



CENTRAL CALCUTTA POLYTECHNIC

21, Convent Road, Philips, Sealdah, Kolkata, West Bengal 700014

DEPT. : COMPUTER SCIENCE AND TECHNOLOGY

- NAME : ASIM BERA
- ROLL : DCCPCSTS3
- NO : 10005504
- SUBJECT : PROGRAMMING IN C
- SESSION : 2020 - 21

Contents

1	Strings in C	2
1.1	Write a C program to find length of a string.	2
1.2	Write a C program to copy one string to another string.	3
1.3	Write a C program to concatenate two strings.	4
1.4	Write a C program to compare two strings.	5
1.5	Write a C program to convert lowercase string to uppercase.	6
1.6	Write a C program to find total number of alphabets, digits or special character in a string.	7
1.7	Write a C program to count total number of words in a string.	8
1.8	Write a C program to find reverse of a string.	9
1.9	Write a C program to check whether a string is palindrome or not.	10
1.10	Write a C program to find first occurrence of a character in a given string.	11
1.11	Write a C program to count frequency of each character in a string.	12

1 Strings in C

1.1 Write a C program to find length of a string.

Source Code :

```
#include <stdio.h>

/// Finds length of a string without null terminator
int lenOfStr(char text[])
{
    int i = 0;
    while (text[i] != '\0')
        i++;
    return i - 1;
}

int main()
{
    char str[100];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    printf("Length of the string is: %d\n", lenOfStr(str));
    return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_06 on ʹ main [?] took 2s
→ gcc 01.c && ./a.out
Enter a string: Hello World
Length of the string is: 11
```

1.2 Write a C program to copy one string to another string.

Source Code :

```
#include <stdio.h>

/// Finds length of a string without null terminator
int lenOfStr(char text[])
{
    int i = 0;
    while (text[i] != '\0')
        i++;
    return i - 1;
}

/**
 * Copy one string to another.
 * @param a new string to assign value to
 * @param b old string to assign value from
 */
void copyStr(char a[], char b[])
{
    int len = lenOfStr(b);
    for (int i = 0; i <= len; i++)
        a[i] = b[i];
}

int main()
{
    char str[100], new_str[100];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    copyStr(new_str, str);
    printf("The copied string is: %s\n", new_str);
    return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_06 on 12 main [?]
→ gcc 02.c && ./a.out
Enter a string: Hey There
The copied string is: Hey There
```

1.3 Write a C program to concatenate two strings.

Source Code :

```
#include <stdio.h>

/// Finds length of a string without null terminator
int lenOfStr(char text[])
{
    int i = 0;
    while (text[i] != '\0')
        i++;
    return i - 1;
}

/// Concat two strings into one.
void concatStr(char a[], char b[], char new_str[])
{
    int len_a = lenOfStr(a), len_b = lenOfStr(b);
    for (int i = 0; i <= len_a + len_b; i++)
    {
        if (i < len_a)
            new_str[i] = a[i];
        else
            new_str[i] = b[i - len_a];
    }
}

int main()
{
    char str_a[100], str_b[100], str_c[200];
    printf("Enter first string: ");
    fgets(str_a, sizeof(str_a), stdin);
    printf("Enter second string: ");
    fgets(str_b, sizeof(str_b), stdin);
    concatStr(str_a, str_b, str_c);
    printf("Concatenated string: ");
    puts(str_c);
    return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_06 on 1/ main [?] took 4s
→ gcc 03.c && ./a.out
Enter first string: Hello
Enter second string: There
Concatenated string: Hello There
```

