

1. Write a program in C to input a string and print it.

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    char str[50];
    fgets(str, sizeof str, stdin);
    printf("The string you entered is : %s\n", str);
    return 0;
}
```

2. Write a program in C to find the length of a string without using library functions.

```
int main() {
    char str[100];
    int l = 0;
    fgets(str, sizeof str, stdin);
    while (str[l] != '\0') {
        l++;
    }
    printf("Length of the string is : %d\n\n", l - 1);
    return 0; }
```

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

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define str_size 100
int main() {
    char str[str_size];    int i, wrd;
    fgets(str, sizeof str, stdin);

    i = 0;
    wrd = 1;
    while (str[i] != '\0') {
        if (str[i] == ' ' || str[i] == '\n' || str[i] == '\t')
        {
            wrd++;
        }

        i++;
    }

    printf("Total number of words in the string is : %d\n",
    wrd - 1);
    return 0;
}
```

4. Count total number of alphabets, digits and special characters.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define str_size 100
int main() {
    char str[str_size];
    int alp, digit, splch, i;
    alp = digit = splch = i = 0;
    fgets(str, sizeof str, stdin);
    while (str[i] != '\0')
    {
        if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' &&
str[i] <= 'Z')) {
            alp++;
        }
        else if (str[i] >= '0' && str[i] <= '9') {
            digit++;
        }
        else {
            splch++;
        }
        i++;
    }

    printf("Number of Alphabets in the string is : %d\n",
alp);
    printf("Number of Digits in the string is : %d\n", digit);
    printf("Number of Special characters in the string is :
%d\n\n", splch);

    return 0;
}
```

5. Count the total number of vowels or consonants

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define str_size 100
int main() {
    char str[str_size];
    int i, len, vowel, cons;

    fgets(str, sizeof str, stdin);

    vowel = 0;
    cons = 0;
    len = strlen(str);
    for (i = 0; i < len; i++) {
```

```

        if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' ||
str[i] == 'o' || str[i] == 'u' || str[i] == 'A' || str[i] == 'E'
|| str[i] == 'I' || str[i] == 'O' || str[i] == 'U') {
            vowel++;
        }

        else if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >=
'A' && str[i] <= 'Z')) {
            cons++;
        }
    }

    printf("\nThe total number of vowel in the string is :
%d\n", vowel);
    printf("The total number of consonant in the string is :
%d\n\n", cons);
    return 0;
}

```

6. Find the maximum occurring character in a string

```

7. #include <stdio.h>
8. #include <string.h>
9. #include <stdlib.h>
10.
11.     #define str_size 100 // Declare the maximum size of the
    string
12.     #define chr_no 255 // Maximum number of characters to
    be allowed
13.
14.     int main() {
15.         char str[str_size]; // Declare a character array to
    store the string
16.         int ch_fre[chr_no]; // Declare an array to store the
    frequency of characters
17.         int i = 0, max; // Declare variables for
    iteration and finding the maximum frequency
18.         int ascii; // Variable to hold the ASCII
    value of a character
19.
20.         fgets(str, sizeof str, stdin); // Read a string
    from the standard input (keyboard)
21.
22.         for (i = 0; i < chr_no; i++) { // Initialize the
    frequency of all characters to zero
23.             ch_fre[i] = 0;
24.         }
25.

```

```

26.         /* Read for frequency of each character */
27.         i = 0;
28.         while (str[i] != '\0') { // Loop until the end of the
string ('\0' character is encountered)
29.             ascii = (int)str[i]; // Get the ASCII value of
the current character
30.             ch_fre[ascii] += 1; // Increment the frequency
count for that character
31.             i++;
32.         }
33.
34.         max = 0;
35.         for (i = 0; i < chr_no; i++) {
36.             if (i != 32) { // Exclude space (' ') character
37.                 if (ch_fre[i] > ch_fre[max]) {
38.                     max = i;
39.                 }
40.             }
41.         }
42.
43.         printf("The Highest frequency of character '%c'
appears number of times : %d \n\n", max, ch_fre[max]); //
Display the character with maximum frequency and its count
44.
45.         return 0;
46. }

```

7. Write a C program to check whether a substring is present in a string.

```

#include <stdio.h>

int main() {
    char str[80], search[20]; // Declare character arrays for the
main string and the substring to search
    int c1 = 0, c2 = 0, i, j, flg; // Declare variables for
counting, iteration, and flag for substring check

    printf("\n\nCheck whether a given substring is present in the
given string :\n"); // Display information about the task
    printf("-----\n");

    printf("Input the string : ");
    fgets(str, sizeof str, stdin); // Read a string from the
standard input (keyboard)

    printf("Input the substring to be searched : ");

```

```
fgets(search, sizeof search, stdin); // Read the substring to be searched
```

```
// Calculate the length of the main string
while (str[c1] != '\0') {
    c1++;
}
c1--;
```

```
// Calculate the length of the substring
while (search[c2] != '\0') {
    c2++;
}
c2--;
```

```
// Check if the substring exists in the main string
for (i = 0; i <= c1 - c2; i++) {
    for (j = i; j < i + c2; j++) {
        flg = 1; // Initialize the flag for matching substring
        if (str[j] != search[j - i]) {
            flg = 0; // Set the flag to zero if characters don't
```

match

```
                break;
            }
        }
        if (flg == 1) {
            break; // Break the loop if substring is found
        }
    }
}
```

```
if (flg == 1) {
    printf("The substring exists in the string.\n\n"); //
Display message if the substring is found
} else {
    printf("The substring does not exist in the string.\n\n");
// Display message if the substring is not found
```

```
    return 0; // Return 0 to indicate successful execution of
the program
}
}
```