' in linked-list and print all information]

11 2021-07-22 no Kwangsu Choi 48 Seoul National University marketer
15 2021-07-12 no Tongbang Choi 26 Cornell University engineer
1 2021-08-25 yes Jihu Choi 74 Harvard University engineer
30 2021-07-13 yes Kyungmin Choi 44 Duke University student
2 2021-08-22 no Seungmin Choi 31 Gachon University staff
```

//P2-1

```
void p2_2(struct NODE* temp2) {
    /* Search for all from Gachon University in linked-list */
    printf("\nP2-2\n[Search for all from Gachon University in linked-list]\n\n");
    struct NODE* ptr = temp2;
    char sub[18] = "Gachon University";
    while (ptr != NULL) {
        if (strstr(ptr->organization, sub) != NULL)
            print_linked_list(ptr);
        ptr = ptr->next;
    }
}
```

```
P1-2
[search the 'Choi' in linked-list and print all information]

11 2021-07-22 no Kwangsu Choi 48 Seoul National University marketer
15 2021-07-12 no Tongbang Choi 26 Cornell University engineer
1 2021-08-25 yes Jihu Choi 74 Harvard University engineer
30 2021-07-13 yes Kyungmin Choi 44 Duke University student
2 2021-08-22 no Seungmin Choi 31 Gachon University staff
```

//P2-2

```
void p2_2(struct NODE* temp2) {
    /* Search for all from Gachon University in linked-list */
    printf("\nP2-2\n[Search for all from Gachon University in linked-list]\n\n");
    struct NODE* ptr = temp2;
    char sub[18] = "Gachon University";
    while (ptr != NULL) {
        if (strstr(ptr->organization, sub) != NULL)
            print_linked_list(ptr);
        ptr = ptr->next;
    }
}
```

```
P2-2
[Search for all from Gachon University in linked-list]

29 2021-06-08 yes Bailey Houghton 31 Gachon University engineer
12 2021-07-22 no Owen Martin 66 Gachon University engineer
8 2021-06-04 no Moises Barlow 57 Gachon University engineer
14 2021-08-15 yes Kwangsu Cho 48 Gachon University executive
27 2021-08-24 no Konner French 42 Gachon University professor
17 2021-08-14 no Chunyong Chang 75 Gachon University student
2 2021-08-22 no Seungmin Choi 31 Gachon University staff
13 2021-06-03 yes Chinho Cho 68 Gachon University student
20 2021-07-30 yes Chinho Kim 52 Gachon University engineer
```

//P3

```
void p3(struct ARRAY* head, int n) {
    /*  Sort the data in the array in tag# order  */
    printf("\nP3\nSort the data in the array in tag# order\n\n");
    int i, j;
    for (i = 0; i <n - 1; i++) {
        for (j = n - 1; j >i; j--)
            if (head[j - 1].tag >head[j].tag)
                swap(&head[j - 1], &head[j]);
    }
    printf("Data in struct array is sorted\n");
    for (int j = 0; j <n; j++)
        print_struct_array(head, j);
}
```

```
P3
Sort the data in the array in tag# order

Data in struct array is sorted
 1 2021-08-25 yes Jihu Choi 74 Harvard University engineer
 2 2021-08-22 no Seungmin Choi 31 Gachon University staff
 3 2021-07-01 no Chinho Park 53 Peking University engineer
 4 2021-07-03 no Jihu Cho 71 Tsinghua University engineer
 5 2021-06-12 yes Chunyong Park 48 University of Cambridge student
 6 2021-06-04 yes Bobby Anderson 33 McGill University engineer
 7 2021-06-28 yes Jihu Park 70 Australian National University student
 8 2021-06-04 no Moises Barlow 57 Gachon University engineer
 9 2021-06-16 yes Kyungmin Kim 45 University of Sydney marketer
10 2021-06-06 yes William Cohen 37 University of Cambridge engineer
11 2021-07-22 no Kwangsu Choi 48 Seoul National University marketer
12 2021-07-22 no Owen Martin 66 Gachon University engineer
13 2021-06-03 yes Chinho Cho 68 Gachon University student
14 2021-08-15 yes Kwangsu Cho 48 Gachon University executive
15 2021-07-12 no Tongbang Choi 26 Cornell University engineer
16 2021-08-16 yes Tongbang Kim 39 Tsinghua University student
17 2021-08-14 no Chunyong Chang 75 Gachon University student
18 2021-06-14 no Tongbang Park 32 New York University engineer
19 2021-06-07 yes Chunyong Kim 34 Harvard University staff
20 2021-07-30 yes Chinho Kim 52 Gachon University engineer
21 2021-07-21 yes Jude Smith 38 Cornell University executive
22 2021-06-29 no Tongbang Cho 29 Northwestern University marketer
23 2021-06-15 yes Seungmin Cho 71 Stanford University professor
24 2021-07-24 no Stefan Wilkerson 48 University of Melbourne executive
25 2021-06-09 no Archie Hunt 60 Fudan University student
26 2021-06-30 yes Sincere Bradley 58 University of Hong Kong staff
27 2021-08-24 no Konner French 42 Gachon University professor
28 2021-08-27 no Kwangsu Park 43 University of Pennsylvania student
29 2021-06-08 yes Bailey Houghton 31 Gachon University engineer
30 2021-07-13 yes Kyungmin Choi 44 Duke University student
```

//P4

