Assignment 6: String

Submitted By: U19CS012(D-12)

1. 1) Without using inbuilt string functions, perform the following in C language:

a) Copy one string into another

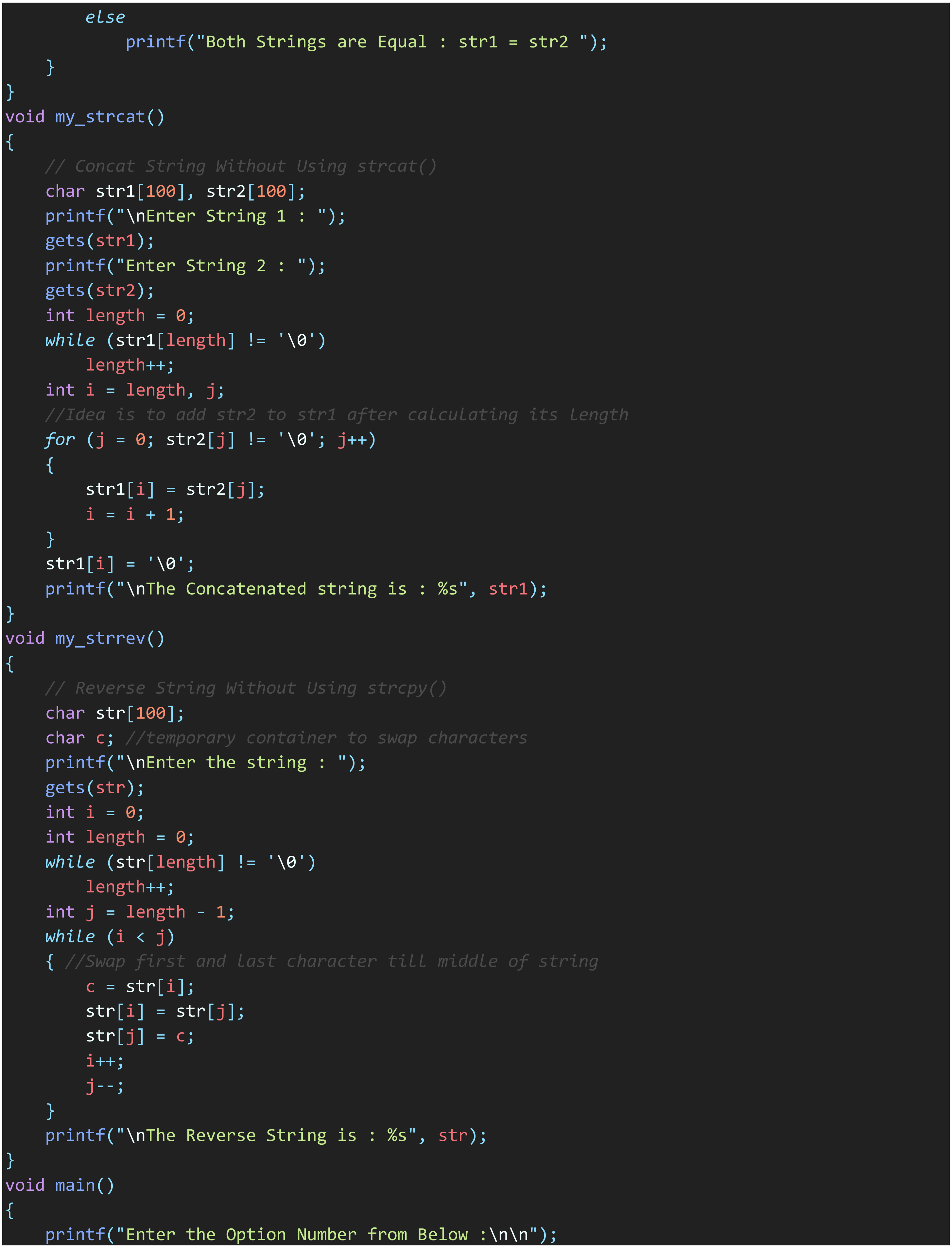
b) Compare two strings

c) Concat two strings

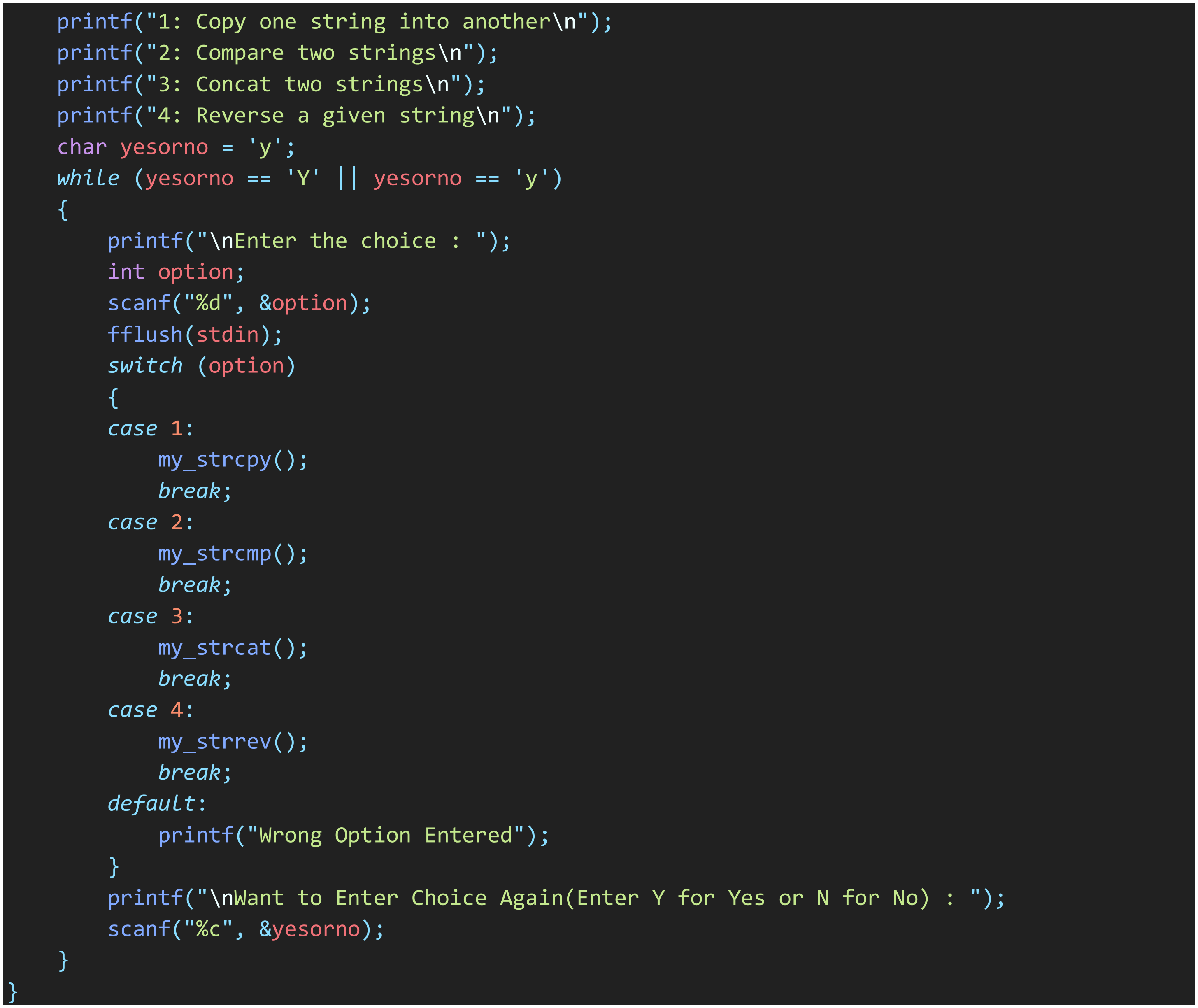
d) Reverse a given string

Code:

```
#include <stdio.h>

void my_strcpy()
{
    // Copy String Without Using strcpy()
    char str1[100], str2[100];
    printf("\nEnter String 1: ");
    gets(str1);
    int i;
    for (i = 0; str1[i] != '\0'; ++i)
    {
        str2[i] = str1[i];
    }
    str2[i] = '\0'; //Assigning last Null Charater to String 2
    printf("The Copied String 2 is : %s ", str2);
}
void my_strcmp()
{
    // Compare String Without Using strcmp()
    char str1[100], str2[100];
    printf("\nEnter String 1 : ");
    gets(str1);
    printf("Enter String 2 : ");
    gets(str2);
    int i = 0;
    // strcmp() -> 0(Equal) -> ">0"(str1 greater) -> "<0"(str2 greater)
    // Checking Character by Character till end of str 1 or str 2
    while (str1[i] == str2[i] && str1[i] != '\0' && str2[i] != '\0')
        i++;
    // Comparing the Different Letter(Lexicographically)
    if (str1[i] > str2[i])
        printf("String 1 is Lexicographically Greater : str1 > str2");
    else
    {
        if (str1[i] < str2[i])
            printf("String 2 is Lexicographically Greater : str1 < str2");
```



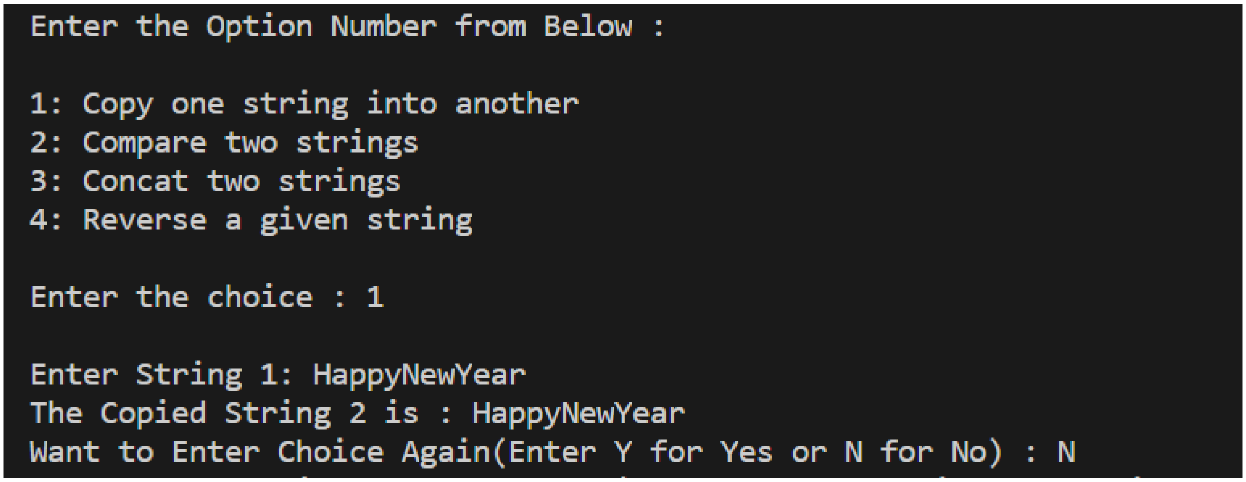
```
        else
            printf("Both Strings are Equal : str1 = str2 ");
    }
}
void my_strcat()
{
    // Concat String Without Using strcat()
    char str1[100], str2[100];
    printf("\nEnter String 1 : ");
    gets(str1);
    printf("Enter String 2 : ");
    gets(str2);
    int length = 0;
