EEL PROJECT

ASSIGNMENT 4

Group members – Nidhi Yadav, Shruti Deshmukh, Sachin Lokhande, Bhavna

Research

We had created a contact management system in which we are comparing the contact names, copying contact information, calculate length of contact name, concatenate contact name with other details we had used different functions to implement operations on strings to build this program

We had used these websites for references

1. Strings in C - GeeksforGeeks
2. C String Functions - GeeksforGeeks

String functions are the built-in functions declared in <string.h> header file that can be used to perform various operations on the strings such as string copying, concatenation, comparison, searching, and more.

Analyse

While creating this programme we were getting compile time and run time error continuously, after discussing among the group members we researched on the problems and found a website called geeks for geeks, it is basically a problem-solving platform where we found the solutions for our problems, thus our errors were rectified.

As we had already mentioned the links of websites referred above for your reference.

Work done by the group members:

Sachin built the programme

Shruti helped with the code, word file, gathering information built the programme

Nidhi made the word file and helped withe code

Bhavna has gathered the information regarding string operators and basic syntax, also helped to rectify the errors (i.e. syntax and compile error) while programming

Ideate

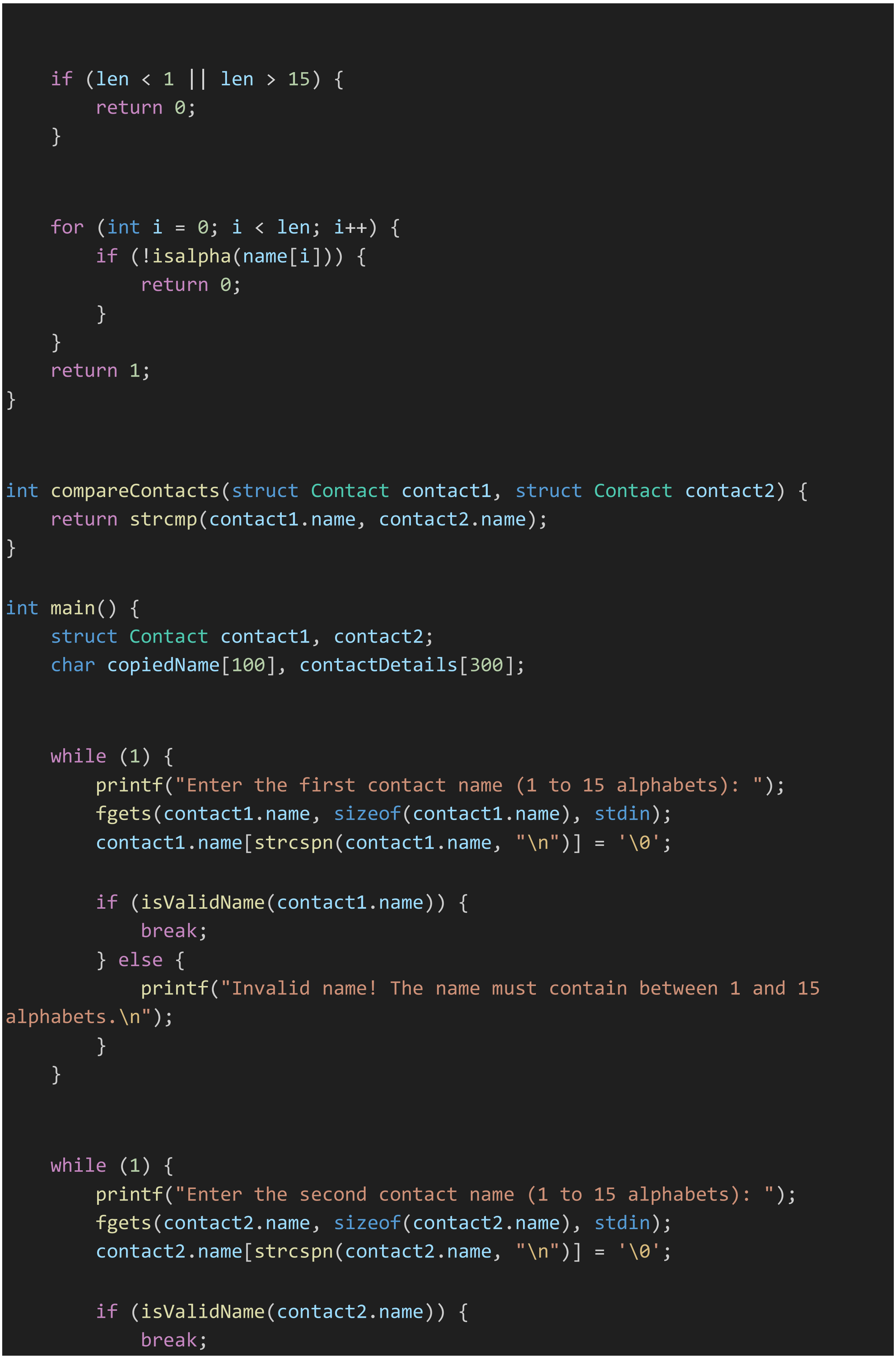
In a **Contact Management System**, you store and manage contact information like **names**, **email addresses**, and **phone numbers**. The system can perform operations on these pieces of data, such as comparing names, copying contact details, calculating the length of contact information, and concatenating the name with other details for display purposes.