```
void p4(struct ARRAY* temp1, struct NODE* sorted_temp, int num) {
    /*  Create a linked list using the sorted data  */
    printf("\nP4\n[Create a linked list using the sorted data]\n\n");
    struct NODE* head = NULL;
    struct NODE* tmp;
    for (int count = 0; count <num; count++) {
        if (count == 0)
            head = createNewNode(temp1[count].tag, temp1[count].date, temp1[count].fee_paid,
temp1[count].name, temp1[count].age, temp1[count].organization, temp1[count].job);
        else {
            tmp = createNewNode(temp1[count].tag, temp1[count].date, temp1[count].fee_paid,
temp1[count].name, temp1[count].age, temp1[count].organization, temp1[count].job);
            linkNode(head, tmp);
        }
    }
    *sorted_temp = *head;
    while (head != NULL) {
        print_linked_list(head);
        head = head->next;
    }
}
```

```
P4
[Create a linked list using the sorted data]

1 2021-08-25 yes Jihu Choi 74 Harvard University engineer
2 2021-08-22 no Seungmin Choi 31 Gachon University staff
3 2021-07-01 no Chinho Park 53 Peking University engineer
4 2021-07-03 no Jihu Cho 71 Tsinghua University engineer
5 2021-06-12 yes Chunyong Park 48 University of Cambridge student
6 2021-06-04 yes Bobby Anderson 33 McGill University engineer
7 2021-06-28 yes Jihu Park 70 Australian National University student
8 2021-06-04 no Moises Barlow 57 Gachon University engineer
9 2021-06-16 yes Kyungmin Kim 45 University of Sydney marketer
10 2021-06-06 yes William Cohen 37 University of Cambridge engineer
11 2021-07-22 no Kwangsu Choi 48 Seoul National University marketer
12 2021-07-22 no Owen Martin 66 Gachon University engineer
13 2021-06-03 yes Chinho Cho 68 Gachon University student
14 2021-08-15 yes Kwangsu Cho 48 Gachon University executive
15 2021-07-12 no Tongbang Choi 26 Cornell University engineer
16 2021-08-16 yes Tongbang Kim 39 Tsinghua University student
17 2021-08-14 no Chunyong Chang 75 Gachon University student
18 2021-06-14 no Tongbang Park 32 New York University engineer
19 2021-06-07 yes Chunyong Kim 34 Harvard University staff
20 2021-07-30 yes Chinho Kim 52 Gachon University engineer
21 2021-07-21 yes Jude Smith 38 Cornell University executive
22 2021-06-29 no Tongbang Cho 29 Northwestern University marketer
23 2021-06-15 yes Seungmin Cho 71 Stanford University professor
24 2021-07-24 no Stefan Wilkerson 48 University of Melbourne executive
25 2021-06-09 no Archie Hunt 60 Fudan University student
26 2021-06-30 yes Sincere Bradley 58 University of Hong Kong staff
27 2021-08-24 no Konner French 42 Gachon University professor
28 2021-08-27 no Kwangsu Park 43 University of Pennsylvania student
29 2021-06-08 yes Bailey Houghton 31 Gachon University engineer
30 2021-07-13 yes Kyungmin Choi 44 Duke University student
```

//P5

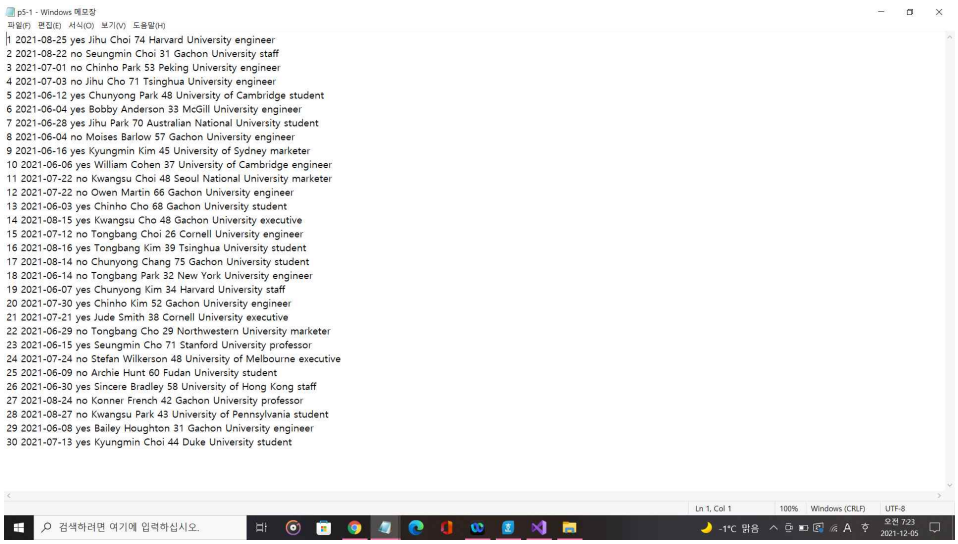
```
void p5(struct NODE* sorted_temp, int num, const char* name) {
    /*  Write the sorted data to a text file  */
    printf("\nP5\n[Write the sorted data to a text file]\n\n");
    struct NODE* ptr = sorted_temp;
    FILE* myOutFile;
    myOutFile = fopen(name, "w");

