**Name: Adithya MSRN: PES1UG20CS621 SEC: K**

**WEEK 3**

**1. Create an telephone directory containing the customers details as : Name, Phone no, Address, Area . Create an ordered doubly list based on name**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct NODE

{

char \*name;

int phone\_number;

char \*address;

char \*area;

struct NODE\* prev;

struct NODE\* next;

}NODE;

typedef struct CustomerDetails

{

NODE\* head;

}CustomerDetails;

void init(CustomerDetails\* cust);

void add(CustomerDetails\* cust);

void display(CustomerDetails\* cust);

void free\_list(CustomerDetails\* cust);

int main()

{

CustomerDetails cust;

init(&cust);

int n;

while(1)

{

printf("\n1. Add\n2. Display\n3. Free List\n\n");

scanf("%d", &n);

printf("\n");

switch(n)

{

case 1:

add(&cust);

break;

case 2:

display(&cust);

break;

case 3:

free\_list(&cust);

return 0;

break;

}

}

}

void init(CustomerDetails\* cust)

{

cust -> head = NULL;

}

void add(CustomerDetails\* cust)

{

NODE\* temp = (NODE\*) malloc(sizeof(NODE));

temp->name=(char\*)malloc(20);

temp->phone\_number=(int\*)malloc(20);

temp->address=(char\*)malloc(20);

temp->area=(char\*)malloc(20);

printf("Name: ");

scanf("%s", temp -> name);

printf("Phone-Number: ");

scanf("%d", &(temp -> phone\_number));

printf("Address: ");

scanf("%s", temp -> address);

printf("Area: ");

scanf("%s", temp -> area);

if(cust -> head == NULL)

{

temp -> next = NULL;

temp -> prev = NULL;

cust -> head = temp;

}

else

{

NODE\* current = cust -> head;

NODE\* prev\_n = NULL;

while(current!=NULL)

{

if(strcmp(temp -> name, current -> name) < 0)

break;

prev\_n = current;

current = current -> next;

}

if(prev\_n==NULL)

{

temp -> next = current;

current -> prev = temp;

temp -> prev = NULL;

cust -> head = temp;

}

else

{

temp -> next = current;

temp -> prev = prev\_n;

prev\_n -> next = temp;

if(current!=NULL)

current -> prev = temp;

}

}

}

void display(CustomerDetails\* cust)

{

NODE\* temp = cust -> head;

while(temp)

{

printf("Name %s\n", temp -> name);

printf("Phone Num %d\n", temp -> phone\_number);

printf("Address %s\n", temp -> address);

printf("Area %s\n", temp -> area);

printf("\n");

temp = temp -> next;

}

}

void free\_list(CustomerDetails\* cust)

{

NODE\* temp = NULL;

while(cust -> head)

{

temp = cust -> head;

cust -> head = cust -> head -> next;

temp -> next = NULL;

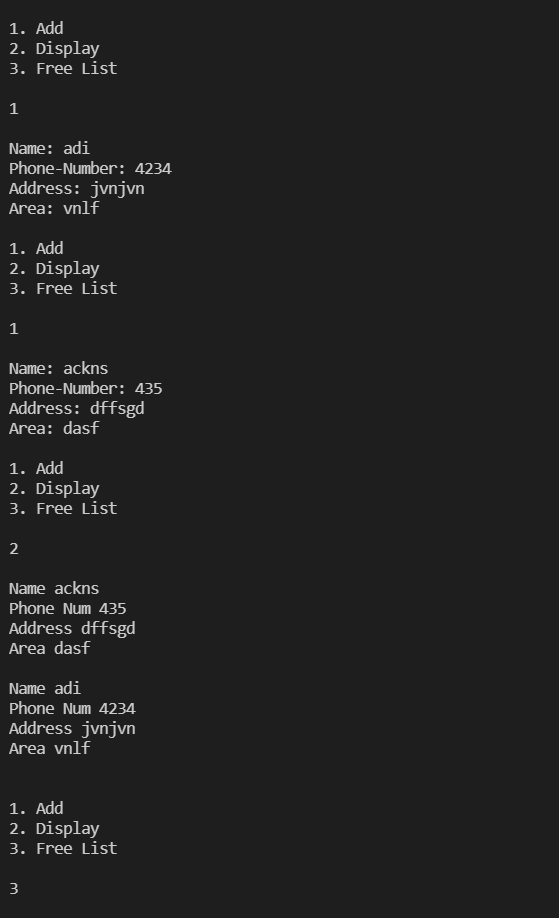
temp -> prev = NULL;

free(temp);