    while (str1[length] != '\0')
        length++;
    int i = length, j;
    //Idea is to add str2 to str1 after calculating its length
    for (j = 0; str2[j] != '\0'; j++)
    {
        str1[i] = str2[j];
        i = i + 1;
    }
    str1[i] = '\0';
    printf("\nThe Concatenated string is : %s", str1);
}
void my_strrev()
{
    // Reverse String Without Using strcpy()
    char str[100];
    char c; //temporary container to swap characters
    printf("\nEnter the string : ");
    gets(str);
    int i = 0;
    int length = 0;
    while (str[length] != '\0')
        length++;
    int j = length - 1;
    while (i < j)
    { //Swap first and last character till middle of string
        c = str[i];
        str[i] = str[j];
        str[j] = c;
        i++;
        j--;
    }
    printf("\nThe Reverse String is : %s", str);
}
void main()
{
    printf("Enter the Option Number from Below :\n\n");
```



```
    printf("1: Copy one string into another\n");
    printf("2: Compare two strings\n");
    printf("3: Concat two strings\n");
    printf("4: Reverse a given string\n");
    char yesorno = 'y';
    while (yesorno == 'Y' || yesorno == 'y')
    {
        printf("\nEnter the choice : ");
        int option;
        scanf("%d", &option);
        fflush(stdin);
        switch (option)
        {
        case 1:
            my_strcpy();
            break;
        case 2:
            my_strcmp();
            break;
        case 3:
            my_strcat();
            break;
        case 4:
            my_strrev();
            break;
        default:
            printf("Wrong Option Entered");
        }
        printf("\nWant to Enter Choice Again(Enter Y for Yes or N for No) : ");
        scanf("%c", &yesorno);
    }
}
```

