Program 1:

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct Node

{

char name[50];

int phone\_number;

char address[500];

char area[50];

struct Node \*next, \*prev;

} Node;

typedef struct CustomerDetails

{

Node \*head;

} CustomerDetails;

void init(CustomerDetails \*p);

void disp(CustomerDetails \*p);

void ordered\_insert(CustomerDetails \*p,char \*name, int p\_no,char \*address, char \*area);

int main()

{

CustomerDetails l;

init(&l);

int choice;

printf("1: Enter details\n2: Display details\n");

scanf("%d",&choice);

char name[50];

int p\_no;

char address[500];

char area[50];

while(choice)

{

switch(choice)

{

case 1:

printf("Enter the Name\n");

scanf("%s",name);

printf("Enter the Phone Number\n");

scanf("%d", &(p\_no));

printf("Enter the Address\n");

scanf("%s", address);

printf("Enter the Area\n");

scanf("%s", area);

ordered\_insert(&l,name, p\_no,address,area);

break;

case 2:

disp(&l);

break;

default:

printf("INVALID CHOICE!!! Retry!!!");

//exit(0);

}

printf("1: Enter details\n2: Display details\n");

scanf("%d",&choice);

}

return 0;

}

void init(CustomerDetails \*p)

{

p->head=NULL;

}

void disp(CustomerDetails \*p)

{

Node \*pres=p->head;

if(p->head==NULL)

{

printf("Empty list\n");

}

else{

printf("NAME\tPHONE NUMBER\t\tADDRESS\t\tAREA\n");

while(pres!=NULL)

{

printf("%s \t%d \t%s \t%s ",pres->name, pres->phone\_number,pres->address,pres->area);

printf("\n");

pres=pres->next;

}

printf("\n");

}

}

Node\* create\_node(char \*name, int p\_no,char \*address,char \*area)

{

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

strcpy(temp->name, name);

temp->phone\_number=p\_no;

strcpy(temp->address, address);

strcpy(temp->area, area);

temp->next=NULL;

temp->prev=NULL;

return temp;

}

void ordered\_insert(CustomerDetails \*p,char \*name, int p\_no,char \*address, char \*area)

{

Node \*temp=create\_node(name,p\_no,address,area);

if(p->head==NULL) //when list is empty

p->head=temp;

else //list is not empty

{

Node \*present=p->head;

Node \*previous=NULL;

while(present!=NULL && strcmp(present->name, name)<0)

{

previous=present;

present=present->next;

}

if(previous==NULL) //front insertion

{

p->head=temp;

temp->next=present;

present->prev = temp;

}

else //middle and end insertion

{

temp->next=present;

present->prev=temp;

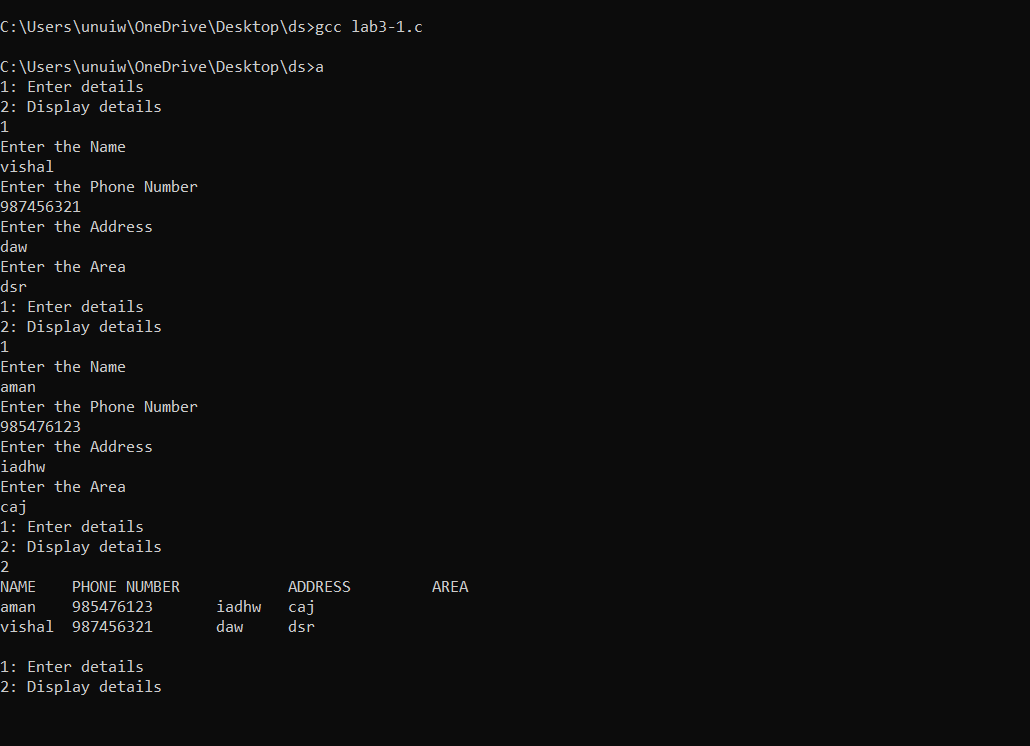
previous->next=temp;