    while (ptr != NULL) {
        fprintf(myOutFile, "%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("\nSorted data is written in 'sorted data.txt'\n");
    fclose(myOutFile);
}
```

```
P5
[Write the sorted data to a text file]


Sorted data is written in 'sorted data.txt'
```

p5-1.txt



//P6-1

```
void p6_1(struct ARRAY* temp1, int* num) {
    /* All &Choi&s canceled registration in struct array */
    struct ARRAY* ptr = temp1;
    int cnt = *num;
    char del[5] = "Choi";
    printf("\nP6-1\n[All &Choi&s canceled registration in struct array]\n");
    for (int i = 0; i <cnt; i++) {
        if (strstr(ptr[i].name, del) != NULL) {
            for (int j = i; j <cnt; j++) {
                ptr[j + 1].tag--;
                ptr[j] = ptr[j + 1]; // overwrite it with the next data.
            }
            cnt--; i--;
        }
    }
    for (int j = 0; j <cnt; j++)
        print_struct_array(temp1, j);
    *num = cnt;
}
```

```
P6-1
[All  "Choi" s canceled registration in struct array]
1 2021-07-01 no Chinho Park 53 Peking University engineer
2 2021-07-03 no Jihu Cho 71 Tsinghua University engineer
3 2021-06-12 yes Chunyong Park 48 University of Cambridge student
4 2021-06-04 yes Bobby Anderson 33 McGill University engineer
5 2021-06-28 yes Jihu Park 70 Australian National University student
6 2021-06-04 no Moises Barlow 57 Gachon University engineer
7 2021-06-16 yes Kyungmin Kim 45 University of Sydney marketer
8 2021-06-06 yes William Cohen 37 University of Cambridge engineer
9 2021-07-22 no Owen Martin 66 Gachon University engineer
10 2021-06-03 yes Chinho Cho 68 Gachon University student
11 2021-08-15 yes Kwangsu Cho 48 Gachon University executive
12 2021-08-16 yes Tongbang Kim 39 Tsinghua University student
13 2021-08-14 no Chunyong Chang 75 Gachon University student
14 2021-06-14 no Tongbang Park 32 New York University engineer
15 2021-06-07 yes Chunyong Kim 34 Harvard University staff
16 2021-07-30 yes Chinho Kim 52 Gachon University engineer
17 2021-07-21 yes Jude Smith 38 Cornell University executive
18 2021-06-29 no Tongbang Cho 29 Northwestern University marketer
19 2021-06-15 yes Seungmin Cho 71 Stanford University professor
20 2021-07-24 no Stefan Wilkerson 48 University of Melbourne executive
21 2021-06-09 no Archie Hunt 60 Fudan University student
22 2021-06-30 yes Sincere Bradley 58 University of Hong Kong staff
23 2021-08-24 no Konner French 42 Gachon University professor
24 2021-08-27 no Kwangsu Park 43 University of Pennsylvania student
25 2021-06-08 yes Bailey Houghton 31 Gachon University engineer
```

//P6-2

