To Implement linked list which has employ data

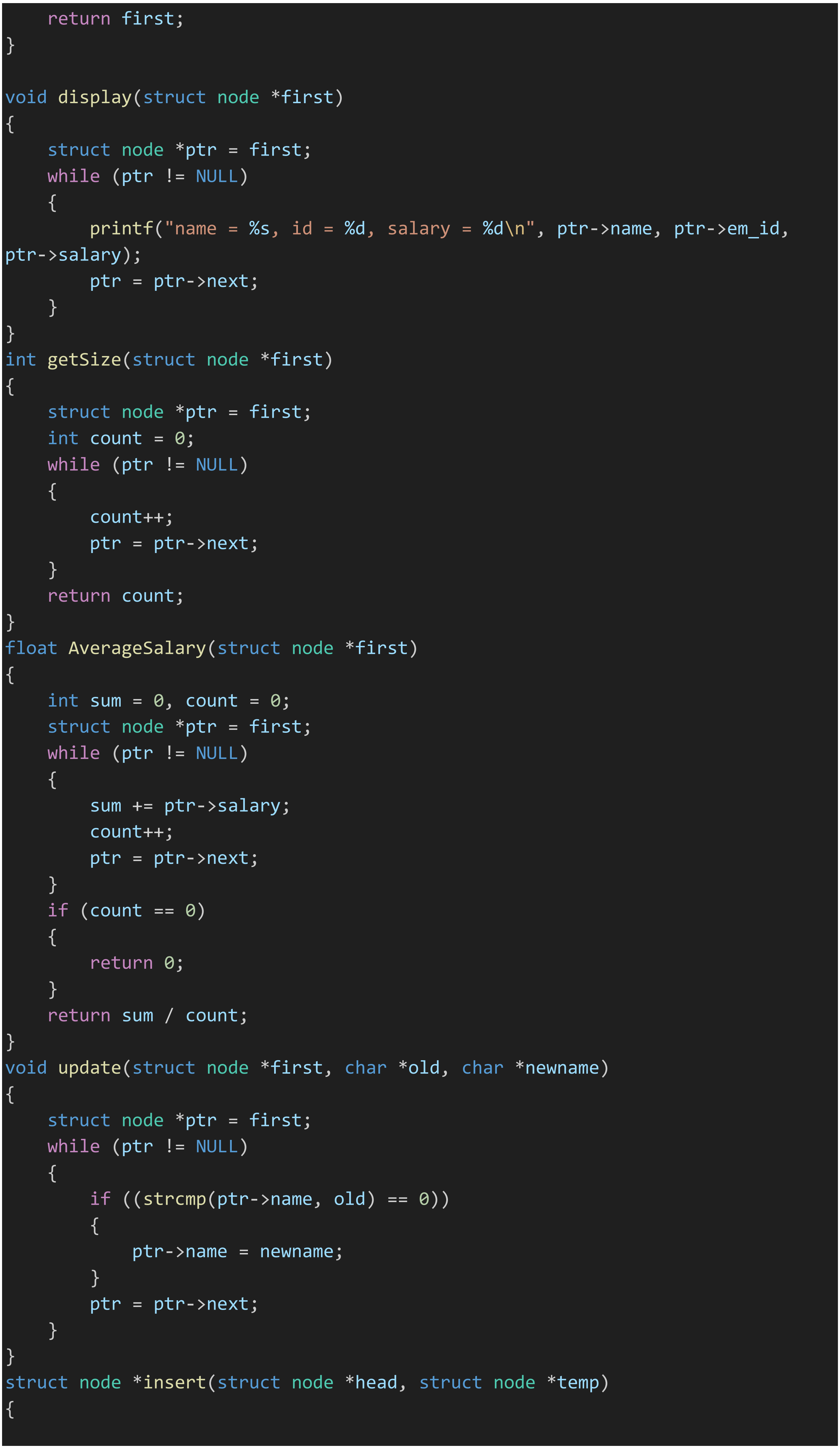
Name :- SUJAL NIMJE

Roll no :- 64

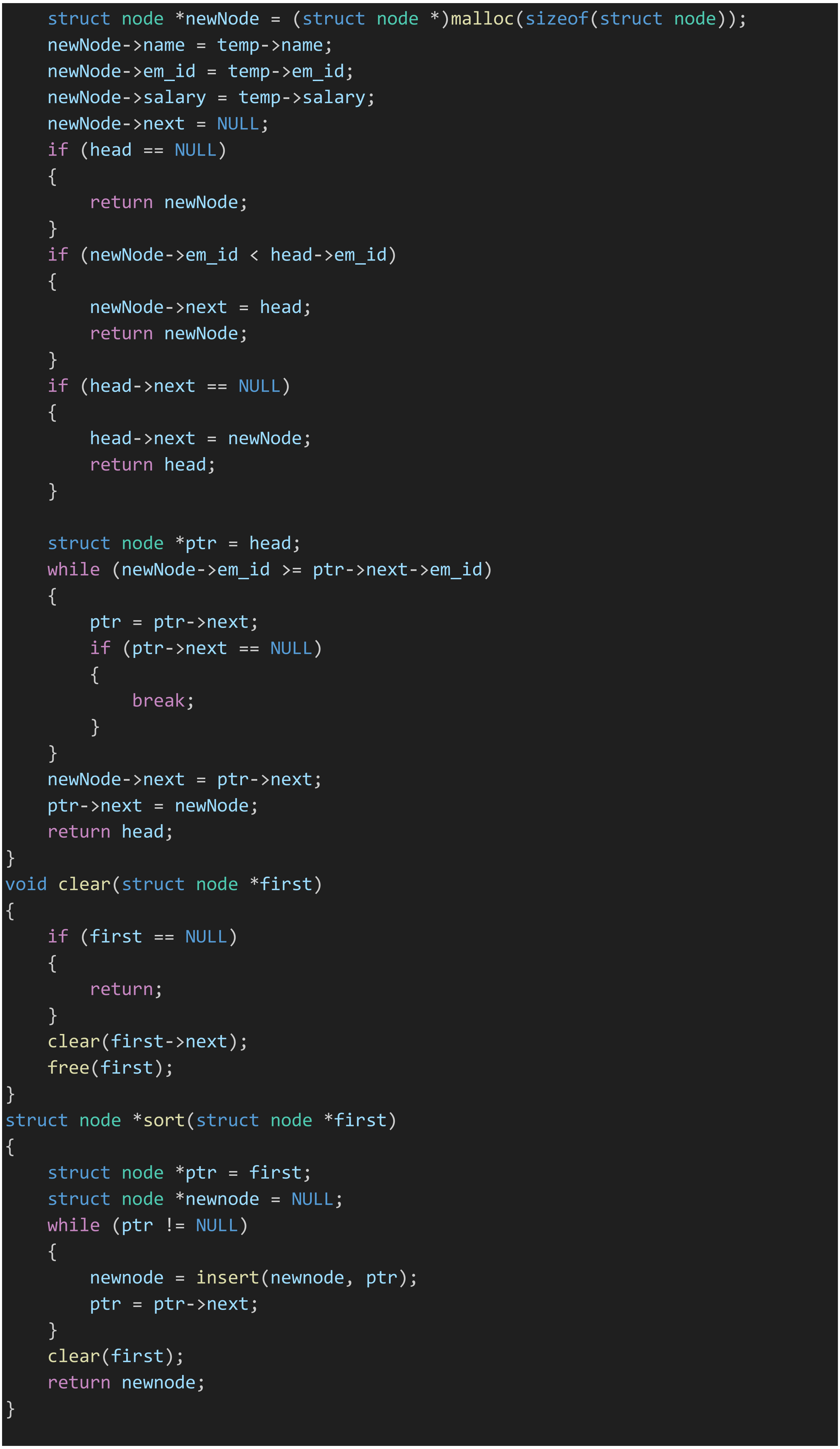
Subject :- DS, practical no : - 5

Code :-

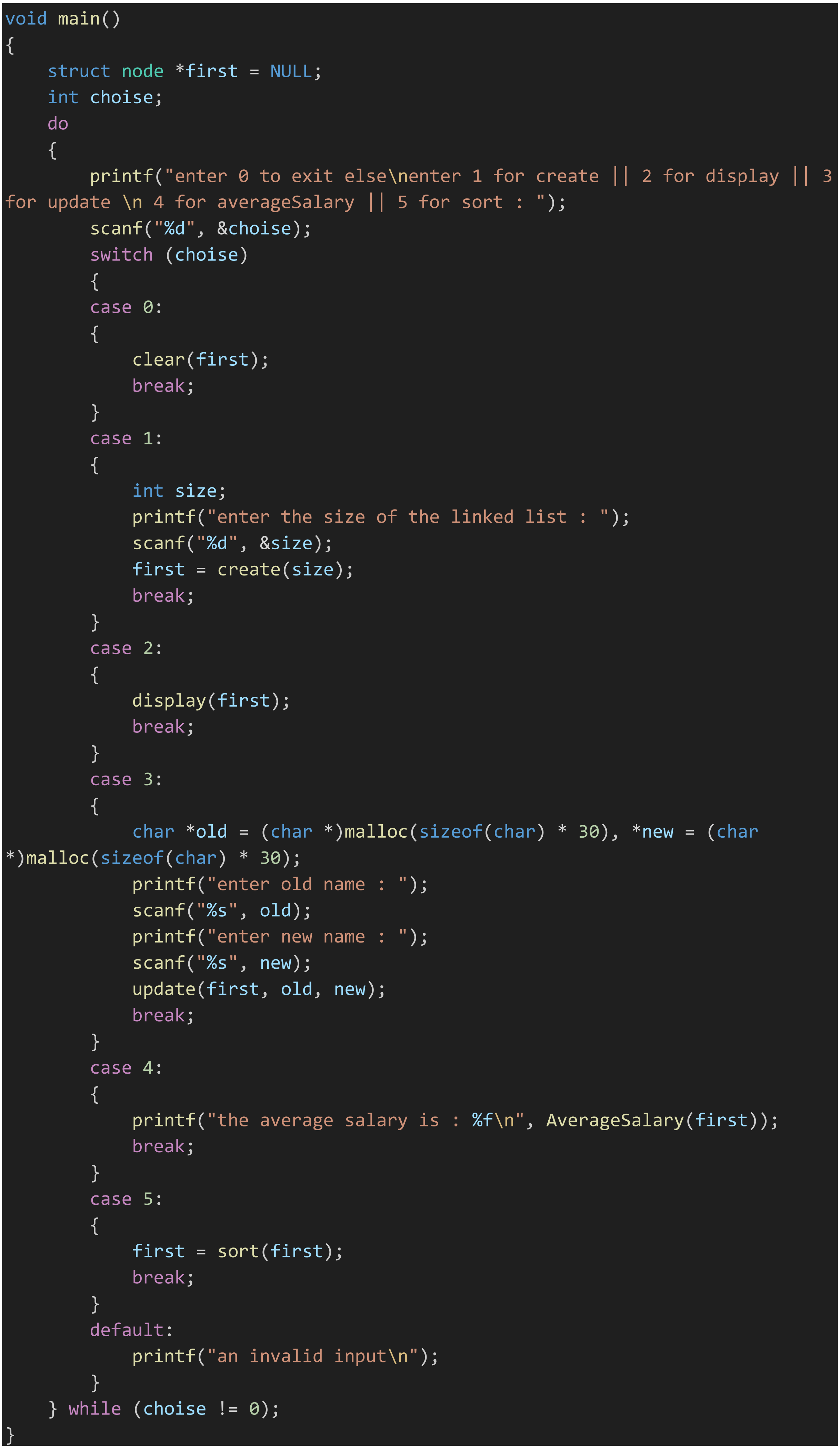
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct node
{
    char *name;
    int em_id;
    int salary;
    struct node *next;
};
struct node *create(int size)
{
    struct node *first = NULL;
    struct node *ptr = NULL;

    while (size > 0)
    {
        struct node *temp = (struct node *)malloc(sizeof(struct node));
        printf("enter the employ name : ");
        char *name = (char *)malloc(sizeof(char) * 30);
        int id, salary;
        scanf("%s", name);
        printf("enter id : ");
        scanf("%d", &id);
        printf("enter salary : ");
        scanf("%d", &salary);
        temp->name = name;
        temp->em_id = id;
        temp->salary = salary;
        temp->next = NULL;
        if (first == NULL)
        {
            first = ptr = temp;
        }
        else
        {
            ptr->next = temp;
            ptr = temp;
        }
        printf("\n");
        size--;
    }
```



