TASK 1

#include<stdio.h>

#include<stdlib.h>

typedef struct info {

int id;

char name[25];

char Dob[50];

char email[100];

struct info \*next;

}info;

info \*head = NULL;

info\* createNode() {

info \*newNode = (info\*)malloc(sizeof(info));

printf("Enter id:");

scanf("%d",&newNode->id);

getchar();

printf("Enter name: ");

scanf("%[^\n]",newNode->name);

getchar();

printf("Enter Dob: ");

scanf("%[^\n]",newNode->Dob);

getchar();

printf("Enter email:");

scanf("%[^\n]",newNode->email);

newNode->next = NULL;

return newNode;

}

void addNode() {

info \*newNode = createNode();

info \*tmp = head;

if(head == NULL) {

head = newNode;

}else {

while(tmp->next != NULL)

tmp = tmp->next;

tmp->next = newNode;

}

}

void printList() {

info \*tmp = head;

while(tmp != NULL) {

printf("id: |%d|\n",tmp->id);

printf("name: |%s|\n",tmp->name);

printf("Dob:- |%s| \n",tmp->Dob);

printf("email: |%s|\n",tmp->email);

printf("----------------\n");

tmp = tmp->next;

}

printf("\n");

}

int mystrlen(char \*str) {

int i=0,cnt=0;

while(\*str != '\0') {

cnt++;

str++;

}

return cnt;

}

int strcmp(char \*str1 ,char \*str2) {

if(mystrlen(str1) != mystrlen(str2))

return 0;

while(\*str1 != '\0') {

if(\*str1 != \*str2)

return 0;

str1++;

str2++;

}

return 1;

}

void searchNode(){

char name[30];

info \*tmp = head;

int flg = 0;

getchar();

printf("Enter name to search:");

scanf("%[^\n]",name);

while(tmp != NULL) {

if(strcmp(name,tmp->name)){

flg = 1;

break;

}

tmp = tmp->next;

}

printf("\n--------------------------------------\n");

if(flg){

printf("Id:%d\n",tmp->id);

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

printf("DOB:%s\n",tmp->Dob);

printf("Email:%s\n",tmp->email);

}else {

printf("Not found");

}

printf("\n--------------------------------------\n");

}

void main() {

int ch;

while(1) {

printf("1.add node\n");

printf("2.print list\n");

printf("3.search node\n");

printf("4.exit\n");

printf("\nSelect any option:");

scanf("%d",&ch);

switch(ch) {

case 1:

addNode();

break;

case 2:

printList();

break;

case 3:

searchNode();

break;

case 4:

exit(0);

break;

default:

printf("Enter correct choice\n");

}

}

}

OUTPUT