```
void p6_2(struct NODE* head) {
    /*  All “Choi”s canceled registration in linked list  */
    int t = 0, cnt = 0, cnt_first = 0;
    struct NODE* ptr = head;
    struct NODE* pNode = NULL;
    printf("\n[All “Choi”s canceled registration in linked list]\n");
    char del[5] = "Choi";
    while (ptr != NULL) {
        if (strstr(ptr->name, del) != NULL) {
            cnt++;
            if (ptr->next == NULL) {
                pNode->next = NULL;
                free(ptr); // clear the memory
                break;
            }
            else if (t == 0) {  // if find "Choi" in first node, 'head = head->next'
                cnt_first++;
                pNode = head;
                ptr = ptr->next;
            }
            else {
                t = 1;
                pNode->next = ptr->next;
                ptr = ptr->next;
            }
        }
        else {
            t = 1;
            pNode = ptr;
            for (int j = 0; j < cnt; j++)
                ptr->tag--;
            ptr = ptr->next;
        }
    }
    for (int i = 0; i < cnt_first; i++) {
        *head = *head->next;
    }
    while (head != NULL) {
        print_linked_list(head);
        head = head->next;
    }
}
```

```
P6-2
[All "Choi" s canceled registration in linked list]
1 2021-07-01 no Chinho Park 53 Peking University engineer
2 2021-07-03 no Jihu Cho 71 Tsinghua University engineer
3 2021-06-12 yes Chunyong Park 48 University of Cambridge student
4 2021-06-04 yes Bobby Anderson 33 McGill University engineer
5 2021-06-28 yes Jihu Park 70 Australian National University student
6 2021-06-04 no Moises Barlow 57 Gachon University engineer
7 2021-06-16 yes Kyungmin Kim 45 University of Sydney marketer
8 2021-06-06 yes William Cohen 37 University of Cambridge engineer
9 2021-07-22 no Owen Martin 66 Gachon University engineer
10 2021-06-03 yes Chinho Cho 68 Gachon University student
11 2021-08-15 yes Kwangsu Cho 48 Gachon University executive
12 2021-08-16 yes Tongbang Kim 39 Tsinghua University student
13 2021-08-14 no Chunyong Chang 75 Gachon University student
14 2021-06-14 no Tongbang Park 32 New York University engineer
15 2021-06-07 yes Chunyong Kim 34 Harvard University staff
16 2021-07-30 yes Chinho Kim 52 Gachon University engineer
17 2021-07-21 yes Jude Smith 38 Cornell University executive
18 2021-06-29 no Tongbang Cho 29 Northwestern University marketer
19 2021-06-15 yes Seungmin Cho 71 Stanford University professor
20 2021-07-24 no Stefan Wilkerson 48 University of Melbourne executive
21 2021-06-09 no Archie Hunt 60 Fudan University student
22 2021-06-30 yes Sincere Bradley 58 University of Hong Kong staff
23 2021-08-24 no Konner French 42 Gachon University professor
24 2021-08-27 no Kwangsu Park 43 University of Pennsylvania student
25 2021-06-08 yes Bailey Houghton 31 Gachon University engineer
```

//P7-1

```
void p7_1(struct ARRAY* temp1, int num) {
    /*  Give Paik's tag# in consideration of the tag# of the registered persons. Add the data to the array  */
    /*  "tag#/2021-11-30/yes/Gildong Paik/35/Gachon University/Student"  */
    printf("\nP7-1\n[Add data in struct array at last index]\n\n");
    struct ARRAY* ptr = temp1;
    ptr[num].tag = num + 1;
    strcpy(ptr[num].date, "2021-11-30");
    strcpy(ptr[num].fee_paid, "yes");
    strcpy(ptr[num].name, "Gildong Paik");
    ptr[num].age = 35;
    strcpy(ptr[num].organization, "Gachon University");
    strcpy(ptr[num].job, "Student");
    /*  print added sturct array  */
    for (int j = 0; j <num + 1; j++)
        print_struct_array(ptr, j);
}
```

