

```

/*
Name: Larry Nguyen
Lab #6
Date : 02/25/2020
Description: This program creates an phonebook that holds contact information.
*/

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

// Global variables
int Count = 0;
int DelCount = 0;

// Phonebook list structure
typedef struct PhoneBookList
{
    char *ContactFirstName;
    char *ContactLastName;
    char *ContactPhoneNumber;
} list;

// Phonebook delete function structure
typedef struct DeleteEntry
{
    char *ContactFirstName;
    char *ContactLastName;
} take;

// Structure pointers
list *PhoneMemory;
take *DeletePhoneMemory;

// Prototypes
void Add();
void Delete();
void Display();

int main(void)
{
    int PhoneBookSelection;
    do { // Main menu display
        printf("\n\nPhone Book:\n\n");
        printf("1) Add friend\n");
        printf("2) Delete friend\n");
        printf("3) Display phone book\n");
        printf("4) Exit\n");
        printf("What do you want to do: ");
        scanf("%d", &PhoneBookSelection);

        // Setting up Switch
        switch (PhoneBookSelection)
        {
            case 1: // Add a contact
                Add();
                break;
            case 2: // Delete a contact
                Delete();
                break;
            case 3: // Display phonebook list
                Display();
                break;
            case 4: // Break loop and ends the program
                break;

            defauPhoneMemory: // Invalid number selection
                printf("\nInvalid selection. Try again.\n");
                break;
        } // End Switch
    } while (PhoneBookSelection != 4); //End Do While loop

    // Freeing up Memory
    free(DeletePhoneMemory);
    free(PhoneMemory);
    DeletePhoneMemory = NULL;
    PhoneMemory = NULL;
    return 0;
}

//Add an entry
void Add()
{
    if (Count == 0)
    {
        PhoneMemory = (list *) malloc ((Count*25) + 25);
    }
}

```

```

else
{
    PhoneMemory = (list *) realloc (PhoneMemory, (Count*50) + 50);
}
if (PhoneMemory == NULL)
{
    printf("Error, no more memory\n");
}
else
{
    // Memory allocation
    PhoneMemory[Count].ContactFirstName = (char *) malloc(sizeof(char)*15);
    PhoneMemory[Count].ContactLastName = (char *) malloc(sizeof(char)*15);
    PhoneMemory[Count].ContactPhoneNumber = (char *) malloc(sizeof(char)*15);
    // Input contact info
    printf("\nEnter their First Name: ");
    scanf("%s", PhoneMemory[Count].ContactFirstName);
    printf("\nEnter their Last Name: ");
    scanf("%s", PhoneMemory[Count].ContactLastName);
    printf("\nEnter their Phone Number: ");
    scanf("%s", PhoneMemory[Count].ContactPhoneNumber);
    printf("\nContact added\n");
}
Count++;
}
//Delete an entry
void Delete()
{
    int i;
    int q = 0;
    char *userName;
    // Memory allocation
    if (DelCount == 0)
    {
        DeletePhoneMemory = (take *) malloc ((DelCount*25) + 25);
    }
    else
    {
        DeletePhoneMemory = (take *) realloc (DeletePhoneMemory, (DelCount*1) + 1);
    }
    if (DeletePhoneMemory == NULL)
    {
        printf("This cannot be deleted (out of memory)\n");
    }
    else
    {
        DeletePhoneMemory[DelCount].ContactFirstName = (char *) malloc(sizeof(char)*15);
        DeletePhoneMemory[DelCount].ContactLastName = (char *) malloc(sizeof(char)*15);
    // User input for deleting contact
        printf("\nEnter their First Name: ");
        scanf("%s", DeletePhoneMemory[DelCount].ContactFirstName);
        printf("\nEnter their Last Name: ");
        scanf("%s", DeletePhoneMemory[DelCount].ContactLastName);
    }
    for (i = 0; i < Count; i++)
    {
        if (PhoneMemory[i].ContactFirstName == NULL && PhoneMemory[i].ContactLastName == NULL) continue;
        if (strcmp(PhoneMemory[i].ContactFirstName, DeletePhoneMemory[DelCount].ContactFirstName) == 0 && strcmp(PhoneMemory[i].ContactLastName, DeletePhoneMemory[DelCount].ContactLastName) == 0)
        {
            printf("\n%s %s has been deleted\n", PhoneMemory[i].ContactFirstName, PhoneMemory[i].ContactLastName);
            PhoneMemory[i].ContactFirstName = NULL;
            PhoneMemory[i].ContactLastName = NULL;
            PhoneMemory[i].ContactPhoneNumber = NULL;
            q = 1;
            break;
        }
    } // End for loop

    if (q != 1)
    {
        printf("\nThat contact does not exist\n");
    }
    DelCount++;
    Count--;
}
//Display all phonebook entries
void Display()
{
    int i;
    printf("\nYour contacts:\n");
    for (i = 0; i < Count; i++)
    {
        if (PhoneMemory[i].ContactFirstName != NULL && PhoneMemory[i].ContactLastName != NULL)
        {
            printf("\n%s %s: %s\n", PhoneMemory[i].ContactFirstName, PhoneMemory[i].ContactLastName, PhoneMemory[i].ContactPhoneNumber);
        }
    }
}

```

```
 } // End for loop  
system("pause");  
}
```