**Arrays and Strings**

**2019115126**

**Swaminathan Navinashok**

Pgm 5.1.1

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

#include <stdio.h>

#include <string.h>

// A function to check if a string str is palindrome

void isPalindrome(char str[])

{

// Start from leftmost and rightmost corners of str

int l = 0;

int h = strlen(str) - 1;

// Keep comparing characters while they are same

while (h > l)

{

if (str[l++] != str[h--])

{

printf("%s is Not Palindrome", str);

return;

}

}

printf("%s is palindrome", str);

}

// Driver program to test above function

int main()

{

char string1[20];

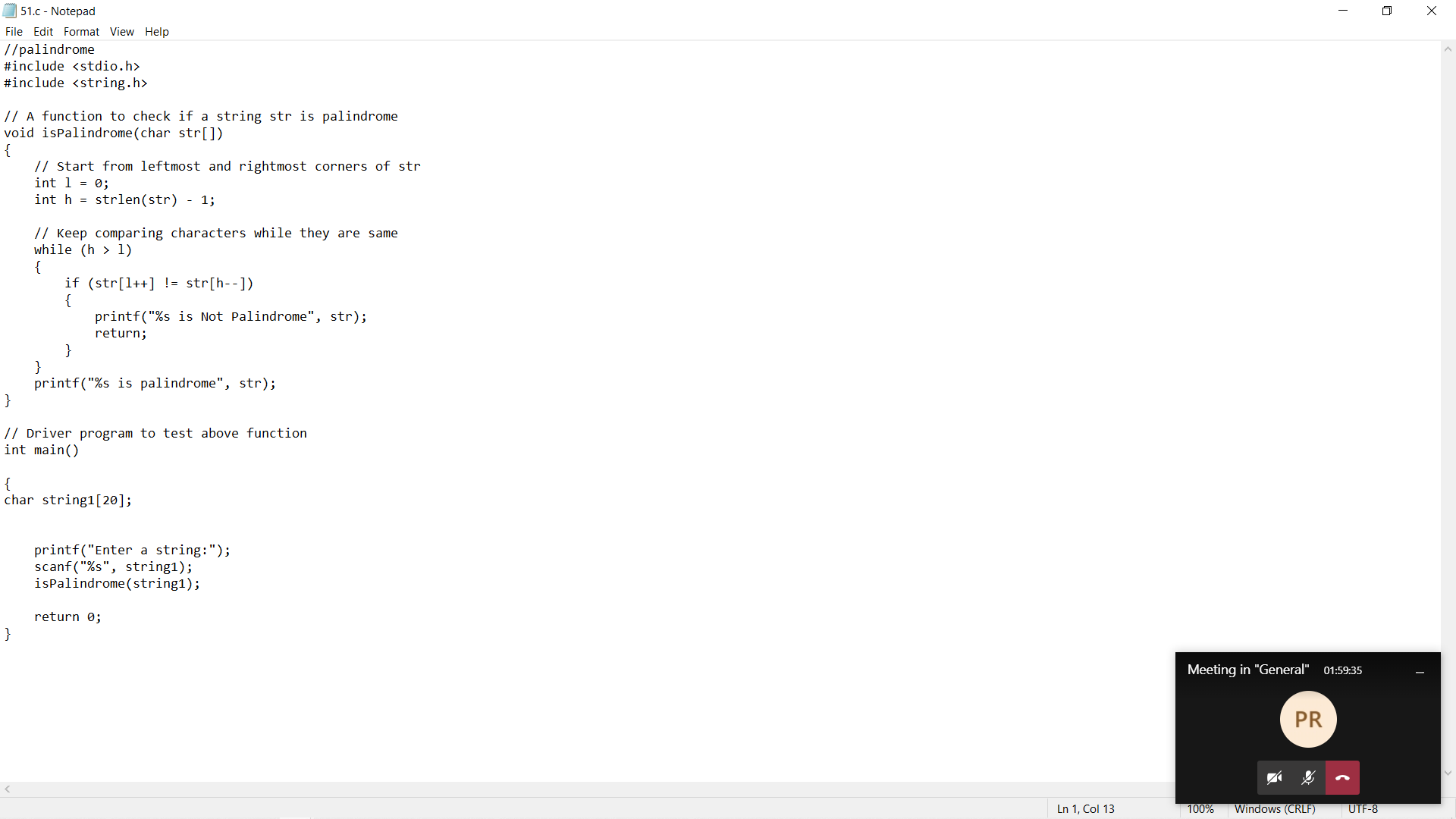
printf("Enter a string:");