```
    return first;
}

void display(struct node *first)
{
    struct node *ptr = first;
    while (ptr != NULL)
    {
        printf("name = %s, id = %d, salary = %d\n", ptr->name, ptr->em_id,
ptr->salary);
        ptr = ptr->next;
    }
}
int getSize(struct node *first)
{
    struct node *ptr = first;
    int count = 0;
    while (ptr != NULL)
    {
        count++;
        ptr = ptr->next;
    }
    return count;
}
float AverageSalary(struct node *first)
{
    int sum = 0, count = 0;
    struct node *ptr = first;
    while (ptr != NULL)
    {
        sum += ptr->salary;
        count++;
        ptr = ptr->next;
    }
    if (count == 0)
    {
        return 0;
    }
    return sum / count;
}
void update(struct node *first, char *old, char *newname)
{
    struct node *ptr = first;
    while (ptr != NULL)
    {
        if ((strcmp(ptr->name, old) == 0))
        {
            ptr->name = newname;
        }
        ptr = ptr->next;
    }
}
struct node *insert(struct node *head, struct node *temp)
{
```

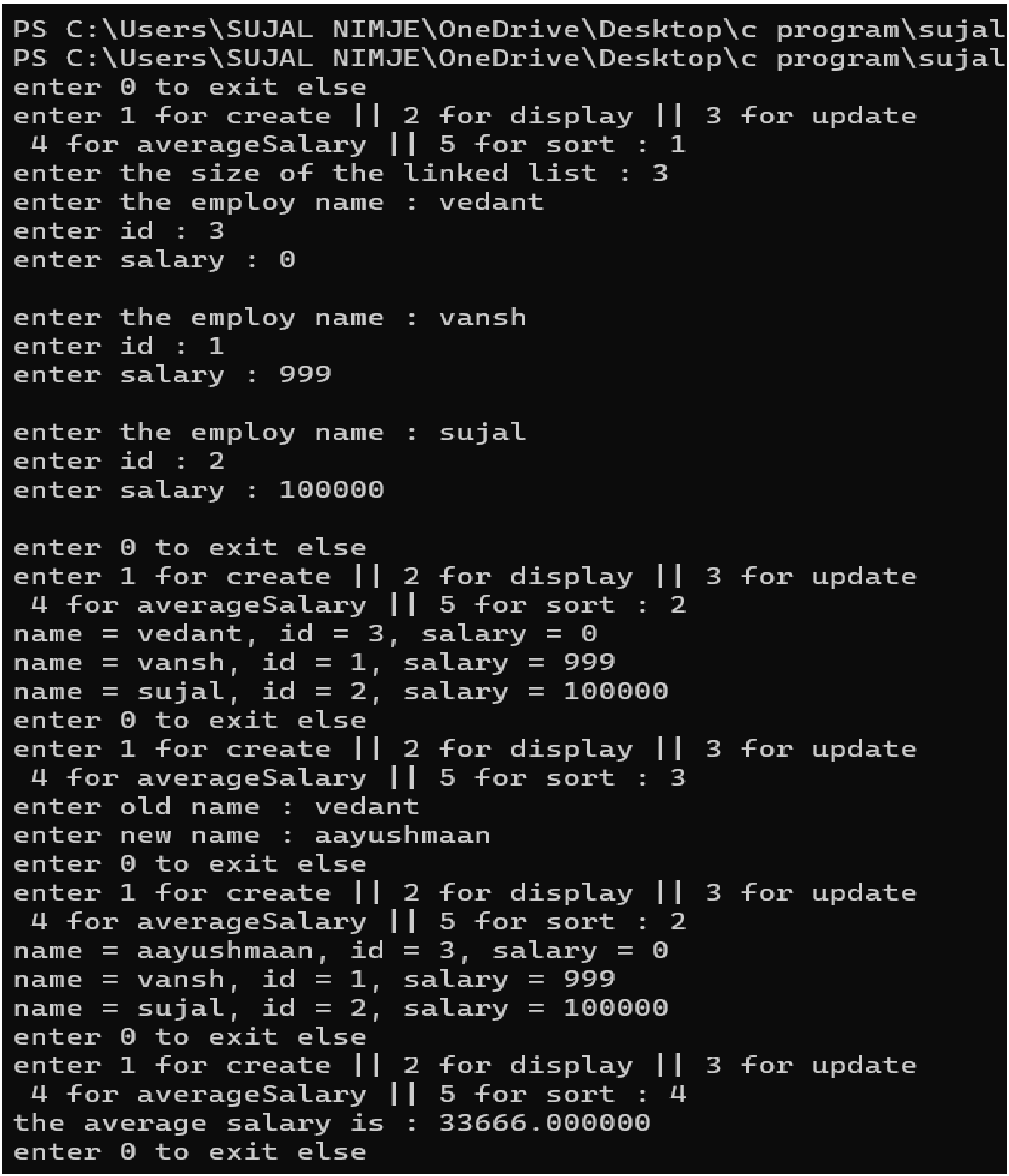


