

ST THOMAS COLLEGE , THRISSUR
Department of Computer Science
BSc Computer Science (2020-23) - Semester III

Assignment 2

Question The source code for a simple contact book program is given below. You are asked to go through the code and its documentation. It lacks certain features and you are expected to improve the functionalities listed below.

1. In the original , you can only store the contact's name and number. Add option to store contact's email and birthday.
2. The contacts are stored one after another. It would be better if they are added(inserted) alphabetically. (*HINT*: strcmp())
3. The searching of a contact by name requires typing the exact name. It would be better if it could search with only a part of the name. (*HINT*: naive search)
4. Add an option to search a contact by the contact's number.
5. The contact details are lost when the program exits. It would be good if the data can be stored in a file.

Source Code

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

#define SIZE_OF_CONTACT_LIST 100

// *****
struct CONTACT {

    long int number;
    char name[50];
};

typedef struct CONTACT CONTACT;

CONTACT List[SIZE_OF_CONTACT_LIST];
int last_pos=0;

void add_contact(CONTACT c){
    List[last_pos]=c;
    last_pos++;
}

void delete_contact(int contact_index){

    CONTACT temp = List[contact_index];

    for(int i=contact_index; i<last_pos; i++)
        List[i]=List[i+1];

    last_pos--;
}
```

```

void edit_contact(CONTACT edited_contact,int index){

    List[index]=edited_contact;

}

void display_contact(CONTACT C){

    printf("\n-----\n");
    printf("\tName      : %s\n",C.name);
    printf("\tPhone Number : %li\n",C.number);
    printf("-----\n");

}

int search_contact_by_name(char name[]){

    for(int i=0;i<last_pos;i++){
        if( strcmp(name,List[i].name) == 0)
            return i;
    }

    return -1;
}

void accept_contact_details(CONTACT *c){
    printf("\nEnter name : ");
    scanf("%s", (c->name));
    printf("Enter phone number : ");
    scanf("%li",&(c->number));
}

//*****

void display_contacts(CONTACT list[],int length){
    printf("\nContact List\n\n");
    for(int i=0;i<length;i++){
        printf("%d )",i+1);
        display_contact(list[i]);
        printf("\n");
    }
}

//*****

void menu_display(){

    printf("\n1. Add Contact");
    printf("\n2. Delete Contact");
    printf("\n3. Edit Contact");
    printf("\n4. Display Contacts");
    printf("\n5. Search Contact By Name");
}

```

```

    printf("\n0. Exit\n");
    printf("Enter option : ");

}

void menu(){
    int choice=1;

    while(choice){

        menu_display();
        scanf("%d",&choice);

        switch(choice){

            case 1: {
                CONTACT C;
                accept_contact_details(&C);
                add_contact(C);
            }
            break;

            case 2: {
                int index;
                printf("Enter Index : ");
                scanf("%d",&index);
                delete_contact(index-1);
            }
            break;

            case 3: {
                int index;

                printf("Enter Index : ");
                scanf("%d",&index);
                CONTACT C;
                accept_contact_details(&C);

                // display original contact
                printf("\nOriginal Contact : ");
                display_contact(List[index-1]);

                // display edited contact
                printf("\nModified Contact : ");
                display_contact(C);

                char c;
                printf("Save edited contact?(Y/N)? ");
                scanf(" %c",&c);
                if(c=='y' || c=='Y')
                    edit_contact(C,index-1);
            }
            break;

```

```

        case 4: display_contacts(List,last_pos);
                break;

        case 5: {
                char name[50];
                printf("Enter Name : ");
                scanf("%s",name);

                int index=search_contact_by_name(name);
                if(index == -1){
                        printf("Not Found");
                        break;
                }
                printf("%d ",index+1);
                display_contact(List[index]);

                break;
        }
}

int main(void) {

    menu();

    return 0;
}

```

Documentation

- The contact details is stored in a structure name **CONTACT**.
- The contacts list is stored as a global array named **List**.
- New contacts are appended to the contact lists and **last_pos** stores the index of the last contact stored(i.e. length of list)
- **add_contact()** is the function that appends the contact to the contacts list and updates **last_pos**
 - argument : a **CONTACT** object that is to be added
- **delete_contact()** deletes a contact at the index passed to it , shifts the contacts to the left and updates **last_pos**
 - argument : the index of the contact to be deleted
- **edit_contact()** replace a contact with an edited copy of itself.
 - arguments : an **CONTACT** object with the modified details , index of the original contact
- **accept_contact_details()** : accepts data about the contact from the user and stores it a contact
 - argument : address of the **CONTACT** object where the data from the user needs to be stored
- **display_contact()** : display the details of a contact with pretty formatting
 - argument : a **CONTACT** object
- **search_contact_by_name()** : searches the contacts list for an exact match (case sensitive).
 - argument : a string name that needs to searched
 - return value : a whole number,i.e., index of the matched contact or -1 meaning no matches