

**Name – Gajanan Purud**

## **Assignment 11**

- 1) write a program to scan string from user then scan a single character and search it in a accepted string.

```
#include <stdio.h>

#include <string.h>

int main() {

char str[100], ch;

int count = 0;

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

int len = strlen(str);

if (len > 0 && str[len - 1] == '\n') {

str[len - 1] = '\0';

}

printf("Enter a character to search: ");

scanf(" %c", &ch); // Note the space before %c to consume leftover newline

for (int i = 0; str[i] != '\0'; i++) {

if (str[i] == ch) {

count++;

}

}

if (count > 0) {

printf("The character '%c' was found %d time(s).\n", ch, count);

} else {

printf("The character '%c' was not found.\n", ch);

}

return 0;}
```

Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Programiz C Online Compiler

main.c

```
1 #include <stdio.h>
2 #include <string.h>
3 int main() {
4     char str[100], ch;
5     int count = 0;
6     printf("Enter a string: ");
7     fgets(str, sizeof(str), stdin);
8     int len = strlen(str);
9     if (len > 0 && str[len - 1] == '\n') {
10         str[len - 1] = '\0';
11     }
12     printf("Enter a character to search: ");
13     scanf("%c", &ch); // Note the space before %c to consume leftover newline
14     for (int i = 0; str[i] != '\0'; i++) {
15         if (str[i] == ch) {
16             count++;
17         }
18     }
19     if (count > 0) {
20         printf("The character '%c' was found %d time(s).\n", ch, count);
21     } else {
22         printf("The character '%c' was not found.\n", ch);
23     }
24     return 0;
25 }
```

Output

Enter a string: gajanan  
Enter a character to search: j  
The character 'j' was found 1 time(s).

=== Code Execution Successful ===

Activate Windows  
Go to Settings to activate Windows.

## 2) WAP Replace all Occurrences of 'a' with \$ in a String

```
#include <stdio.h>

#include <string.h>

int main() {

    char str[100];

    printf("Enter a string: ");

    fgets(str, sizeof(str), stdin);

    int len = strlen(str);

    if (len > 0 && str[len - 1] == '\n') {

        str[len - 1] = '\0';

    }

    for (int i = 0; str[i] != '\0'; i++) {

        if (str[i] == 'a') {

            str[i] = '$';

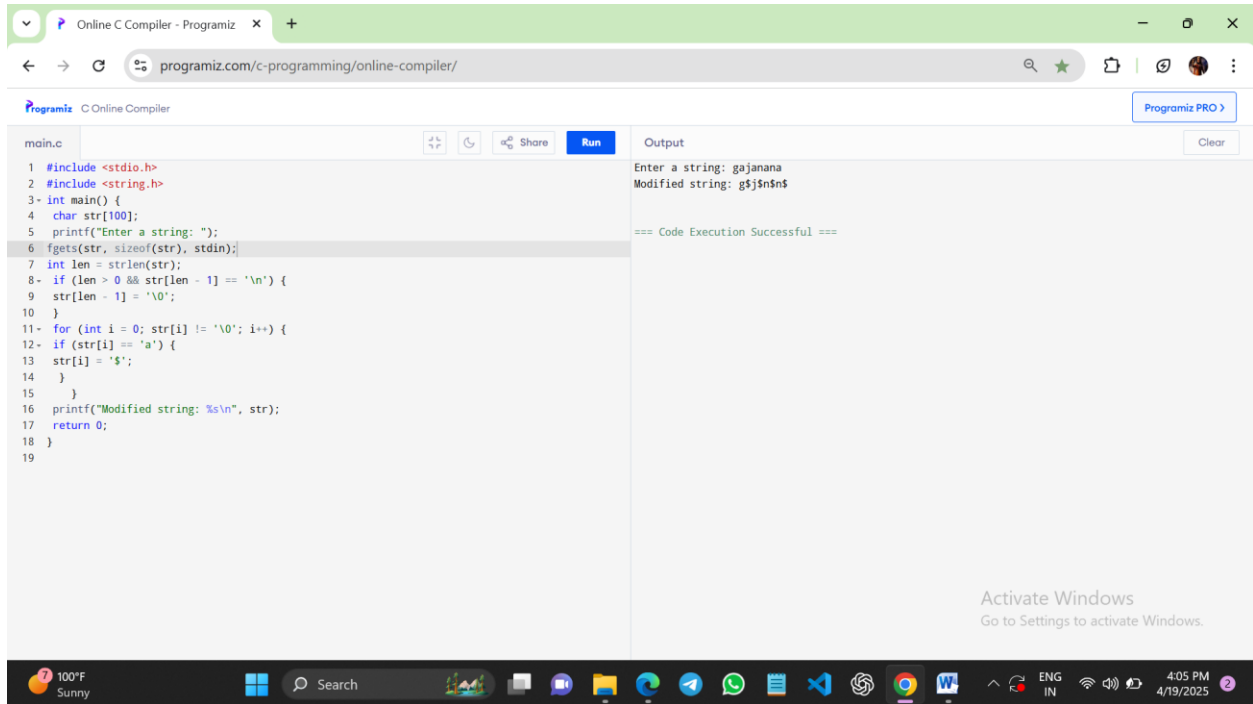
        }

    }

    printf("Modified string: %s\n", str);
```

```
return 0;
```

```
}
```