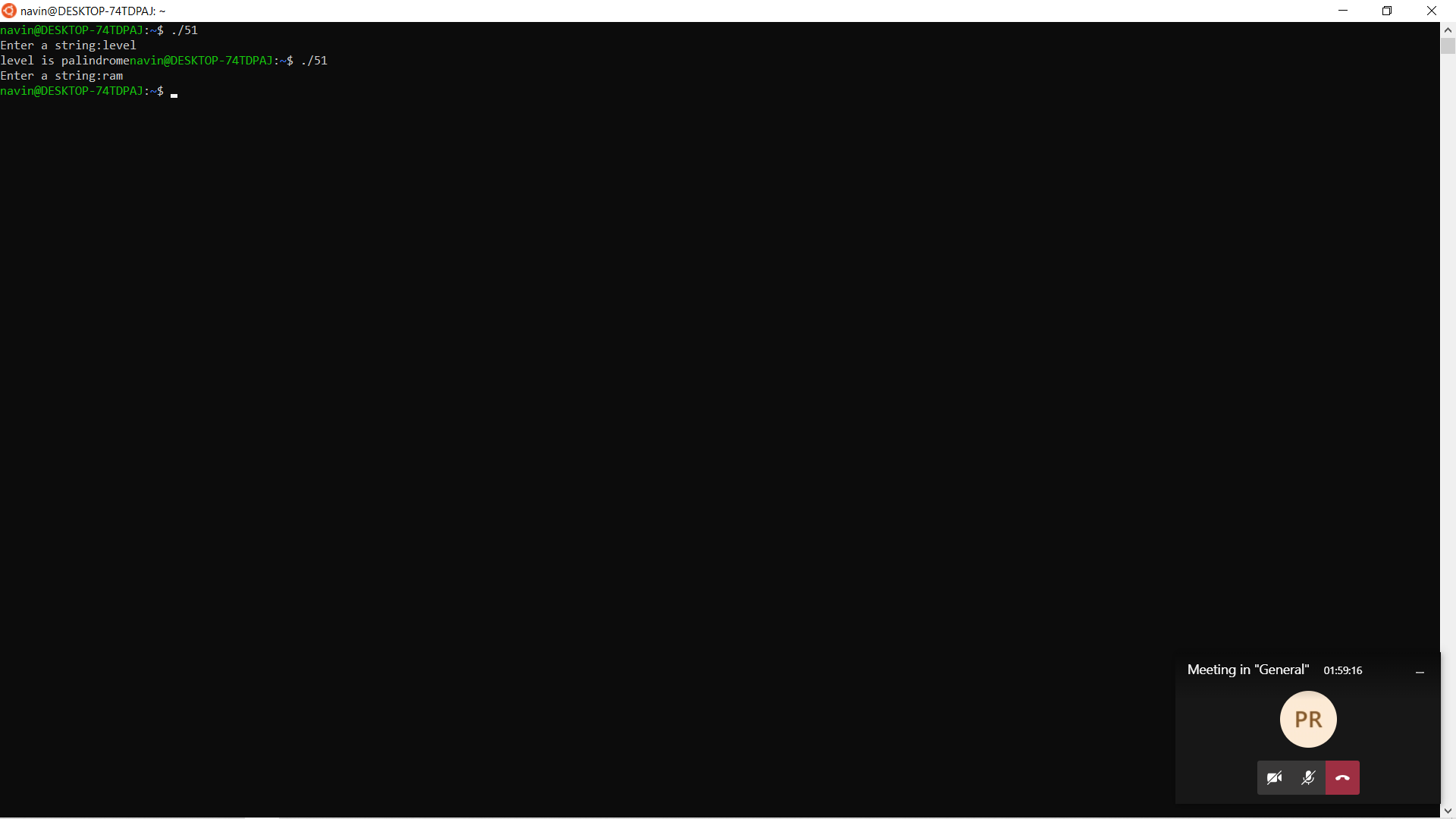
scanf("%s", string1);

isPalindrome(string1);

return 0;