1.4 Write a C program to compare two strings.

Source Code :

```
#include <stdio.h>

/// Finds length of a string without null terminator
int lenOfStr(char text[])
{
    int i = 0;
    while (text[i] != '\0')
        i++;
    return i - 1;
}

/// Compare two strings.
/// Clones behavior of strcmp
int compStr(char a[], char b[])
{
    int len_a = lenOfStr(a), len_b = lenOfStr(b);
    if (len_a != len_b)
        return len_a > len_b ? 1 : -1;
    int i = 0;
    while (i <= len_a)
    {
        if (a[i] != b[i])
            return a[i] - b[i];
        i++;
    }
    return 0;
}

int main()
{
    char str_a[100], str_b[100];
    int r;
    printf("Enter first string: ");
    fgets(str_a, sizeof(str_a), stdin);
    printf("Enter second string: ");
    fgets(str_b, sizeof(str_b), stdin);
    r = compStr(str_a, str_b);
    printf("Both strings are %s. Output: %d\n", !r ? "Same" : "Different", r);
    return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_06 on 7 main [?] took 7s
→gcc 04.c && ./a.out
Enter first string: Asim
Enter second string: Bera
Both strings are Different. Output: -1

ccp-assignments/c_lang/assignment_06 on 7 main [?] took 5s
→gcc 04.c && ./a.out
Enter first string: asim
Enter second string: asim
Both strings are Same. Output: 0
```

1.5 Write a C program to convert lowercase string to uppercase.

Source Code :

```
#include <stdio.h>

/// Finds length of a string without null terminator
int lenOfStr(char text[])
{
    int i = 0;
    while (text[i] != '\0')
        i++;
    return i - 1;
}

/// Convert lowercase characters to uppercase
void upperCaseStr(char lower[], char upper[])
{
    int len_l = lenOfStr(lower);
    for (int i = 0; i <= len_l; i++)
        upper[i] = lower[i] >= 97 && lower[i] <= 122 ? lower[i] - 32 : lower[i];
}

int main()
{
    char str[100], str_u[100];
    printf("Enter a lowercase string: ");
    fgets(str, sizeof(str), stdin);
    upperCaseStr(str, str_u);
    printf("Uppercase: %s\n", str_u);
    return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_06 on / main [!?] took 3s
→gcc 05.c && ./a.out
Enter a lowercase string: %% string !!
Uppercase: %% STRING !!
```

1.6 Write a C program to find total number of alphabets, digits or special character in a string.

Source Code :

```
#include <stdio.h>

/// Finds length of a string without null terminator
int lenOfStr(char text[])
{
    int i = 0;
    while (text[i] != '\0')
        i++;
    return i - 1;
}

/// Prints total number of alphabets, digits and spcial characters
void printCount(char val[])
{
    int alpha = 0, digits = 0, spc = 0, len = lenOfStr(val);
    for (int i = 0; i < len; i++)
    {
        if ((val[i] >= 97 && val[i] <= 122) || (val[i] >= 65 && val[i] <= 90))
            alpha++;
        else if (val[i] >= 48 && val[i] <= 57)
            digits++;
        else
            spc++;
    }
    printf("Result:\nAlphabets: %d, Digits: %d, Special: %d\n", alpha, digits, spc);
}

int main()
{
    char str[100];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    printCount(str);
    return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_06 on ʹ main [!?] took 49s
→gcc 06.c && ./a.out
Enter a string: Ara.Ara...!@#123
Result:
Alphabets: 6, Digits: 3, Special: 7
```


1.7 Write a C program to count total number of words in a string.

Source Code :

```
#include <stdio.h>

/// Finds length of a string without null terminator
int lenOfStr(char text[])
{
    int i = 0;
    while (text[i] != '\0')
        i++;
    return i - 1;
}

/// Prints total number of words
void printTotalWords(char val[])
{
    int words = 0, len = lenOfStr(val);
    int cur = 0, next = 0;
    for (int i = 0; i < len; i++)
    {
        cur = (val[i] >= 97 && val[i] <= 122) || (val[i] >= 65 && val[i] <= 90);
        next = (val[i + 1] >= 97 && val[i + 1] <= 122) || (val[i + 1] >= 65 && val[i + 1] <=
        → 90);
        if (i == 0 && cur)
            words++;
        if (val[i] == 32 && next)
            words++;
    }
    printf("Number of words: %d.\n", words);
}

int main()
{
    char str[100];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    printTotalWords(str);
    return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_06 on ʹ main [!?] took 1m 25s
→gcc 07.c && ./a.out
Enter a string: Have a happy new year....
Number of words: 5.
```