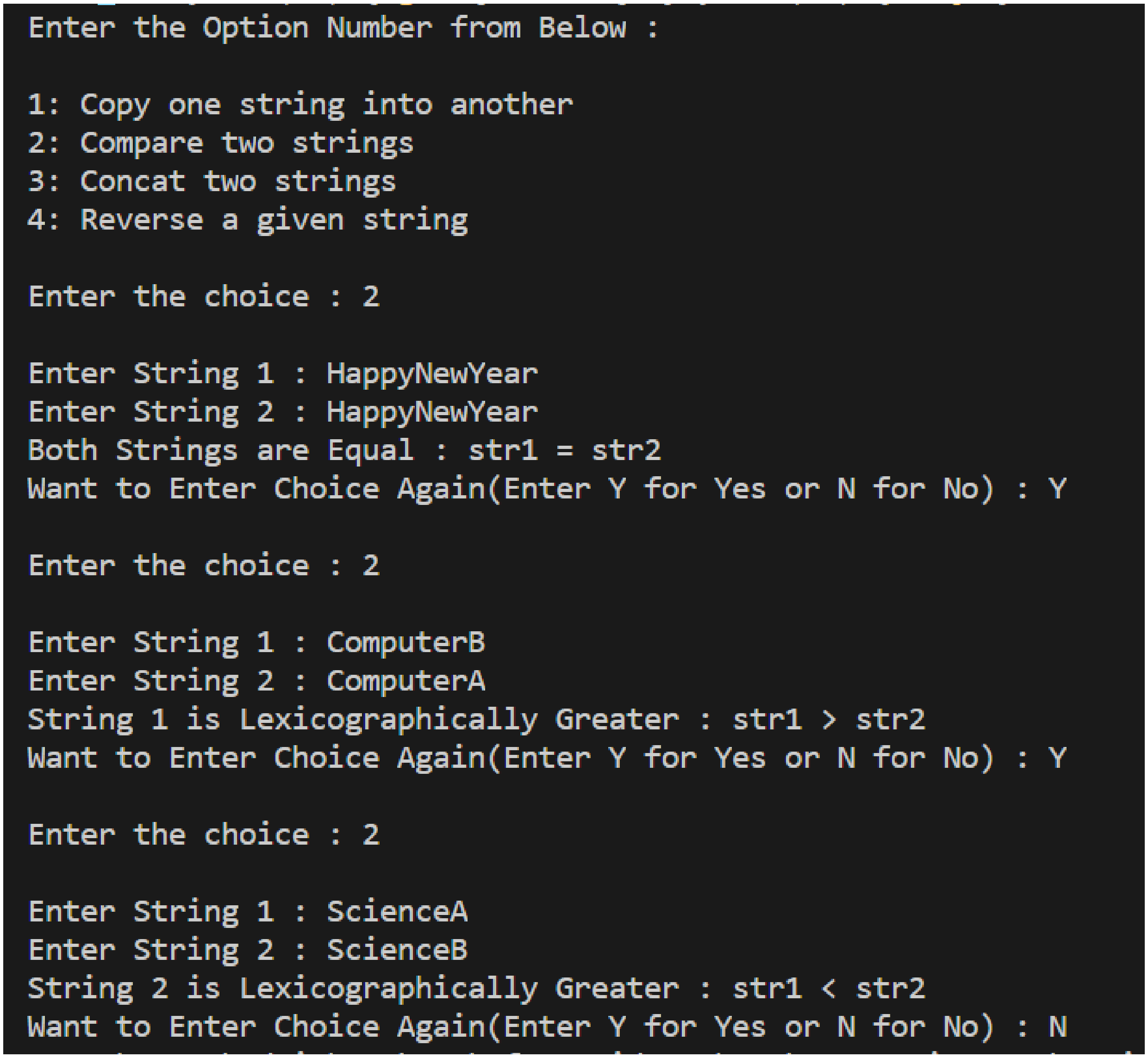
Output:

A.) Copy String



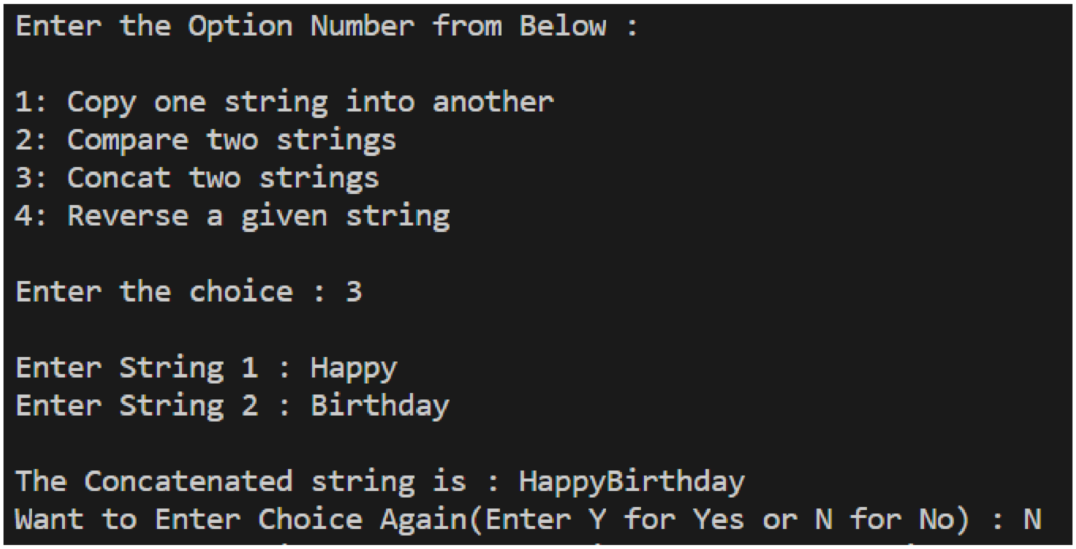
```
Enter the Option Number from Below :

1: Copy one string into another
2: Compare two strings
3: Concat two strings
4: Reverse a given string

Enter the choice : 1

Enter String 1: HappyNewYear
The Copied String 2 is : HappyNewYear
Want to Enter Choice Again(Enter Y for Yes or N for No) : N
```

B.) Compare Strings



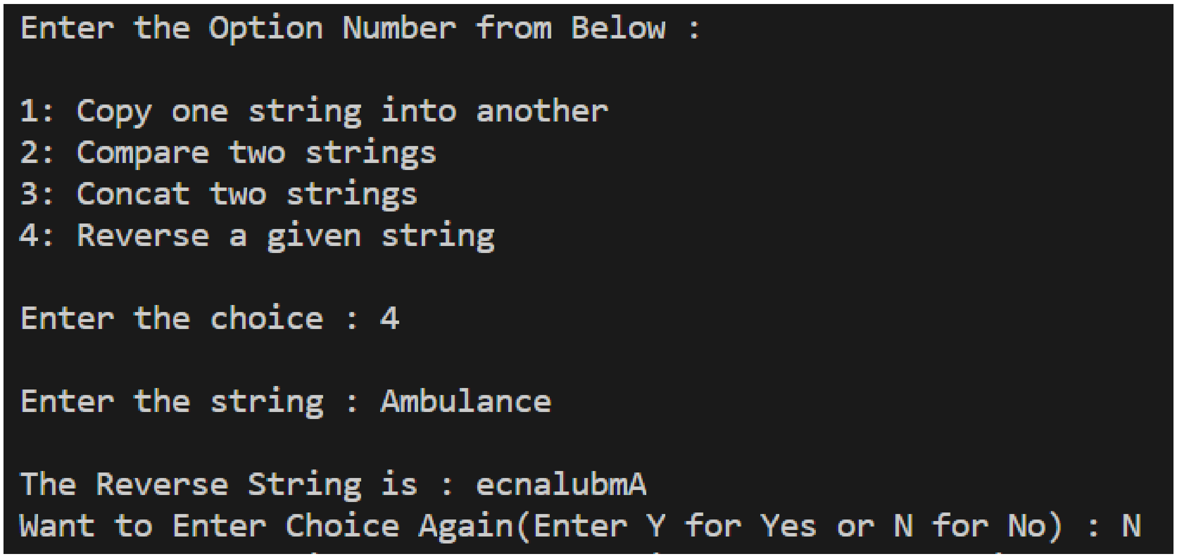
```
Enter the Option Number from Below :

1: Copy one string into another
2: Compare two strings
3: Concat two strings
4: Reverse a given string

Enter the choice : 2

Enter String 1 : HappyNewYear
Enter String 2 : HappyNewYear
Both Strings are Equal : str1 = str2
Want to Enter Choice Again(Enter Y for Yes or N for No) : Y

Enter the choice : 2

Enter String 1 : ComputerB
Enter String 2 : ComputerA
String 1 is Lexicographically Greater : str1 > str2
Want to Enter Choice Again(Enter Y for Yes or N for No) : Y

Enter the choice : 2

Enter String 1 : ScienceA
Enter String 2 : ScienceB
String 2 is Lexicographically Greater : str1 < str2
Want to Enter Choice Again(Enter Y for Yes or N for No) : N
```

C.) Concat Strings



```
Enter the Option Number from Below :

1: Copy one string into another
2: Compare two strings
3: Concat two strings
4: Reverse a given string

Enter the choice : 3

Enter String 1 : Happy
Enter String 2 : Birthday

The Concatenated string is : HappyBirthday
Want to Enter Choice Again(Enter Y for Yes or N for No) : N
```