```
P7-1
[Add data in struct array at last index]

1 2021-07-01 no Chinho Park 53 Peking University engineer
2 2021-07-03 no Jihu Cho 71 Tsinghua University engineer
3 2021-06-12 yes Chunyong Park 48 University of Cambridge student
4 2021-06-04 yes Bobby Anderson 33 McGill University engineer
5 2021-06-28 yes Jihu Park 70 Australian National University student
6 2021-06-04 no Moises Barlow 57 Gachon University engineer
7 2021-06-16 yes Kyungmin Kim 45 University of Sydney marketer
8 2021-06-06 yes William Cohen 37 University of Cambridge engineer
9 2021-07-22 no Owen Martin 66 Gachon University engineer
10 2021-06-03 yes Chinho Cho 68 Gachon University student
11 2021-08-15 yes Kwangsu Cho 48 Gachon University executive
12 2021-08-16 yes Tongbang Kim 39 Tsinghua University student
13 2021-08-14 no Chunyong Chang 75 Gachon University student
14 2021-06-14 no Tongbang Park 32 New York University engineer
15 2021-06-07 yes Chunyong Kim 34 Harvard University staff
16 2021-07-30 yes Chinho Kim 52 Gachon University engineer
17 2021-07-21 yes Jude Smith 38 Cornell University executive
18 2021-06-29 no Tongbang Cho 29 Northwestern University marketer
19 2021-06-15 yes Seungmin Cho 71 Stanford University professor
20 2021-07-24 no Stefan Wilkerson 48 University of Melbourne executive
21 2021-06-09 no Archie Hunt 60 Fudan University student
22 2021-06-30 yes Sincere Bradley 58 University of Hong Kong staff
23 2021-08-24 no Konner French 42 Gachon University professor
24 2021-08-27 no Kwangsu Park 43 University of Pennsylvania student
25 2021-06-08 yes Bailey Houghton 31 Gachon University engineer
26 2021-11-30 yes Gildong Paik 35 Gachon University Student
```

//P7-2

```
void p7_2(struct NODE* temp2, int* num) {
    /* add data in sorted linked-list at last, "tag#/2021-11-30/yes/Gildong Paik/35/Gachon University/Student" */
    printf("\nP7-2\n[Add data in linked-list at last node]\n\n");
    struct NODE* ptr = temp2;
    struct NODE* tmp;
    char name[NAME] = "Gildong Paik";
    char date[DATE] = "2021-11-30";
    char fee_paid[FEE_PAID] = "yes";
    char organization[ORGANIZATION] = "Gachon University";
    char job[JOB] = "Student";
    while (1) {
        if (ptr->next == NULL) {
            tmp = createNewNode(*num + 1, date, fee_paid, name, 35, organization, job);
            linkNode(ptr, tmp);
            break;
        }
        ptr = ptr->next;
    }
    /*  print added linked-list  */
    while (temp2 != NULL) {
        print_linked_list(temp2);
        temp2 = temp2->next;
    }
    num++;
}
```

```
P7-2
[Add data in linked-list at last node]

