< 1. Understand the Problem >

[Draw the problem & create examples.]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| tag | data | fee\_paid | name | age | oraganization | job |
| 29 | 2022.6.8 | yes | Hougthon | 31 | Gachon | engineer |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| tag | data | fee\_paid | name | age | oraganization | job |
| 22 | 2002.6.29 | no | Cho | 29 | Northwestern | marketer |
| 29 | 2022.6.8 | yes | Hougthon | 31 | Gachon | engineer |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| tag | data | fee\_paid | name | age | oraganization | job |
| 15 | 20022.7.12 | no | Choi | 26 | Cornell | engineer |
| 22 | 2002.6.29 | no | Cho | 29 | Northwestern | marketer |
| 29 | 2022.6.8 | yes | Hougthon | 31 | Gachon | engineer |

asfsafssafsadfsa node1 node2 node3

29 | node3

31 | NULL

26 | node2

[Identify cases to consider.]

- Sort by age  
- Connecting nodes & linked lists

- Output specific information only

- Insert additional people

- Delete Information & Unmemory

< 2. Outline a solution >

// 데이터 수만큼 루프 & 데이터 나이순 정렬

Loop through the Data struct's number and the struct's age

Sort the struct Data by age

// 정렬된 데이터 만큼 루프 돌기 & 데이터 링킹

Loop through the Sorted Data and the Node struct

Linking the Data to Node

// 노드 루프돌기 & 특정 정보 출력하기

Loop Until the node next is not NULL

if find information

print struct ( tag, data ...etc )

// 노드 루프돌기 & 특정 정보삭제하기

Loop Until the node next is not NULL

if find information

Delete Node & free

// 특정 나이까지 루프 & 정보 삽입

Loop Until the extra age is less than Node age

insert extra data

< 3. Form a program struct >

**[Flow graph]**

File sort

File input

Make node

sort File linking

Find “fee-yes”

print data

Find “staff”

Delete data

Find age & linking

Insert data

**[Form a program Structure]**

struct {

- input data (typedef struct \_Data)

- inking Node (typedef struct \_Node)

} Data & Node

void File procesing {

- verifies file (void verify\_file)

- verifies memory (void verify\_memory)

- opens file (FILE\* open\_file)

- write file (void print\_data)

- read file data (void read\_data)

- file sort (void sort\_data) }

void additonal features {

- linking node (insert\_node)

- print fee paid (void is\_paid)

- extra human (void add\_human)

- delete human (void Delete)

- free linked list (void free\_linked\_list) }

main ()

< 4. Write a Program Outline : Pseudo Code >

void sort\_data(Data data[MAX]){

for (int i = 0; i < MAX; i++)

for (int j = 0; j < MAX - i - 1; j++)

if (data[j].age > data[j + 1].age) exchange data

}

void insert\_node(Node\* head, Data data){

malloc - create New node

new->data = data;

new->next = NULL;

ptr = head;

While( ! NUll ) ptr = ptr->next;

ptr->next = new

}

void is\_paid(Node\* head){

while( Node != NULL)

if( find information)

print (information)

}

void add\_human(Node\* head){

create extra Node

while ( Until find information)

prev\_ptr = ptr;

ptr = ptr->next;

prev\_ptr->next = extra\_node;

extra\_node->next = ptr;

}

void Delete(Node\* head){

while(Node != NULL)

if (find information)

cunnect Node & re connet Node

free( Delete information Node)

}

main()