The screenshot shows a web browser window with the URL `programiz.com/c-programming/online-compiler/`. The page title is "Online C Compiler - Programiz". The code editor contains the following C program:

```
1 #include <stdio.h>
2 #include <string.h>
3 int main() {
4     char str[100];
5     printf("Enter a string: ");
6     fgets(str, sizeof(str), stdin);
7     int len = strlen(str);
8     if (len > 0 && str[len - 1] == '\n') {
9         str[len - 1] = '\0';
10    }
11    for (int i = 0; str[i] != '\0'; i++) {
12        if (str[i] == 'a') {
13            str[i] = '$';
14        }
15    }
16    printf("Modified string: %s\n", str);
17    return 0;
18 }
19
```

The output window shows the following text:

```
Enter a string: gajana
Modified string: g$j$na$

=== Code Execution Successful ===
```

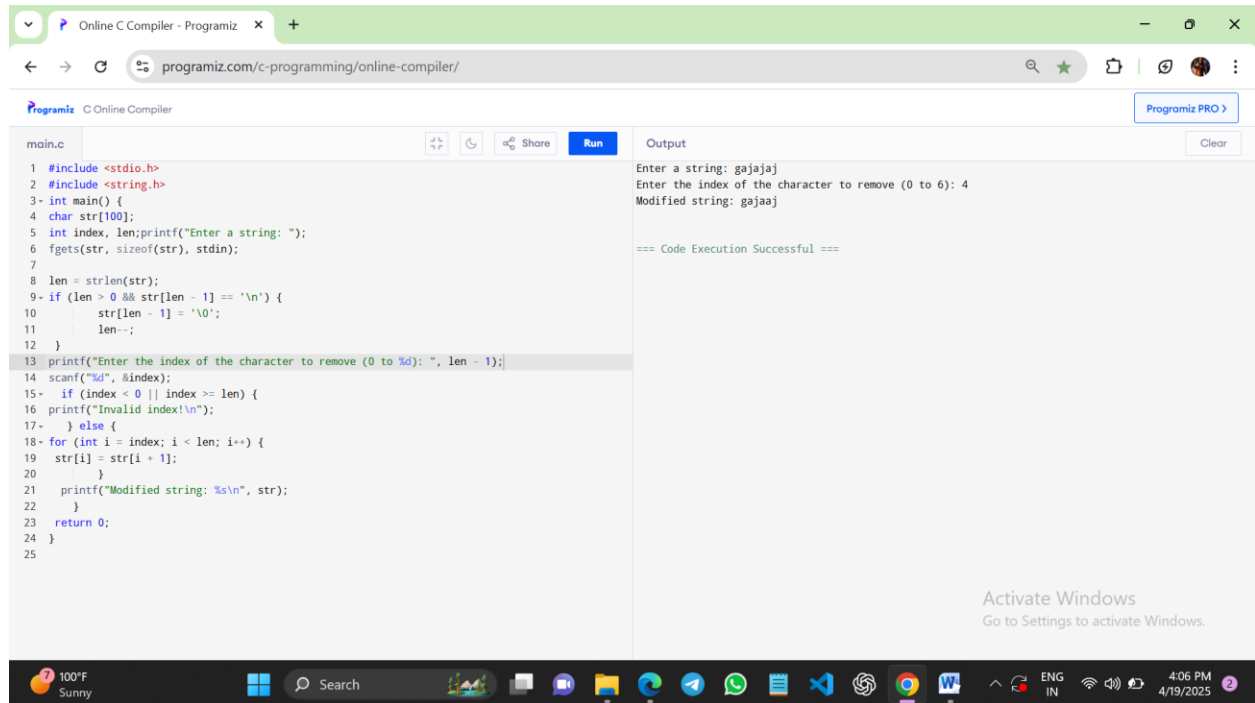
The Windows taskbar at the bottom shows the date and time as 4:05 PM on 4/19/2025.