```
    struct node *newNode = (struct node *)malloc(sizeof(struct node));
    newNode->name = temp->name;
    newNode->em_id = temp->em_id;
    newNode->salary = temp->salary;
    newNode->next = NULL;
    if (head == NULL)
    {
        return newNode;
    }
    if (newNode->em_id < head->em_id)
    {
        newNode->next = head;
        return newNode;
    }
    if (head->next == NULL)
    {
        head->next = newNode;
        return head;
    }

    struct node *ptr = head;
    while (newNode->em_id >= ptr->next->em_id)
    {
        ptr = ptr->next;
        if (ptr->next == NULL)
        {
            break;
        }
    }
    newNode->next = ptr->next;
    ptr->next = newNode;
    return head;
}
void clear(struct node *first)
{
    if (first == NULL)
    {
        return;
    }
    clear(first->next);
    free(first);
}
struct node *sort(struct node *first)
{
    struct node *ptr = first;
    struct node *newnode = NULL;
    while (ptr != NULL)
    {
        newnode = insert(newnode, ptr);
        ptr = ptr->next;
    }
    clear(first);
    return newnode;
}
```

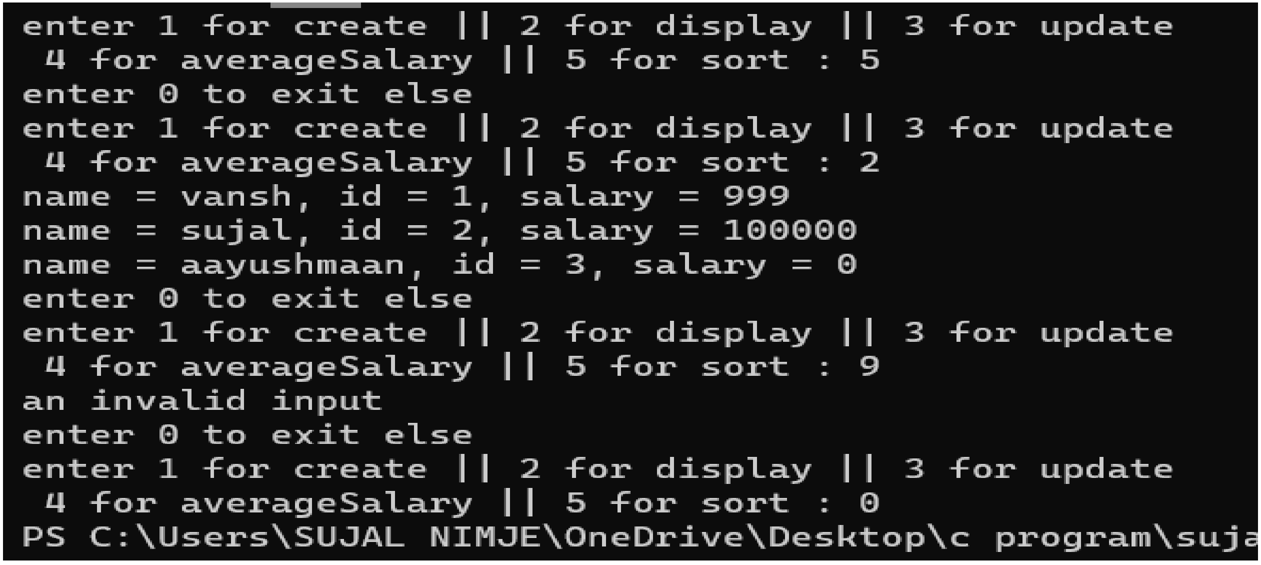


