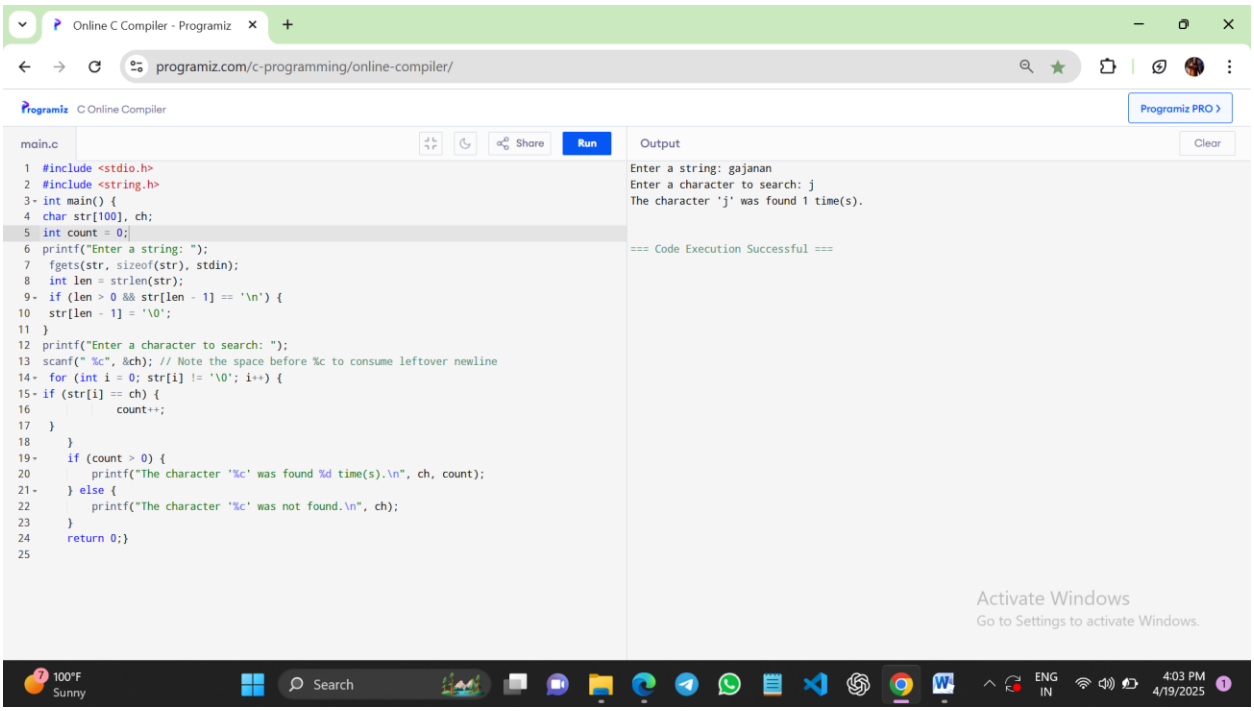
Name – Gajanan Purud

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;}
```

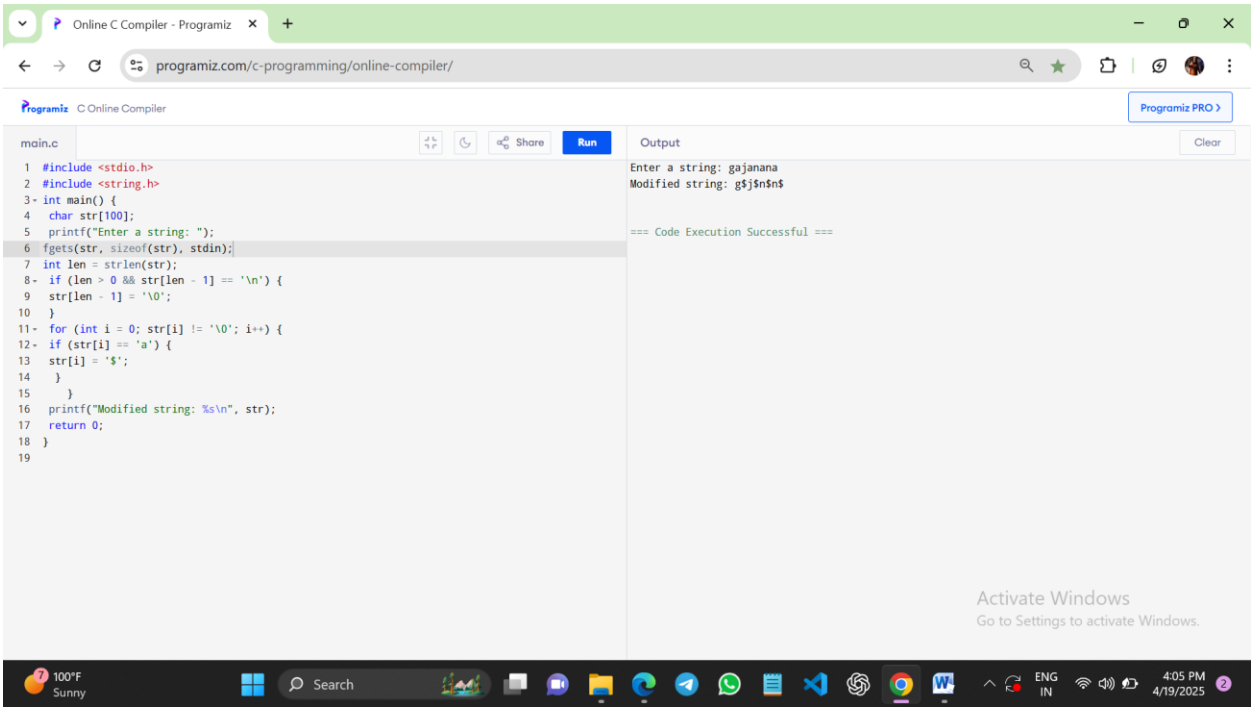


2) WAP Replace all Occurrences of ‘a’ with $ in a String