### 3) WAP to Remove the nth Index Character from a Non-Empty String

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

int main() {
    char str[100];
    int index, len; printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    len = strlen(str);
    if (len > 0 && str[len - 1] == '\n') {
        str[len - 1] = '\0';
        len--;
    }
    printf("Enter the index of the character to remove (0 to %d): ", len - 1);
    scanf("%d", &index);
    if (index < 0 || index >= len) {
        printf("Invalid index!\n");
    } else {
        for (int i = index; i < len; i++) {
            str[i] = str[i + 1];
        }
        printf("Modified string: %s\n", str);
    }
    return 0;
}
```



```
main.c
1 #include <stdio.h>
2 #include <string.h>
3 int main() {
4     char str[100];
5     int index, len; printf("Enter a string: ");
6     fgets(str, sizeof(str), stdin);
7
8     len = strlen(str);
9     if (len > 0 && str[len - 1] == '\n') {
10         str[len - 1] = '\0';
11         len--;
12     }
13     printf("Enter the index of the character to remove (0 to %d): ", len - 1);
14     scanf("%d", &index);
15     if (index < 0 || index >= len) {
16         printf("Invalid index!\n");
17     } else {
18         for (int i = index; i < len; i++) {
19             str[i] = str[i + 1];
20         }
21         printf("Modified string: %s\n", str);
22     }
23     return 0;
24 }
25
```

Output

```
Enter a string: gajaja
Enter the index of the character to remove (0 to 6): 4
Modified string: gajaa

=== Code Execution Successful ===
```

Activate Windows  
Go to Settings to activate Windows.

#### 4) WAP to Form a New String where the First Character and the Last Character have been Exchanged

```
#include <stdio.h>

#include <string.h>

int main() {

    char str[100], temp;

    int len;

    printf("Enter a string: ");

    fgets(str, sizeof(str), stdin);

    len = strlen(str);

    if (len > 0 && str[len - 1] == '\n') {

        str[len - 1] = '\0';

        len--;

    }

    if (len > 1) {

        temp = str[0];

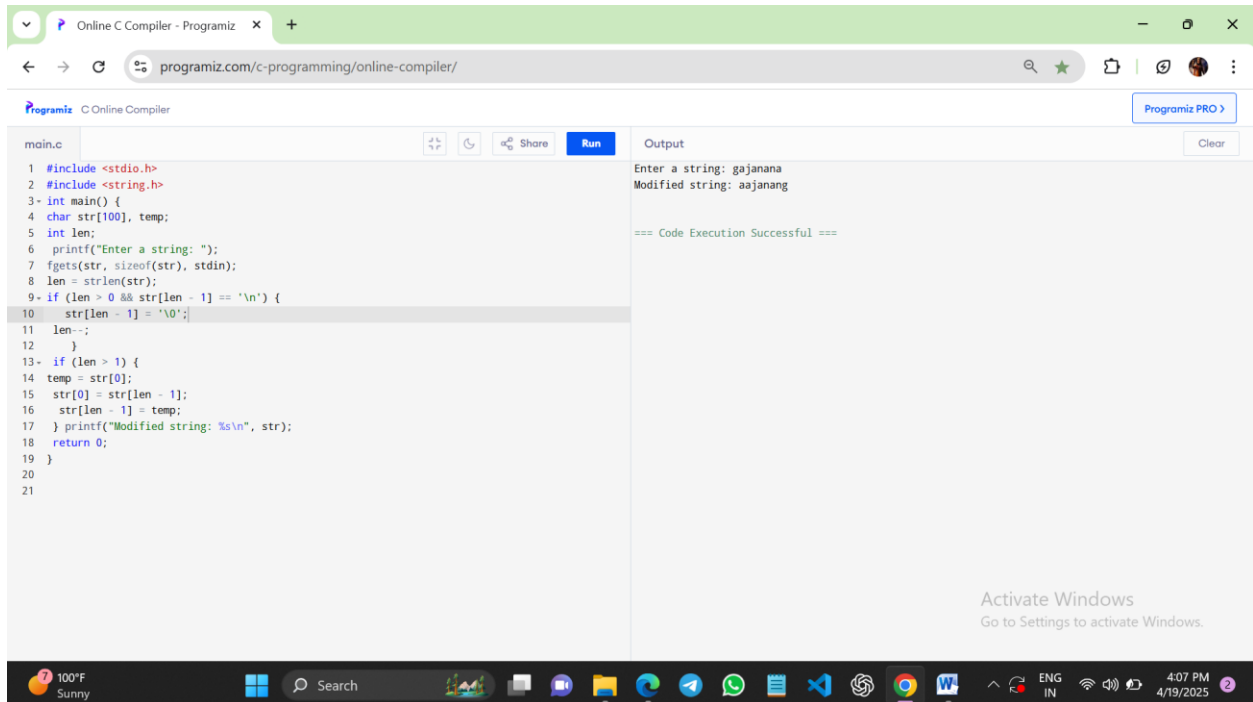
        str[0] = str[len - 1];