1.8 Write a C program to find reverse of a string.

Source Code :

```
#include <stdio.h>

/// Finds length of a string without null terminator
int lenOfStr(char text[])
{
    int i = 0;
    while (text[i] != '\0')
        i++;
    return i - 1;
}

/// Reverse a string
void reverseStr(char val[], char rev[])
{
    int len_l = lenOfStr(val);
    for (int i = 0; i < len_l; i++)
        rev[len_l - i - 1] = val[i];
    // add null terminator at the end
    rev[len_l] = '\0';
}

int main()
{
    char str[100], str_r[100];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    reverseStr(str, str_r);
    printf("Reversed: %s\n", str_r);
    return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_06 on / main [!?]
→gcc 08.c && ./a.out
Enter a string: asim bera
Reversed: areb misa
```

1.9 Write a C program to check whether a string is palindrome or not.

Source Code :

```
#include <stdio.h>

/// Finds length of a string without null terminator
int lenOfStr(char text[])
{
    int i = 0;
    while (text[i] != '\0')
        i++;
    return i - 1;
}

/// Checks wheather the string is palindrome or not
int reverseStr(char val[])
{
    int len_l = lenOfStr(val), p = 1;
    for (int i = 0; i < len_l; i++)
    {
        if (val[i] != val[len_l - i - 1])
        {
            p = 0;
            break;
        }
    }
    return p;
}

int main()
{
    char str[100];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    printf("The string %s a palindrome.\n", reverseStr(str) ? "is" : "is not");
    return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_06 on / main [!?]
→gcc 09.c && ./a.out
Enter a string: asdsa
The string is palindrome.

ccp-assignments/c_lang/assignment_06 on / main [!?] took
→gcc 09.c && ./a.out
Enter a string: asim
The string is not a palindrome.
```

1.10 Write a C program to find first occurrence of a character in a given string.

Source Code :

```
#include <stdio.h>

/// Finds length of a string without null terminator
int lenOfStr(char text[])
{
    int i = 0;
    while (text[i] != '\0')
        i++;
    return i - 1;
}

/// Finds first occur position of a character
int findChar(char text[], char q)
{
    int len = lenOfStr(text), pos = -1;
    for (int i = 0; i < len; i++)
    {
        if (text[i] == q)
        {
            pos = i;
            break;
        }
    }
    return pos;
}

int main()
{
    char str[100], query;
    int pos;
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    printf("Enter char to query: ");
    scanf(" %c", &query);
    pos = findChar(str, query);
    if (pos < 0)
        printf("Character not found.\n");
    else
        printf("Character found as position (Starting at 0): %d.\n", pos);
    return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_06 on ʘ main [!?] took 13s
→gcc 10.c && ./a.out
Enter a string: hey there.
Enter char to query: r
Character found as position (Starting at 0): 7.
```

```
ccp-assignments/c_lang/assignment_06 on ʘ main [!?] took 12s
→gcc 10.c && ./a.out
Enter a string: hello
Enter char to query: &
Character not found.
```

1.11 Write a C program to count frequency of each character in a string.

Source Code :

```
#include <stdio.h>

/// Finds length of a string without null terminator
int lenOfStr(char text[])
{
    int i = 0;
    while (text[i] != '\0')
        i++;
    return i - 1;
}

/// Count frequency of each and every character
void countFreq(char text[])
{
    int len = lenOfStr(text), char_map[200]; // ascii
    for (int i = 0; i < 200; i++)
        char_map[i] = 0;
    for (int i = 0; i < len; i++)
        char_map[text[i]]++;

    printf("Frequency of characters: \n");
    for (int j = 0; j < 200; j++)
    {
        if (char_map[j])
            printf("%c: %d, ", j, char_map[j]);
    }
    printf("\n");
}

int main()
{
    char str[100];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    countFreq(str);
    return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_06 on ʘ main [!?]
→gcc 11.c && ./a.out
Enter a string: This is a STRING
Frequency of characters:
: 3, G: 1, I: 1, N: 1, R: 1, S: 1, T: 2, a: 1, h: 1, i: 2, s: 2,
```