temp->prev = previous;

}

}

}



Program 2:

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

typedef struct Department{

char name[25];

int number;

} Dep;

typedef struct Node{

char ssn[15];

char name[25];

Dep department;

char designations[20];

int salary;

int phone\_number;

int age;

struct Node \*next, \*prev;

} Node;

typedef struct Employees {

Node \*head;

} Emp;

void init(Emp \*li);

void insert(Emp \*p,char \*name, char \*ssn,char \*dep,char \*des,int age,int sal,int p\_n,int code);

void del(Emp \*p);

void display(Emp \*p,char des[]);

int main()

{

Emp list;

init(&list);

int choice=2;

while (choice) {

printf("Enter SSN, Employee name,Department,designations,department code,salary,phone number and age\n");

int s,p,a,c;

char n[15],m[25],d[20],dp[25];

scanf("%s",n );

fflush(stdout);

scanf("%s",m );

fflush(stdout);

scanf("%s",dp);

fflush(stdout);

scanf("%s",d);

fflush(stdout);

scanf("%d",&c);

scanf("%d",&s);

scanf("%d",&p);

scanf("%d",&a);

insert(&list,n,m,dp,d,a,s,p,c);

printf("Enter 0 to exit");

scanf("%d",&choice);

}

del(&list);

choice=1;

while (choice) {

printf("Enter department of which u want details:");

char dep[20];

scanf("%s",dep);

display(&list,dep);

printf("Enter 0 to exit");

scanf("%d",&choice);

}

return 0;

}

void init(Emp \*li)

{

li->head=NULL;

}

Node\* create\_node(char \*name, char \*ssn,char \*dep,char \*des,int age,int sal,int p\_n,int code)

{

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

strcpy(temp->name, name);

temp->phone\_number=p\_n;

strcpy(temp->ssn, ssn);

strcpy(temp->designations, des);

strcpy(temp->department.name, dep);

temp->salary=sal;

temp->age=age;

temp->department.number=code;

temp->next=NULL;

temp->prev=NULL;

return temp;

}

void insert(Emp \*p,char \*ssn, char \*name,char \*dep,char \*des,int age,int sal,int p\_n,int code)

{

Node \*temp=create\_node(name,ssn,dep,des,age,sal,p\_n,code);

if(p->head==NULL)

p->head=temp;

else

{

temp->next=p->head;

p->head->prev=temp;

p->head=temp;

}

}

void del(Emp \*p)

{

Node \*temp=p->head;

while(temp!=NULL)

{

if(temp->age>58)

{

if(temp->prev==NULL)

{

temp=temp->next;

p->head=temp;

free(temp->prev);

temp->prev=NULL;

}

else if(temp->next==NULL)

{

temp->prev->next=NULL;

free(temp);

}

else{

temp->prev->next=temp->next;

temp->next->prev=temp->prev;

free(temp);

}

}

temp=temp->next;

}

}

void display(Emp \*p,char des[])

{

printf("display\n");

Node \*temp=p->head;

while(temp!=NULL)

{

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

if(strcmp(temp->department.name,des)==0)

{

printf("\nSSN:%s\nName:%s\nDesignation:%s\nSalary:%d\nPhone Number:%d\nage:%d\n\n",temp->ssn,temp->name,temp->designations,temp->salary,temp->phone\_number,temp->age );

}

temp=temp->next;

}

}