        str[len - 1] = temp;
```

```
} printf("Modified string: %s\n", str);
```

```
return 0;
```

```
}
```



The screenshot shows a web browser window with the URL `programiz.com/c-programming/online-compiler/`. The page title is "Online C Compiler - Programiz". The interface includes a code editor on the left and an output window on the right. The code in the editor is a C program that reverses a string. The output window shows the input string "gajanana" and the modified string "aajanang". The program execution was successful.

```
main.c
1 #include <stdio.h>
2 #include <string.h>
3 int main() {
4     char str[100], temp;
5     int len;
6     printf("Enter a string: ");
7     fgets(str, sizeof(str), stdin);
8     len = strlen(str);
9     if (len > 0 && str[len - 1] == '\n') {
10        str[len - 1] = '\0';
11        len--;
12    }
13    if (len > 1) {
14        temp = str[0];
15        str[0] = str[len - 1];
16        str[len - 1] = temp;
17    } printf("Modified string: %s\n", str);
18    return 0;
19 }
20
21
```

Output

```
Enter a string: gajanana
Modified string: aajanang

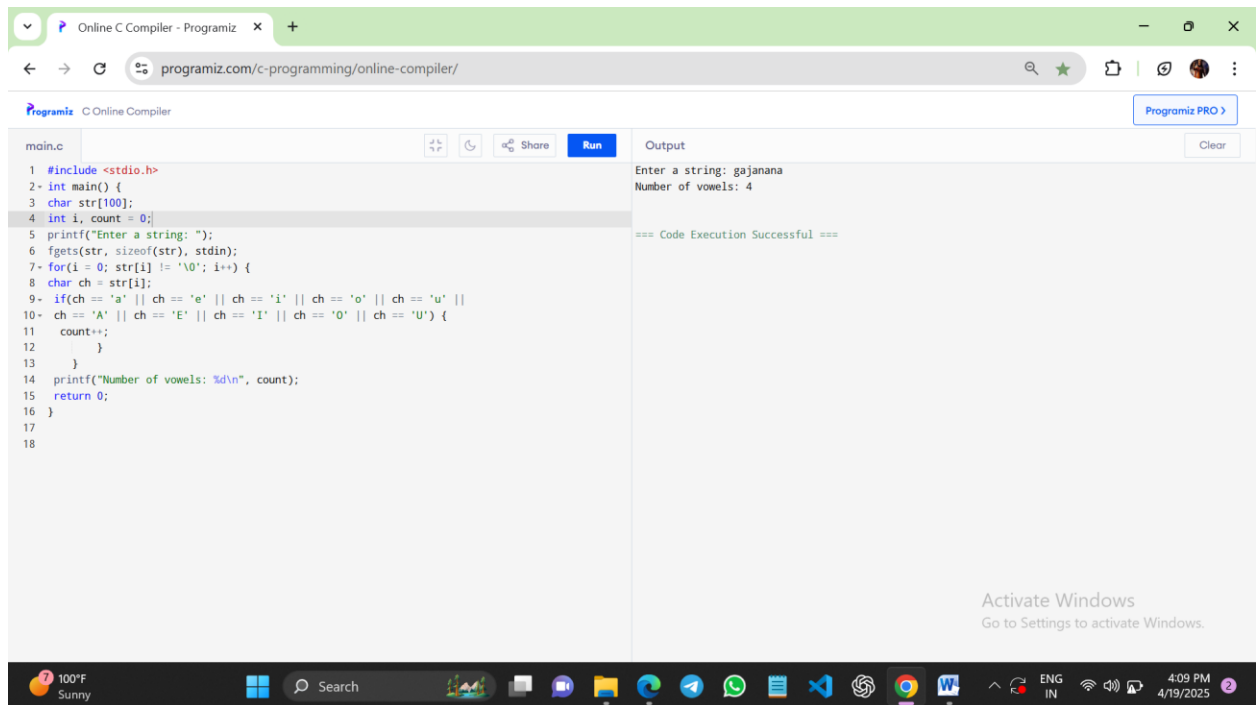
=== Code Execution Successful ===
```

Activate Windows  
Go to Settings to activate Windows.

## 5) WAP to Count the Number of Vowels in a String