```
#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);
```

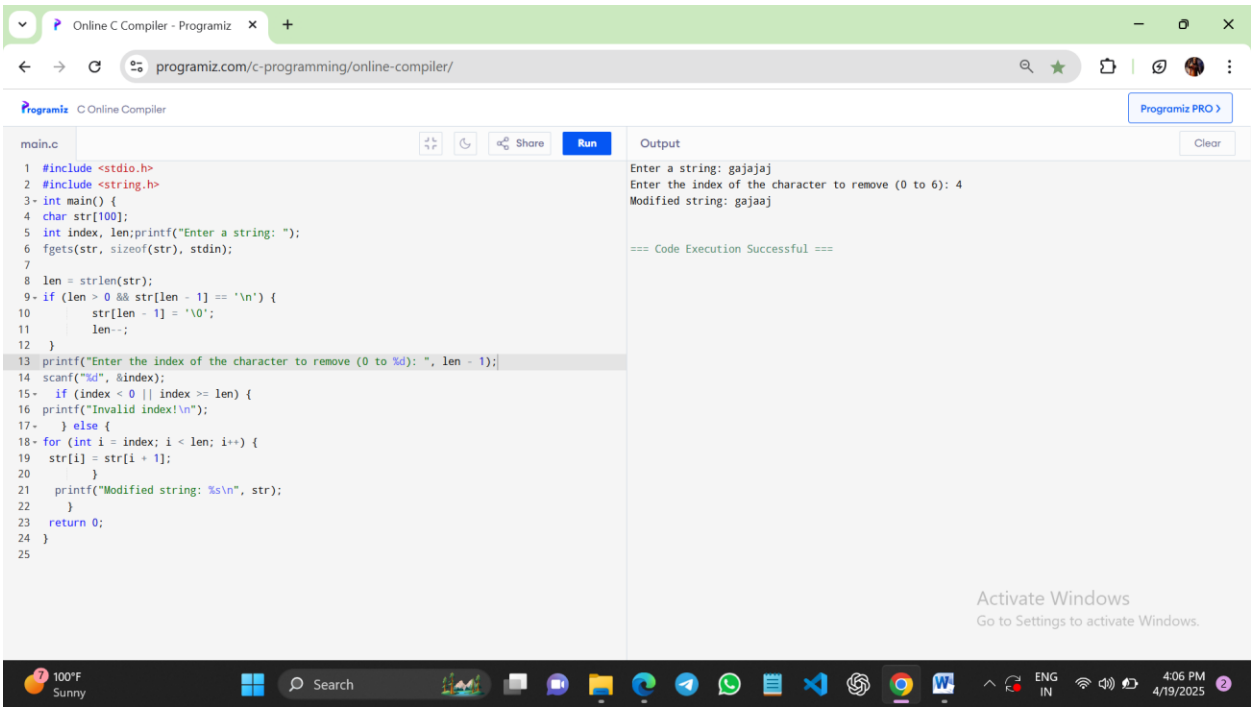
return 0;

}



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

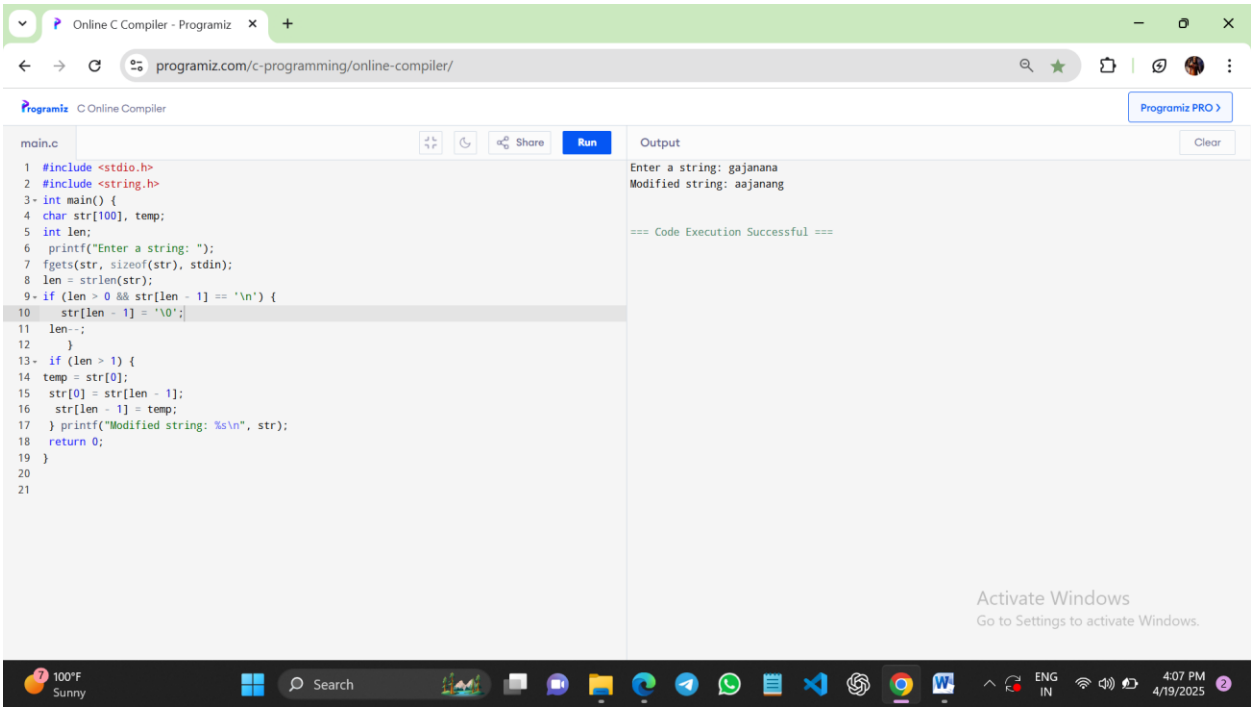
```
#include <stdio.h>
#include <string.h>
int main() {
char str[100];
int index, len;printf("Enter a string: ");
fgets(str, sizeof(str), stdin);