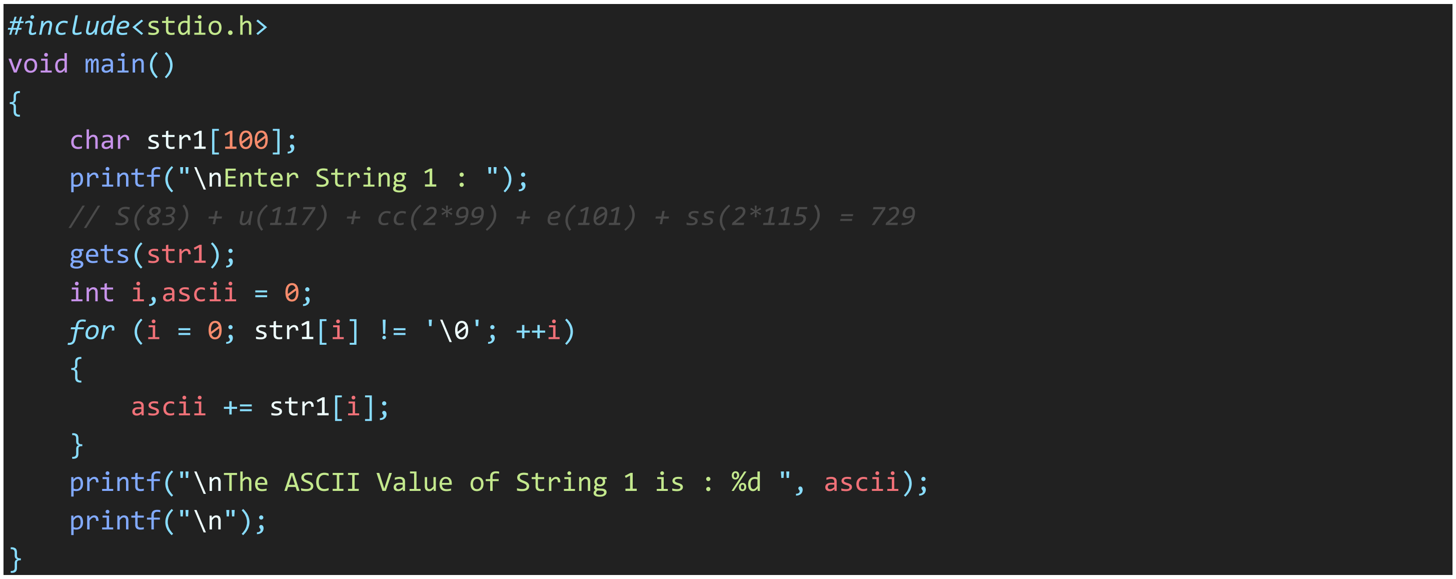
D.) Reverse String



```
Enter the Option Number from Below :

1: Copy one string into another
2: Compare two strings
3: Concat two strings
4: Reverse a given string

Enter the choice : 4

Enter the string : Ambulance

The Reverse String is : ecnalubmA
Want to Enter Choice Again(Enter Y for Yes or N for No) : N
```

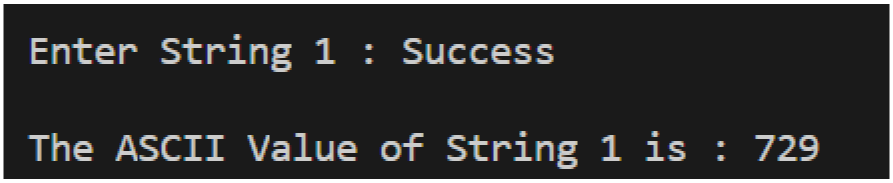
2. Write a program to print equivalent ASCII code of the string entered by user.

Code:



```c
#include<stdio.h>
void main()
{
    char str1[100];
    printf("\nEnter String 1 : ");
    // S(83) + u(117) + cc(2*99) + e(101) + ss(2*115) = 729
    gets(str1);
    int i,ascii = 0;
    for (i = 0; str1[i] != '\0'; ++i)
    {
        ascii += str1[i];
    }
    printf("\nThe ASCII Value of String 1 is : %d ", ascii);
    printf("\n");
}
```

Output:



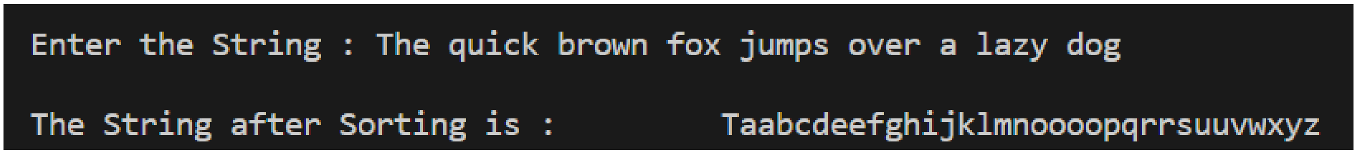
```
Enter String 1 : Success

The ASCII Value of String 1 is : 729
```

3. Write a program to sort a given string in alphabetical order.

Code:

```c
#include <stdio.h>
void main()
{
    char str[100];
    printf("\nEnter the String : ");
    gets(str);
    char c; //For Swapping
    int length = 0;
    while (str[length] != '\0')
        length++;
    int i, j, n = length;
    for (i = 0; i < n - 1; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (str[i] > str[j])
            {
                c = str[i];
                str[i] = str[j];
                str[j] = c;
            }
        }
    }
    printf("\nThe String after Sorting is : %s ", str);
}
```

Output:



```
Enter the String : The quick brown fox jumps over a lazy dog

The String after Sorting is :          Taabcdeefghijklmnoooopqrrsuuvwxyz
```