```
#include <stdio.h>

int main() {
    char str[100];
    int i, count = 0;
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    for(i = 0; str[i] != '\0'; i++) {
        char ch = str[i];
        if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
           ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {
            count++;
        }
    }
    printf("Number of vowels: %d\n", count);
    return 0;
}
```



6) WAP to Take in a String and Replace Every Blank Space with special symbol.

```
#include <stdio.h>
int main() {
    char str[100];
    int i;
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    for(i = 0; str[i] != '\0'; i++) {
        if(str[i] == ' ') {
            str[i] = '#';
        }
    }
    printf("Modified string: %s\n", str);
    return 0;
}
```

### 7) WAP to Remove the Characters of Odd Index Values in a String

```
#include <stdio.h>

int main() {

    char str[100], result[100];

    int i, j = 0;

    printf("Enter a string: ");

    fgets(str, sizeof(str), stdin);

    for(i = 0; str[i] != '\0'; i++) {

        if(i % 2 == 0) {

            result[j++] = str[i];

        }

    }

    result[j] = '\0'; // Null-terminate the new string

    printf("String after removing characters at odd indices: %s\n", result);

    return 0;

}
```

### 8) WAP to Calculate the Number of Words Present in a String

```
#include <stdio.h>

#include <ctype.h>

int main() {

    char str[200];

    int i, wordCount = 0, inWord = 0;

    printf("Enter a string: ");

    fgets(str, sizeof(str), stdin);
```

```

for(i = 0; str[i] != '\0'; i++) {

    if (!isspace(str[i])) {

        if (!inWord) {

            wordCount++;

            inWord = 1;

        }

    } else {

        inWord = 0;

    }

}

return 0;

}

```

## 9) WAP to Take in Two Strings and Display the Larger String without Using Built-in Functions

```

#include <stdio.h>

int main() {
    char str1[100], str2[100];
    int len1 = 0, len2 = 0;

    printf("Enter first string: ");
    fgets(str1, sizeof(str1), stdin);

    printf("Enter second string: ");
    fgets(str2, sizeof(str2), stdin);

    while(str1[len1] != '\0') {
        if(str1[len1] == '\n') break;
        len1++;
    }

    while(str2[len2] != '\0') {
        if(str2[len2] == '\n') break;
        len2++;
    }

    if(len1 > len2) {
        printf("Larger string: %s", str1);
    } else if(len2 > len1) {
        printf("Larger string: %s", str2);
    } else {
        printf("Both strings are of equal length.\n");
    }

    return 0;
}

```



```
}
```

10) Write a program to check the string is palindrome or not.

```
include <stdio.h>
int main() {
char str[100];
int i, length = 0, isPalindrome = 1;
printf("Enter a string: ");
fgets(str, sizeof(str), stdin);
while(str[length] != '\0' && str[length] != '\n') {
length++;
}

for(i = 0; i < length / 2; i++) {
if(str[i] != str[length - 1 - i]) {
isPalindrome = 0;
break;
}
}

if(isPalindrome) {
printf("The string is a palindrome.\n");
} else {
printf("The string is not a palindrome.\n");
}

return 0;
}
```