We did the following modifications in the code:

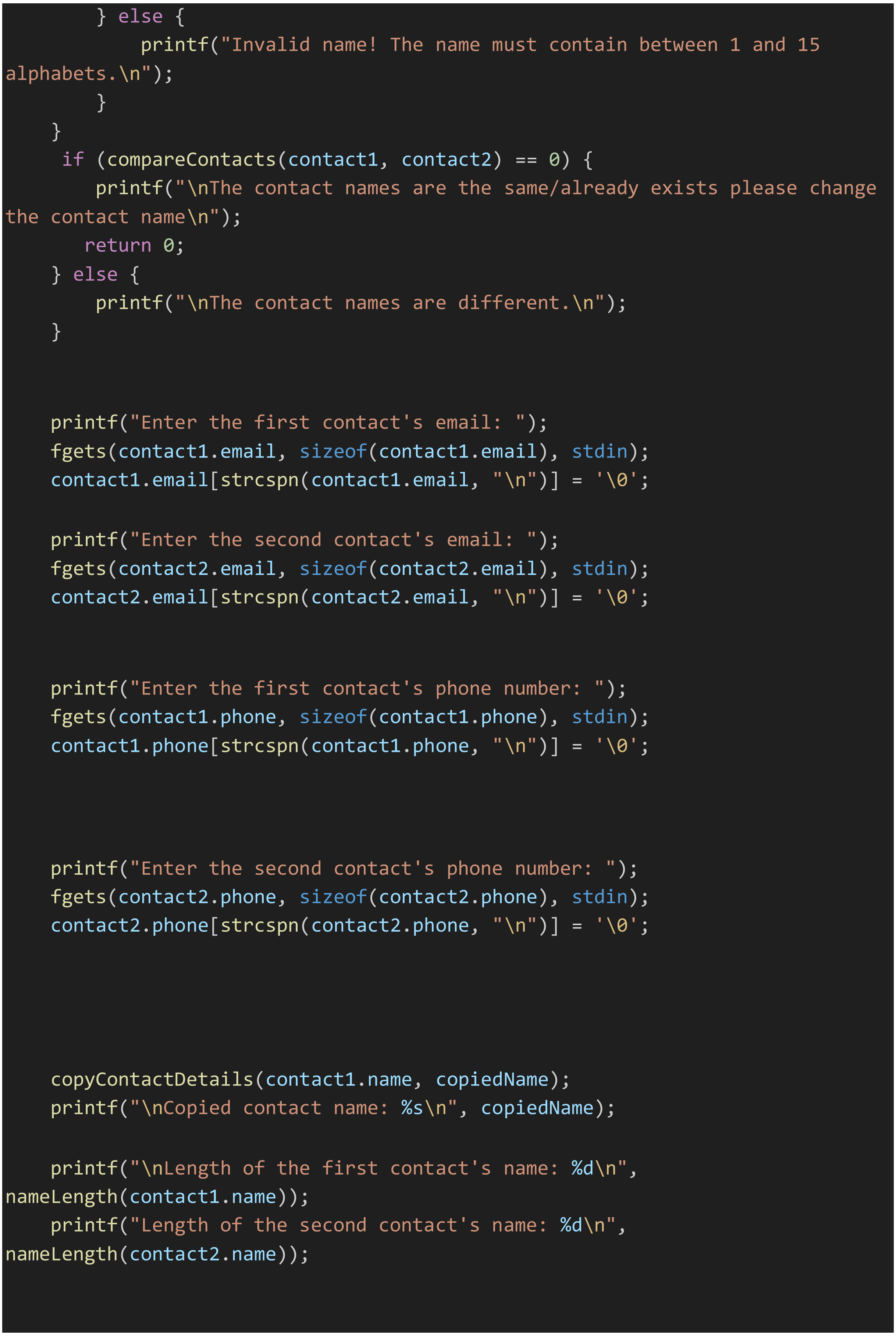
1. **Compare Contact Names**: Check if two contacts have the same name.
2. **Copy Contact Information**: Copy one contact's name or full details to a new contact.
3. **Calculate Length of Contact Name**: Measure how long the contact's name is.
4. **Concatenate Contact Name with Other Details**: Combine the contact's name, email, and phone number into a single string for display.
5. We used fgets and fput operation to retrieve the data from the file and print it