4. Perform the following in C language:

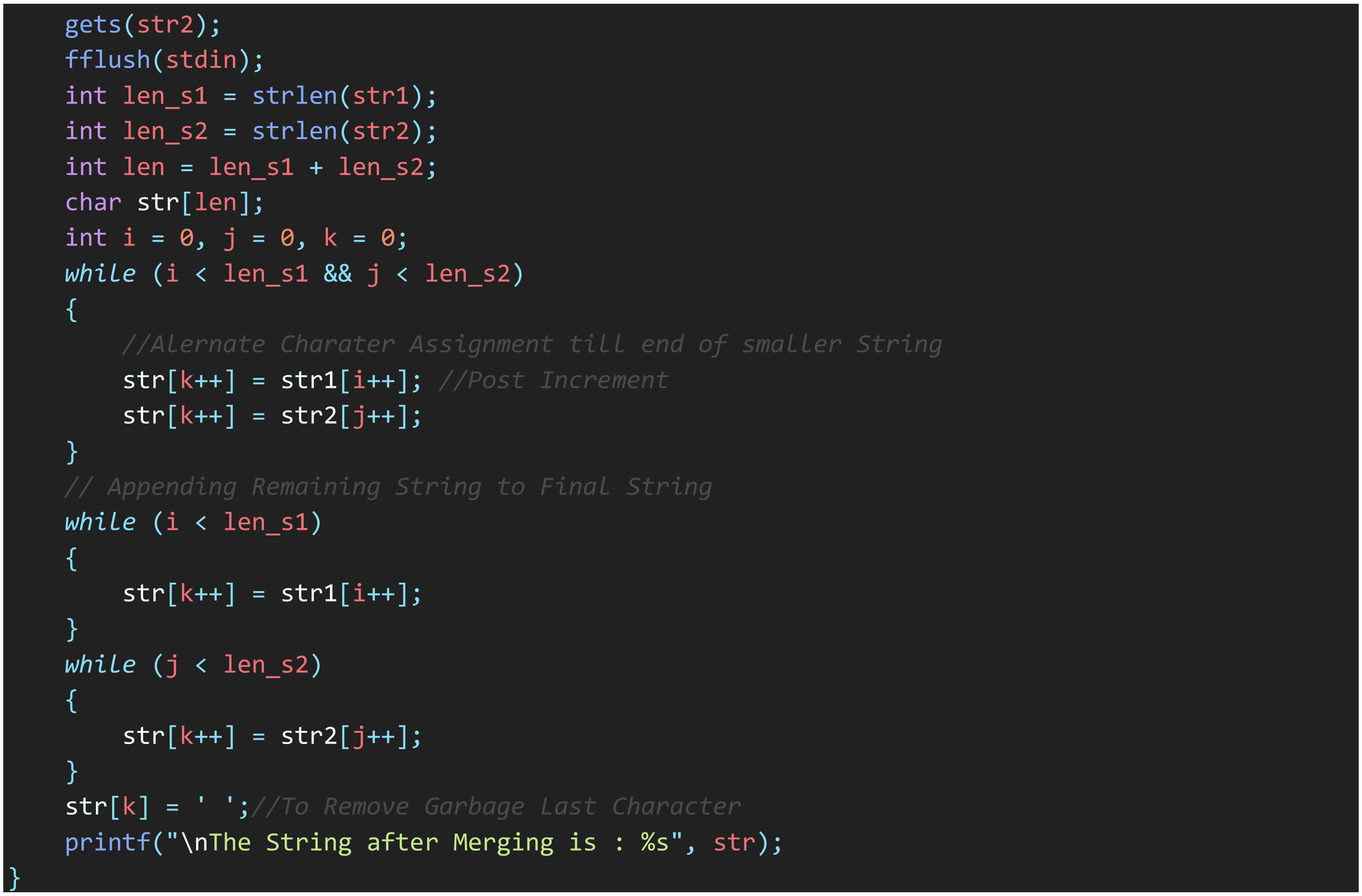
For two strings given by user, merge them and print the third string in the following manner:

Input: String1: FCP String2: subject

Output string: FsCuPbject

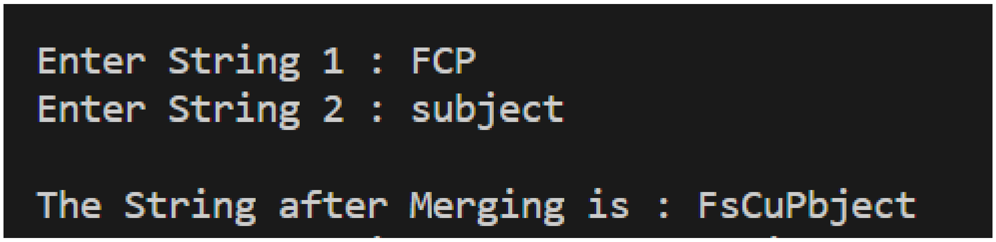
Code:

```c
#include <stdio.h>
#include <string.h>
void main()
{
    char str1[100], str2[100];
    printf("\nEnter String 1 : ");
    gets(str1);
    printf("Enter String 2 : ");
```



```c
    gets(str2);
    fflush(stdin);
    int len_s1 = strlen(str1);
    int len_s2 = strlen(str2);
    int len = len_s1 + len_s2;
    char str[len];
    int i = 0, j = 0, k = 0;
    while (i < len_s1 && j < len_s2)
    {
        //Alernate Charater Assignment till end of smaller String
        str[k++] = str1[i++]; //Post Increment
        str[k++] = str2[j++];
    }
    // Appending Remaining String to Final String
    while (i < len_s1)
    {
        str[k++] = str1[i++];
    }
    while (j < len_s2)
    {
        str[k++] = str2[j++];
    }
    str[k] = ' ';//To Remove Garbage Last Character
    printf("\nThe String after Merging is : %s", str);
}
```

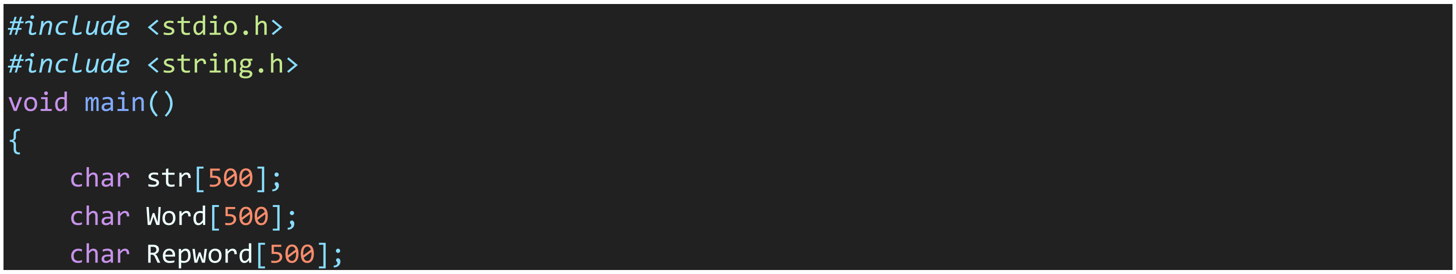
Output:



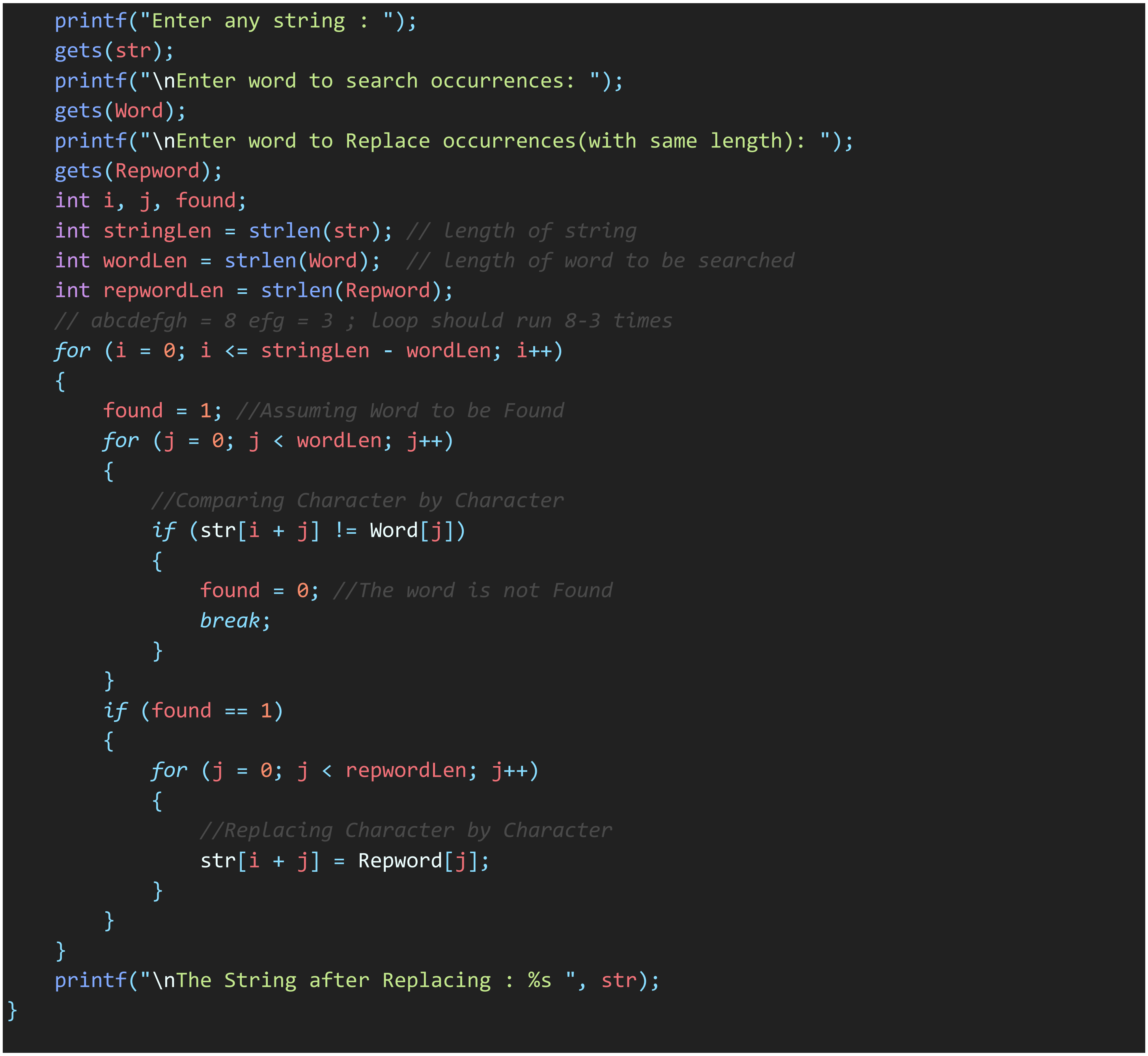
```
Enter String 1 : FCP
Enter String 2 : subject

The String after Merging is : FsCuPbject
```

5. Write a program in C to replace a particular word by another word from a given string.

Code: (Question 6 was Solved First and then This Question Became Easy)!

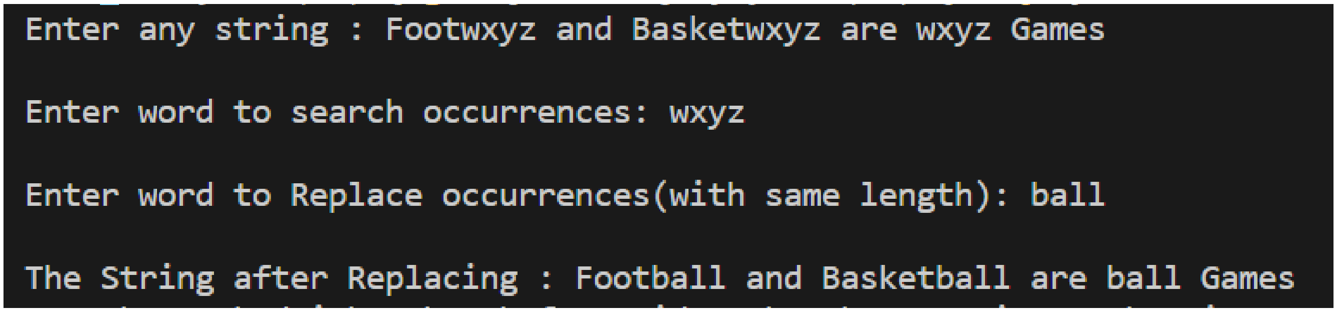


```c
#include <stdio.h>
#include <string.h>
void main()
{
    char str[500];
    char Word[500];
    char Repword[500];
```



```c
    printf("Enter any string : ");
    gets(str);
    printf("\nEnter word to search occurrences: ");
    gets(Word);
    printf("\nEnter word to Replace occurrences(with same length): ");
    gets(Repword);
    int i, j, found;
    int stringLen = strlen(str); // length of string
    int wordLen = strlen(Word);  // length of word to be searched
    int repwordLen = strlen(Repword);
    // abcdefgh = 8 efg = 3 ; loop should run 8-3 times
    for (i = 0; i <= stringLen - wordLen; i++)
    {
        found = 1; //Assuming Word to be Found
        for (j = 0; j < wordLen; j++)
        {
            //Comparing Character by Character
            if (str[i + j] != Word[j])
            {
                found = 0; //The word is not Found
                break;
            }
        }
        if (found == 1)
        {
            for (j = 0; j < repwordLen; j++)
            {
                //Replacing Character by Character
                str[i + j] = Repword[j];
            }
        }
    }
    printf("\nThe String after Replacing : %s ", str);
}
```