```c
void main()
{
    struct node *first = NULL;
    int choise;
    do
    {
        printf("enter 0 to exit else\nenter 1 for create || 2 for display || 3 for update \n 4 for averageSalary || 5 for sort : ");
        scanf("%d", &choise);
        switch (choise)
        {
        case 0:
        {
            clear(first);
            break;
        }
        case 1:
        {
            int size;
            printf("enter the size of the linked list : ");
            scanf("%d", &size);
            first = create(size);
            break;
        }
        case 2:
        {
            display(first);
            break;
        }
        case 3:
        {
            char *old = (char *)malloc(sizeof(char) * 30), *new = (char *)malloc(sizeof(char) * 30);
            printf("enter old name : ");
            scanf("%s", old);
            printf("enter new name : ");
            scanf("%s", new);
            update(first, old, new);
            break;
        }
        case 4:
        {
            printf("the average salary is : %f\n", AverageSalary(first));
            break;
        }
        case 5:
        {
            first = sort(first);
            break;
        }
        default:
            printf("an invalid input\n");
        }
    } while (choise != 0);
}
```

Output :-



```
PS C:\Users\SUJAL NIMJE\OneDrive\Desktop\c program\sujal
PS C:\Users\SUJAL NIMJE\OneDrive\Desktop\c program\sujal
enter 0 to exit else
enter 1 for create || 2 for display || 3 for update
 4 for averageSalary || 5 for sort : 1
enter the size of the linked list : 3
enter the employ name : vedant
enter id : 3
enter salary : 0

enter the employ name : vansh
enter id : 1
enter salary : 999

enter the employ name : sujal
enter id : 2
enter salary : 100000

enter 0 to exit else
enter 1 for create || 2 for display || 3 for update
 4 for averageSalary || 5 for sort : 2
name = vedant, id = 3, salary = 0
name = vansh, id = 1, salary = 999
name = sujal, id = 2, salary = 100000
enter 0 to exit else
enter 1 for create || 2 for display || 3 for update
 4 for averageSalary || 5 for sort : 3
enter old name : vedant
enter new name : aayushmaan
enter 0 to exit else
enter 1 for create || 2 for display || 3 for update
 4 for averageSalary || 5 for sort : 2
name = aayushmaan, id = 3, salary = 0
name = vansh, id = 1, salary = 999
name = sujal, id = 2, salary = 100000
enter 0 to exit else
enter 1 for create || 2 for display || 3 for update
 4 for averageSalary || 5 for sort : 4
the average salary is : 33666.000000
enter 0 to exit else
```



```
enter 1 for create || 2 for display || 3 for update
 4 for averageSalary || 5 for sort : 5
enter 0 to exit else
enter 1 for create || 2 for display || 3 for update
 4 for averageSalary || 5 for sort : 2
name = vansh, id = 1, salary = 999
name = sujal, id = 2, salary = 100000
name = aayushmaan, id = 3, salary = 0
enter 0 to exit else
enter 1 for create || 2 for display || 3 for update
 4 for averageSalary || 5 for sort : 9
an invalid input
enter 0 to exit else
enter 1 for create || 2 for display || 3 for update
 4 for averageSalary || 5 for sort : 0
PS C:\Users\SUJAL NIMJE\OneDrive\Desktop\c program\suja
```