}





Pgm 5.1.2

### C program to implement the following string functions.

### Mystrlen()

### // C program to find the length of string

### #include <stdio.h>

### #include <string.h>

### void mystrlen(char Str[])

### {

### int i;

### for (i = 0; Str[i] != '\0'; ++i);

### 

### printf("Length of Str is %d", i);

### }

### int main()

### {

### char Str[1000];

### 

### 

### printf("Enter the String: ");

### scanf("%s", Str);

### 

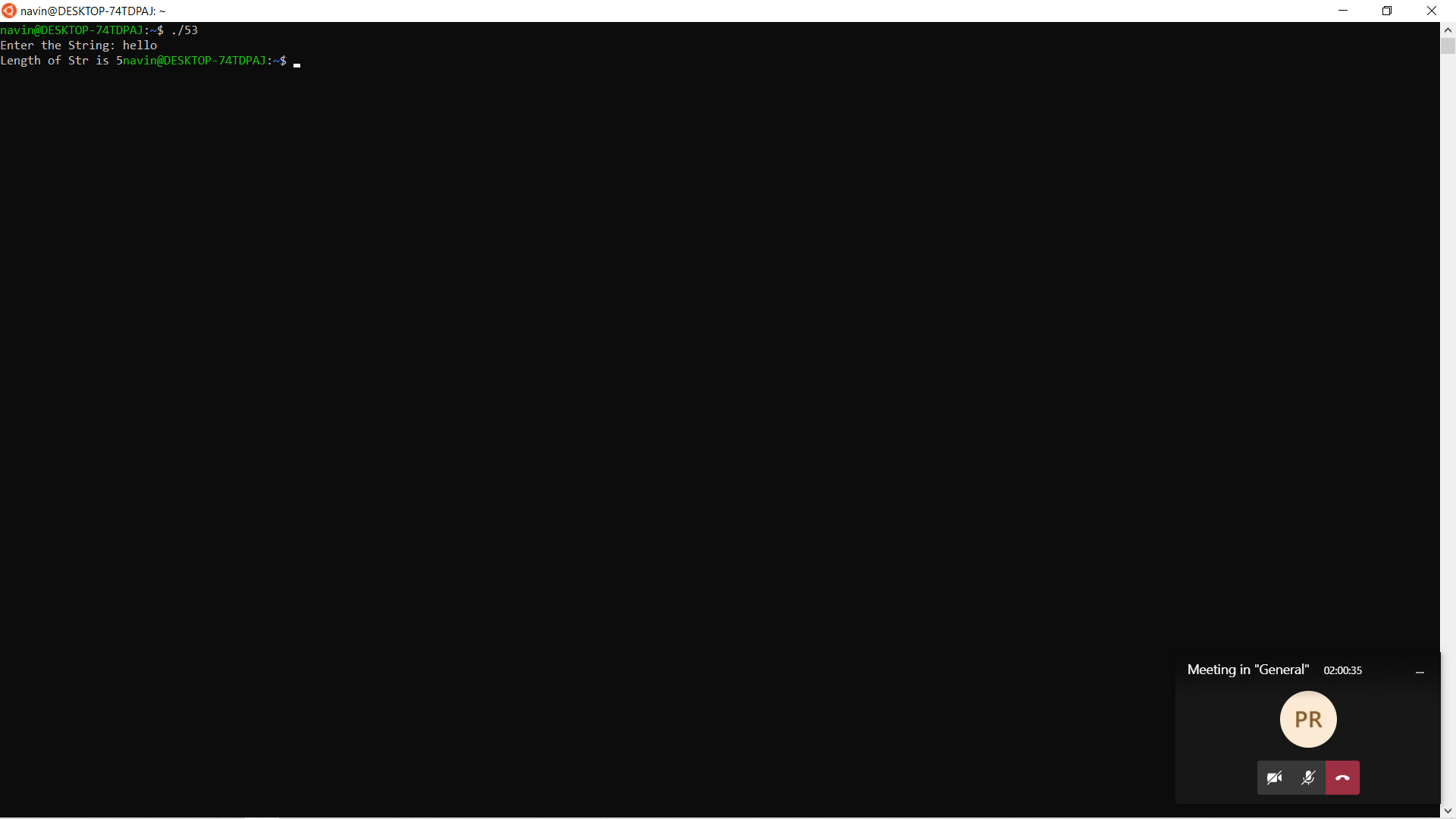
### mystrlen(Str);