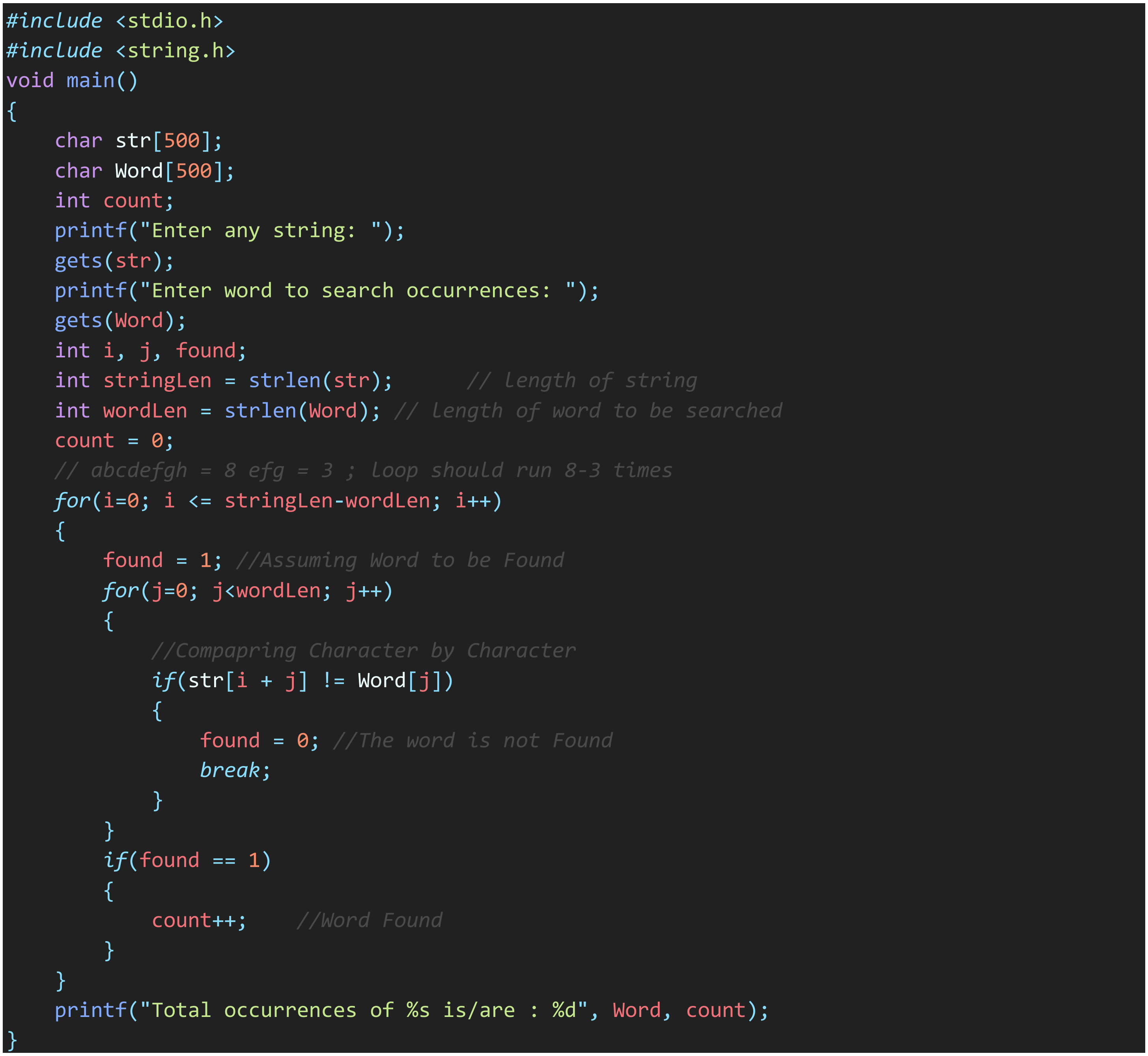
Output:



```
Enter any string : Footwxyz and Basketwxyz are wxyz Games

Enter word to search occurrences: wxyz

Enter word to Replace occurrences(with same length): ball

The String after Replacing : Football and Basketball are ball Games
```

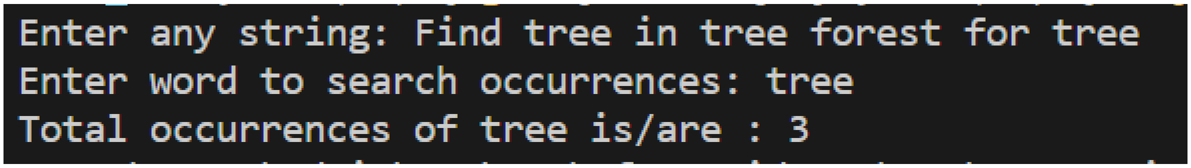
6. Write a program to count all occurrences of a particular word from a given string.

Code:



```
#include <stdio.h>
#include <string.h>
void main()
{
    char str[500];
    char Word[500];
    int count;
    printf("Enter any string: ");
    gets(str);
    printf("Enter word to search occurrences: ");
    gets(Word);
    int i, j, found;
    int stringLen = strlen(str);      // length of string
    int wordLen = strlen(Word); // length of word to be searched
    count = 0;
    // abcdefgh = 8 efg = 3 ; loop should run 8-3 times
    for(i=0; i <= stringLen-wordLen; i++)
    {
        found = 1; //Assuming Word to be Found
        for(j=0; j<wordLen; j++)
        {
            //Compapring Character by Character
            if(str[i + j] != Word[j])
            {
                found = 0; //The word is not Found
                break;
            }
        }
        if(found == 1)
        {
            count++;     //Word Found
        }
    }
    printf("Total occurrences of %s is/are : %d", Word, count);
}
```

Output:



```
Enter any string: Find tree in tree forest for tree
Enter word to search occurrences: tree
Total occurrences of tree is/are : 3
```

Submitted By:

Bhagya Rana

U19CS012 (D-12) (CSE, SVNIT)