Build

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


struct Contact {
    char name[100];
    char email[100];
    char phone[20];
};


int compareNames(char name1[], char name2[]) {
    return strcmp(name1, name2);
}


void copyContactDetails(char src[], char dest[]) {
    strcpy(dest, src);
}


int nameLength(char name[]) {
    return strlen(name);
}

void concatenateContactInfo(char name[], char email[], char phone[], char result[]) {
    strcpy(result, name);
    strcat(result, " | ");
    strcat(result, email);
    strcat(result, " | ");
    strcat(result, phone);
}


int isValidName(char name[]) {
    int len = strlen(name);
```



```
    if (len < 1 || len > 15) {
        return 0;
    }


    for (int i = 0; i < len; i++) {
        if (!isalpha(name[i])) {
            return 0;
        }
    }
    return 1;
}


int compareContacts(struct Contact contact1, struct Contact contact2) {
    return strcmp(contact1.name, contact2.name);
}

int main() {
    struct Contact contact1, contact2;
    char copiedName[100], contactDetails[300];


    while (1) {
        printf("Enter the first contact name (1 to 15 alphabets): ");
        fgets(contact1.name, sizeof(contact1.name), stdin);
        contact1.name[strcspn(contact1.name, "\n")] = '\0';

        if (isValidName(contact1.name)) {
            break;
        } else {
            printf("Invalid name! The name must contain between 1 and 15
alphabets.\n");
        }
    }


    while (1) {
        printf("Enter the second contact name (1 to 15 alphabets): ");
        fgets(contact2.name, sizeof(contact2.name), stdin);
        contact2.name[strcspn(contact2.name, "\n")] = '\0';

        if (isValidName(contact2.name)) {
            break;
```

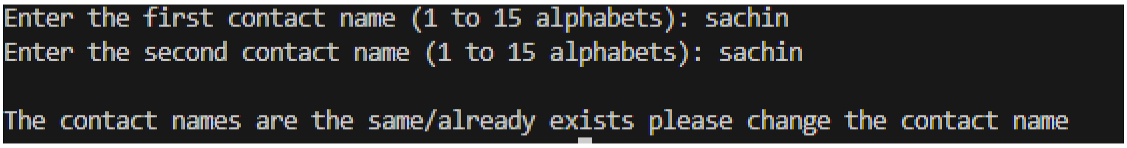


```
        } else {
            printf("Invalid name! The name must contain between 1 and 15
alphabets.\n");
        }
    }
     if (compareContacts(contact1, contact2) == 0) {
        printf("\nThe contact names are the same/already exists please change
the contact name\n");
       return 0;
    } else {
        printf("\nThe contact names are different.\n");
    }


    printf("Enter the first contact's email: ");
    fgets(contact1.email, sizeof(contact1.email), stdin);
    contact1.email[strcspn(contact1.email, "\n")] = '\0';

    printf("Enter the second contact's email: ");
    fgets(contact2.email, sizeof(contact2.email), stdin);
    contact2.email[strcspn(contact2.email, "\n")] = '\0';


