Name: Subhranil Nandy
Group: HX
RollNo: 2022ITB012

📋 Assignment 5

🎯 1. Write a Program to find the length of a given string, including and excluding the spaces, using the pointers. Could you print the output as follows ?

```
Length of string including spaces: xxx
Length of string excluding spaces: yyy
```

language-C

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    char *p = calloc(100, 1);
    int len, lenExcl;
    len=0;
    lenExcl=0;
    printf("Enter the string: ");
    scanf("%[^\n]", p);
    printf("Received string = '%s'\n", p);

    while(*p != '\0')
    {
        if(*p != ' ')
            lenExcl++;
        len++;
        p++;
    }
    printf("Length of string including spaces: %d\n", len);
    printf("Length of string excluding spaces: %d\n", lenExcl);
    return 0;
}
```

OUTPUT

```
student@student-HP-ProDesk-600-G5-MT: ~/2022ITB012/Assignment5
File  Edit  View  Search  Terminal  Help
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$ gcc 12_a5_1.c
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$ ./a.out
Enter the string:    All is   well!
Received string = '   All is   well!  '
Length of string including spaces: 19
Length of string excluding spaces: 10
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$
```

🎯 2. Write a program to read two strings through the keyboard. Compare these two strings character by character. Display the similar characters found in both strings and count the number of dissimilar characters. Print the output as follows.

```
Similar characters found in both strings are as follows.
< characters >
The strings are different at ___ places.
The string characters are similar at ___ places.
```

language-C

```c
#include<stdio.h>
int len(char *str)
{
    int len=0;
    while(*str != '\0')
    {
        len++;
        str++;
    }
    return len;
}
int main()
{
    char str1[100];
    char str2[100];
    int same=0, diff=0;
    printf("Enter String 1: ");
    scanf("%[^\n]", str1);
    getchar();
    printf("Enter String 2: ");
    scanf("%[^\n]", str2);

    printf("Received strings:\n");
    printf("'%s' and '%s'\n", str1, str2);
    printf("Similar characters found in both strings are as follows:\n");
    int min = len(str1)<len(str2)?len(str1):len(str2);
    int max = len(str1)>len(str2)?len(str1):len(str2);
    int i = 0;
    for(i=0;i<min;i++)
    {
        if(str1[i] == str2[i])
        {
            printf("%c ", str1[i]);
            same++;
        }
    }
    diff=max-same;
    printf("\nThe strings are different at %d places.\n", diff);
    printf("The strings are similar at %d places.\n", same);
    return 0;
}
```

OUTPUT

```
student@student-HP-ProDesk-600-G5-MT: ~/2022ITB012/Assignment5
File  Edit  View  Search  Terminal  Help
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$ gcc 12_a5_2.c
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$ ./a.out
Enter String 1: Football
Enter String 2: foot
Received strings:
'Football' and 'foot'
Similar characters found in both strings are as follows:
o o t
The strings are different at 5 places.
The strings are similar at 3 places.
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$
```

3. Write a program to enter three characters using pointers. Use the `memcmp()` function [compares a specified number of characters from two buffers] for comparing the three characters. In case the entered characters are the same, display the message "the characters are same" otherwise, indicate their appearance before or after one another or display the status of characters in alphabetic order.

language-C

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

int main()
{
    char c1, c2, c3;
    printf("Enter 3 characters :");
    scanf("%c %c %c", &c1, &c2, &c3);
    printf("Characters entered: %c %c %c\n", c1, c2, c3);
    printf("Appearance order  : 1 2 3\n");
    if(memcmp(&c1, &c2, 1) == 0 && memcmp(&c2, &c3, 1) == 0)
        printf("The characters are same");
    else
    {
        printf("The status of characters in alphabetic order with their appearance ordering:\n");
        if(memcmp(&c1, &c2, 1)>0 && memcmp(&c1, &c3, 1)>0)
        {
            if(memcmp(&c2, &c3, 1)>0)
                printf("%c %c %c\n1 2 3", c1, c2, c3);
            else
                printf("%c %c %c\n1 3 2", c1, c3, c2);
        }
        else if(memcmp(&c2, &c1, 1)>0 && memcmp(&c2, &c3, 1)>0)
        {
            if(memcmp(&c1, &c3, 1)>0)
                printf("%c %c %c\n2 1 3", c2, c1, c3);
            else
                printf("%c %c %c\n2 3 1", c2, c3, c1);
        }
        else
        {
            if(memcmp(&c1, &c2, 1)>0)
                printf("%c %c %c\n3 1 2", c3, c1, c2);
            else
                printf("%c %c %c\n3 2 1", c3, c2, c1);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT

```
student@student-HP-ProDesk-600-G5-MT: ~/2022ITB012/Assignment5
File  Edit  View  Search  Terminal  Help
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$ gcc 12_a5_3.c
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$ ./a.out
Enter 3 characters :c a b
Characters entered: c a b
Appearance order  : 1 2 3
The status of characters in alphabetic order with their appearance ordering:
c b a
1 3 2
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$
```

---

4. Write a program to read a string. Could you print the string up to the first occurrence of the character entered through the keyboard?

language-C

```c
#include<stdio.h>
int main()
{
    char str[100];
    char c;
    char out[100];
    int i=0;
    int flag=0;
    printf("Enter the string : ");
    scanf("%[^\n]", str);
    scanf("%*c");
    printf("Enter the character: ");
    scanf("%c", &c);
    while(str[i] != '\0')
    {
        if(str[i] == c)
        {
            out[i] = str[i];
            flag=1;
            i++;
            break;
        }
        out[i] = str[i];
        i++;
    }
    out[i]='\0';
    if(flag == 1)
        printf("String upto first occurence of %c: %s\n", c, out);
    else
        printf("Character %c is not present in string.\n", c);
    return 0;
}
```

OUTPUT

```
student@student-HP-ProDesk-600-G5-MT: ~/2022ITB012/Assignment5
File  Edit  View  Search  Terminal  Help
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$ gcc 12_a5_4.c
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$ ./a.out
Enter the string : String occurence
Enter the character: c
String upto first occurence of c: String oc
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$
```

---

5. Write a program that stores lists of names (the last name first) and ages in parallel arrays and sorts the names into alphabetical order keeping the ages with the correct names. Sample output is as follows.

```
Original list
----------------------
Tendulkar, Sachin 42
Ganguly, Sourav 44
Richards, Vivian 58
```

```
Alphabetized list
-----------------
Ganguly, Sourav 44
Richards, Vivian 58
Tendulkar, Sachin 42
```

language-C

```c
#include<stdio.h>
#include<string.h>
#include<ctype.h>
void sortNames(char names[][50], int *ages, int n);
int compare(char *name1, char *name2);
void display(char names[][50], int *ages, int n);
int main()
{
    int n;
    char name[50];

    printf("Enter the no. of names: ");
    scanf("%d", &n);
    char names[100][50];
    int ages[100];
    for(int i=0;i<n;i++)
    {
        printf("Name %d: ", i+1);
        scanf("%s %s", name, names[i]);// Storing the last name first and first name in another var.
        strcat(names[i], ", ");
        strcat(names[i], name);

        printf("Age: ");
        scanf("%d", &ages[i]);
    }
    printf("Original list\n");
    printf("-------------\n");
    display(names, ages, n);
    sortNames(names, ages, n);
    printf("\nAlphabetized list\n");
    printf("-----------------\n");
    display(names, ages, n);
    return 0;
}
void sortNames(char names[][50], int *ages, int n)
{
    // Insertion sort implementation
    int i, j;
    char nkey[50];
    int akey; // !Always use keys in this sort
    for(i=1;i<n;i++)
    {
        strcpy(nkey, names[i]);
        akey = ages[i];
        for(j=i-1;j>=0 && compare(nkey, names[j]) == -1;j--)
        {
                strcpy(names[j+1], names[j]);
                ages[j+1] = ages[j];
        }
        strcpy(names[j+1], nkey);
        ages[j+1] = akey;
    }
}
int compare(char *name1, char *name2)
{
    int i,j;
    if(name1 == name2)
        return 0;
    for(i=0, j=0;;i++, j++)
    {
        if(name1[i] == ',')
        {
            i+=2;
        }
        if(name2[j] == ',')
        {
            j+=2;
        }
        if(toupper(name1[i])< toupper(name2[j]))
            return -1;
        else if(toupper(name1[i])> toupper(name2[j]))
            return 1;
    }
    // return 0;
}
void display(char names[][50], int *ages, int n)
{
    for(int i=0;i<n;i++)
        printf("%s %d\n", names[i], ages[i]);
}
```

OUTPUT

```
student@student-HP-ProDesk-600-G5-MT: ~/2022ITB012/Assignment5
File  Edit  View  Search  Terminal  Help
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$ gcc 12_a5_5.c
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$ ./a.out
Enter the no. of names: 3
Name 1: Sachin Tendulkar
Age: 42
Name 2: Sourav Ganguly
Age: 44
Name 3: Vivian Richards
Age: 58
Original list
-------------
Tendulkar, Sachin 42
Ganguly, Sourav 44
Richards, Vivian 58

Alphabetized list
-----------------
Ganguly, Sourav 44
Richards, Vivian 58
Tendulkar, Sachin 42
student@student-HP-ProDesk-600-G5-MT:~/2022ITB012/Assignment5$
```