1 2021-07-01 no Chinho Park 53 Peking University engineer
2 2021-07-03 no Jihu Cho 71 Tsinghua University engineer
3 2021-06-12 yes Chunyong Park 48 University of Cambridge student
4 2021-06-04 yes Bobby Anderson 33 McGill University engineer
5 2021-06-28 yes Jihu Park 70 Australian National University student
6 2021-06-04 no Moises Barlow 57 Gachon University engineer
7 2021-06-16 yes Kyungmin Kim 45 University of Sydney marketer
8 2021-06-06 yes William Cohen 37 University of Cambridge engineer
9 2021-07-22 no Owen Martin 66 Gachon University engineer
10 2021-06-03 yes Chinho Cho 68 Gachon University student
11 2021-08-15 yes Kwangsu Cho 48 Gachon University executive
12 2021-08-16 yes Tongbang Kim 39 Tsinghua University student
13 2021-08-14 no Chunyong Chang 75 Gachon University student
14 2021-06-14 no Tongbang Park 32 New York University engineer
15 2021-06-07 yes Chunyong Kim 34 Harvard University staff
16 2021-07-30 yes Chinho Kim 52 Gachon University engineer
17 2021-07-21 yes Jude Smith 38 Cornell University executive
18 2021-06-29 no Tongbang Cho 29 Northwestern University marketer
19 2021-06-15 yes Seungmin Cho 71 Stanford University professor
20 2021-07-24 no Stefan Wilkerson 48 University of Melbourne executive
21 2021-06-09 no Archie Hunt 60 Fudan University student
22 2021-06-30 yes Sincere Bradley 58 University of Hong Kong staff
23 2021-08-24 no Konner French 42 Gachon University professor
24 2021-08-27 no Kwangsu Park 43 University of Pennsylvania student
25 2021-06-08 yes Bailey Houghton 31 Gachon University engineer
26 2021-11-30 yes Gildong Paik 35 Gachon University Student
```

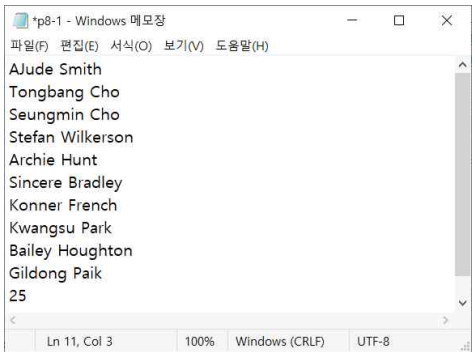
//P8-1

```
void p8_1(struct ARRAY* temp1, int num, unsigned char* org_check_sum) {
    /*  Copy the most recent data in the array then make "p8-1.txt" and checksum.  */
    struct ARRAY* ptr = temp1;
    int cnt = num;
    // last 10 data in the array write in "p8-1.txt"
    FILE* myOutFile = fopen("p8-1.txt", "w"); // original textfile.
    for (int i = cnt - 9; i <= cnt; i++) {
        fprintf(myOutFile, "%s\n", ptr[i].name);
    }
    // checksum in original textfile.
    unsigned char name = 0;
    unsigned char check_sum = { NULL };
    for (int i = cnt - 9; i <= cnt; i++) {
        for (int j = 0; j < strlen(ptr[i].name); j++) {
            if (j == 0)
                name = ptr[i].name[j];
            else
                name += ptr[i].name[j];
        }
        if (i == cnt - 9)
            check_sum = name;
        else
            check_sum ^= name;
    }
    fprintf(myOutFile, "%d\n", check_sum);
    printf("\nOriginal data checksum : %d\n", check_sum);
    fclose(myOutFile);
    *org_check_sum = check_sum;
}
```

```
P8-1
[Copy the most recent data in the array then make 'p8 - 1.txt' and checksum]


Original data checksum : 25
```

P8-2.txt



//P8-2

```
void p8_2(unsigned char org_check_sum) {
    /*  Copy the "p8-1.txt" at "p8-2.txt" and check the checksum. If checksum is same, the copy is success.
*/
    struct ARRAY checksum[11] = { NULL };
    char line[NAME];
    unsigned char name = 0;
    unsigned char check_sum = 0;
    /*  Read the original textfile. */
    FILE* myInFile = fopen("p8-1.txt", "r");
    for (int i = 0; i < 10; i++) {
        fgets(line, NAME, myInFile);
        strcpy(checksum[i].name, strtok(line, "\n"));
    }
    fclose(myInFile);
    /*  checksum in copy textfile.  */
    FILE* myOutFile = fopen("p8-2.txt", "w");
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < strlen(checksum[i].name); j++) {
            if (j == 0)
                name = checksum[i].name[j];
            else
                name += checksum[i].name[j];
        }
        if (i == 0)
            check_sum = name;
        else
            check_sum ^= name;
    }
    /*  write a copy textfile.  */
    for (int i = 0; i < 10; i++)
        fprintf(myOutFile, "%s\n", checksum[i].name);
    fprintf(myOutFile, "%d\n", check_sum);
    fclose(myOutFile);

    printf("\nCopy data checksum : %d\n", check_sum);
    /*  Compare the checksum against the checksum in the original data.  */
    if (org_check_sum == check_sum)
        printf("\nVerification of successful copy\n");
    else
        printf("\nFail of data copy\n");
}
```

```
P8-2
[Copy the 'p8-1.txt' at 'p8-2.txt' and check the checksum]


Copy data checksum : 25

Verification of successful copy
```

P8-2.txt