    printf("Enter the first contact's phone number: ");
    fgets(contact1.phone, sizeof(contact1.phone), stdin);
    contact1.phone[strcspn(contact1.phone, "\n")] = '\0';



    printf("Enter the second contact's phone number: ");
    fgets(contact2.phone, sizeof(contact2.phone), stdin);
    contact2.phone[strcspn(contact2.phone, "\n")] = '\0';




    copyContactDetails(contact1.name, copiedName);
    printf("\nCopied contact name: %s\n", copiedName);

    printf("\nLength of the first contact's name: %d\n",
nameLength(contact1.name));
    printf("Length of the second contact's name: %d\n",
nameLength(contact2.name));
```

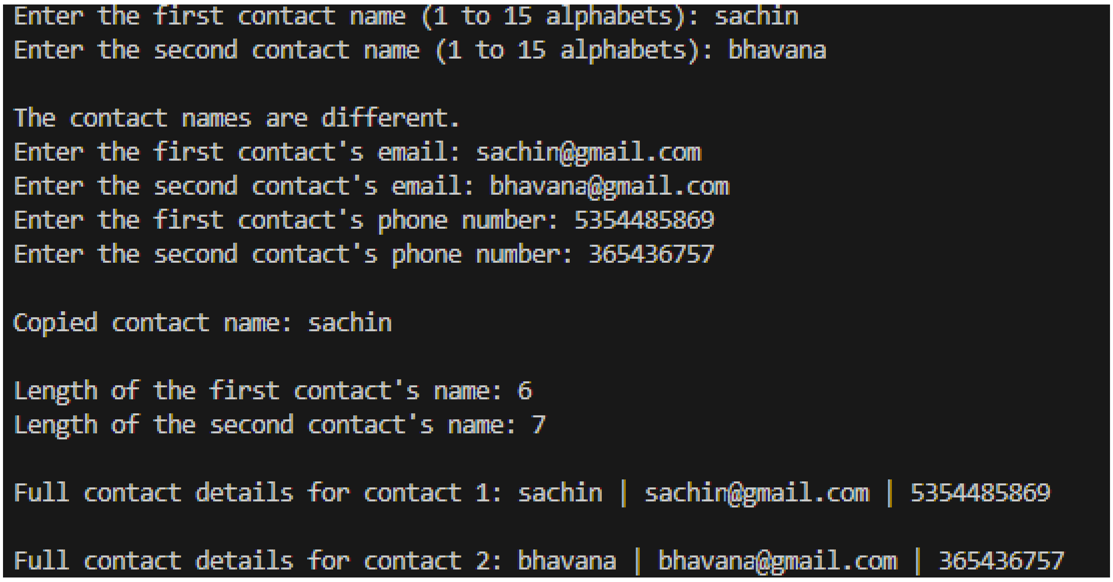
```
    concatenateContactInfo(contact1.name, contact1.email, contact1.phone, 
contactDetails);
    printf("\nFull contact details for contact 1: %s\n", contactDetails);

    concatenateContactInfo(contact2.name, contact2.email, contact2.phone, 
contactDetails);
    printf("\nFull contact details for contact 2: %s\n", contactDetails);

    return 0;
}
```

Test

SCREENSHOT OF OUR COMPLETE CODE WITH OUTPUT OF OUR CODE



```
Enter the first contact name (1 to 15 alphabets): sachin
Enter the second contact name (1 to 15 alphabets): sachin

The contact names are the same/already exists please change the contact name
```



```
 Enter the first contact name (1 to 15 alphabets): sachin
 Enter the second contact name (1 to 15 alphabets): bhavana

 The contact names are different.
 Enter the first contact's email: sachin@gmail.com
 Enter the second contact's email: bhavana@gmail.com
 Enter the first contact's phone number: 5354485869
 Enter the second contact's phone number: 365436757

 Copied contact name: sachin

 Length of the first contact's name: 6
 Length of the second contact's name: 7

 Full contact details for contact 1: sachin | sachin@gmail.com | 5354485869

 Full contact details for contact 2: bhavana | bhavana@gmail.com | 365436757
```

Implement

Now we can say that output is implemented and our code is also running successfully.

We can implement the project for the calculation of expense of students.

We know that GitHub is used by many people and in corporate sector. So publishing on GitHub will help us to reach out to a large number of consumers.

NidhiYadav-411/contact-management-system