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;
}
```



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

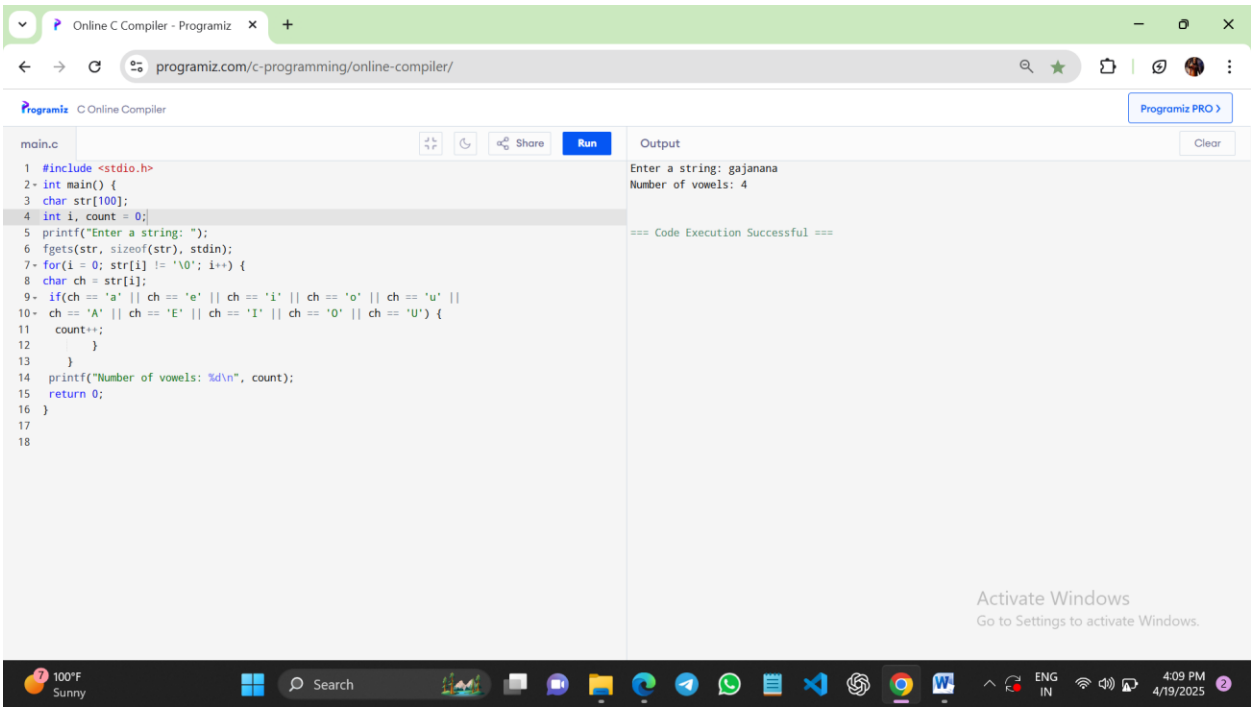
```
#include <stdio.h>

#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;

}
```



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;
}
```