}

}

**Output**



**2. Create an Employee with the following fields – SSN, Name, Dept, Designations, Salary, Phone no, Age. Every employee should work in a department and department shoud have a name and number associated with it.**

**Client.c**

#include <stdio.h>

#include <stdlib.h>

#include "header.h"

int main()

{

Employees e;

int n;

init(&e);

while (1)

{

printf("1. ADD 2. DELETE 3. DISPLAY 4.EXIT\n");

scanf("%d", &n);

switch (n)

{

case 1:

insert(&e);

break;

case 2:

delete\_58(&e);

break;

case 3:

display(&e);

break;

}

}

}

**Header.h**

typedef struct Department

{

char name[20];

int number;

} Department;

typedef struct Node

{

char ssn[20];

char name[20];

Department \*department;

char designation[20];

int salary;

int phone\_number;

int age;

struct Node \*next, \*prev;

} node\_t;

typedef struct Employees

{

node\_t \*head;

} Employees;

void init(Employees \*p);

void insert(Employees \*p);

void display(Employees \*p);

void delete\_58(Employees \*p);

**Server.c**

#include <stdio.h>

#include <stdlib.h>

Server.c

#include <string.h>

#include "header.h"

void init(Employees \*p)

{

p->head = NULL;

}

void display(Employees \*p)

{

if (p->head == NULL)

{

printf("Empty list");

}

else

{

node\_t \*present = p->head;

while (present != NULL)

{

printf("SSN :%s\n", present->ssn);

printf("Name :%s\n", present->name);

printf("department name :%s\n", present->department->name);

printf("department number :%d\n", present->department->number);

printf("designation :%s\n", present->designation);

printf("salary :%d\n", present->salary);

printf("phone number :%d\n", present->phone\_number);

printf("age :%d\n", present->age);

printf("\n");

present = present->next;

}

}

}

node\_t \*createnode()

{

node\_t \*temp = (node\_t \*)malloc(sizeof(node\_t));

Department \*d = (Department \*)malloc(sizeof(Department));

temp->department = d;

printf("SSN :");

scanf("%s", temp->ssn);

printf("Name :");

scanf("%s", temp->name);

printf("department name :");

scanf("%s", temp->department->name);

printf("department number :");

scanf("%d", &temp->department->number);

printf("designation :");

scanf("%s", temp->designation);

printf("salary :");

scanf("%d", &temp->salary);

printf("phone number :");

scanf("%d", &temp->phone\_number);

printf("age :");

scanf("%d", &temp->age);

temp->next = NULL;

temp->prev = NULL;

printf("\n");

return temp;

}

void insert(Employees \*p)

{

node\_t \*temp = createnode();

if (p->head == NULL)

{

p->head = temp;

}

else

{

temp->next = p->head;

p->head->prev = temp;

p->head = temp;

}

}

void delete\_58(Employees \*p)

{

if (p->head == NULL)

{

printf("Empty!");

}

else

{

int state = 1;

while (state == 1)

{

if (p->head == NULL)

{

state = 0;

}

else

{

node\_t \*present = p->head;

node\_t \*prev = NULL;

if (present->next == NULL && (present->age > 58))

{

free(present);

p->head = NULL;

}

else

{

while (present != NULL && ((present->age) < 58))

{

prev = present;

present = present->next;

}

if (present == NULL)

{

state = 0;

}

else if (prev == NULL)

{

p->head = present->next;

p->head->prev = NULL;

free(present);

}

else

{

prev->next = present->next;

if (present->next != NULL)

present->next->prev = prev;

free(present);

}

}

}

}

}

}

**Output:**