### 

### return 0;

### }

### 



### Mystrcmp()

### //strcmp not inbuilt

### #include<stdio.h>

### int mystrcmp(char[], char[]); // function prototype declaration

### int main()

### {

### char aj1[100], aj2[100];

### int compare;

### printf("\n\nEnter 1st string: ");

### scanf("%s", aj1);

### printf("\n\nEnter 2nd string: ");

### scanf("%s", aj2);

### compare = mystrcmp(aj1, aj2); // function call

### 

### if(compare == 1)

### printf("\n\nBoth the strings are exactly same.\n\n");

### else

### printf("\n\nBoth the strings are different.\n");

### 

### return 0;

### }

### int mystrcmp(char mj1[], char mj2[]) // function definition

### {

### int i = 0, flag = 0;

### while(mj1[i] != '\0' && mj2[i] != '\0') // until atleast one string ends

### {

### /\*

### Even if they differ by a single character,

### don't iterate further

### \*/

### if(mj1[i] != mj2[i])

### {

### flag = 1;

### break;

### }

### i++;

### }

### /\*

### If all the characters are sequentially same as well as

### both strings have ended

### \*/

### if(flag == 0 && mj1[i] == '\0' && mj2[i] == '\0')

### return 1;

### else

### return 0;

### }

### 

### Mystrcpy()

### // CPP program to copy one string to other

### // without using in-built function

### #include <stdio.h>

### 

### void mystrcpy(char s1[],char s2[])

### {

### int i;

### // Print the string s1

### printf("string s1 : %s\n", s1);

### 

### // Execute loop till null found

### for (i = 0; s1[i] != '\0'; ++i) {

### // copying the characters by

### // character to str2 from str1

### s2[i] = s1[i];

### }

### 

### s2[i] = '\0';

### 

### // printing the destination string

### printf("String s2 : %s", s2);

### }

### 

### int main()

### {

### // s1 is the source( input) string and s2 is the destination string

### char s1[100], s2[100];

### 

### printf("Enter the String: ");

### scanf("%s", s1);

### printf("Enter the String 2: ");

### scanf("%s", s2);

### mystrcpy(s1,s2);

### 

### return 0;

### }

### 

### 

### 

### Mystrrev()

### //string reverse

### #include<stdio.h>

### #include<string.h>

### void mystrrev(char mystrg[])

### {

### 

### int leng, g;// This will find the length of your string with the help of strlen() function of string.h header file

### leng = strlen(mystrg);

### // iterate through each and every character of the string for printing it backwards or reverse direction

### for(g = leng - 1; g >= 0; g--) {

### printf("%c", mystrg[g]);

### }

### }

### int main(void)

### {

### char mystrg[60];

### 

### // Printing the program name and what the program will do

### printf("Program in C for reversing a given string \n ");

### printf("Please insert the string you want to reverse: ");

### // fetch the input string from the user

### scanf( "%s", mystrg );

### mystrrev(mystrg);

### 

### return 0;

### }

### 

### 

Pgm 5.1.3

C program to count total number of alphabets (within which the count the number of vowels, consonants), digits and special characters in a string.

#include<stdio.h>

void main()

{

char str[200];

int i,vowels=0,consonants=0,digits=0,spaces=0,specialCharacters=0;

printf("Enter a string\n");

scanf( "%[^\n]s", str );

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

{

vowels++;

}

else if((str[i]>='a'&& str[i]<='z') || (str[i]>='A'&& str[i]<='Z'))

{

consonants++;

}

else if(str[i]>='0' && str[i]<='9')

{

digits++;

}

else if (str[i]==' ')

{

spaces++;

}

else

{

specialCharacters++;

}

}

printf("\nVowels = %d",vowels);

printf("\nConsonants = %d",consonants);

printf("\nDigits = %d",digits);

printf("\nWhite spaces = %d",spaces);

printf("\nSpecial characters = %d",specialCharacters);

}

