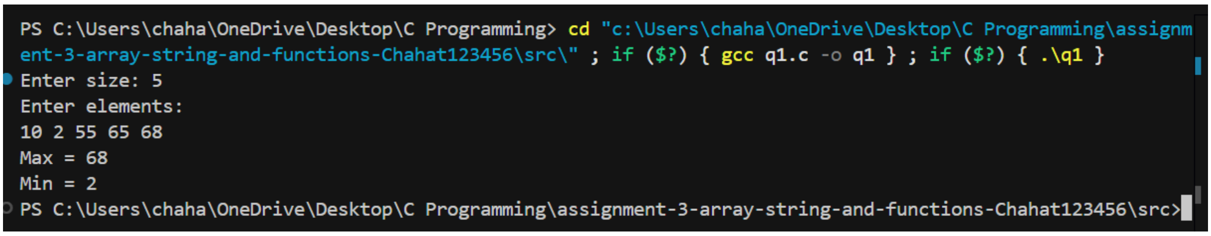
Name : Chahat R.S. Rathour

1. **Write a C program to find the maximum and minimum elements in an array.**

```
#include <stdio.h>
int main()
{
    int n, i;
    printf("Enter size: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter elements:\n");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    int max = arr[0], min = arr[0];

    for (i = 1; i < n; i++)
    {
        if (arr[i] > max)
            max = arr[i];
        if (arr[i] < min)
            min = arr[i];
    }

    printf("Max = %d\nMin = %d", max, min);
    return 0;
}
```

Output:

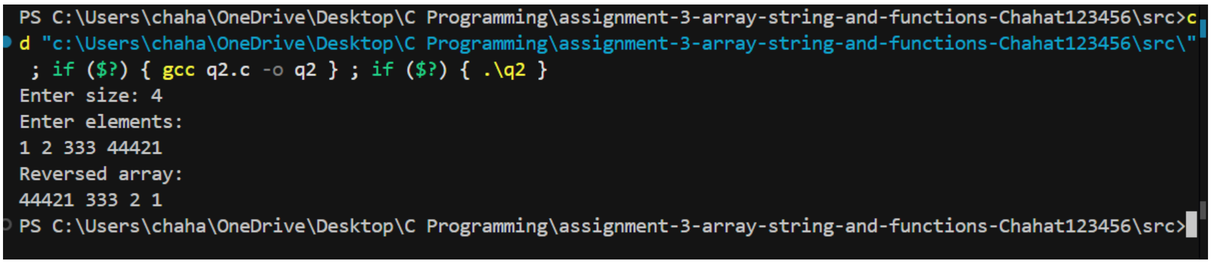


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming> cd "c:\Users\chaha\OneDrive\Desktop\C Programming\assignm
ent-3-array-string-and-functions-Chahat123456\src\" ; if ($?) { gcc q1.c -o q1 } ; if ($?) { .\q1 }
Enter size: 5
Enter elements:
10 2 55 65 68
Max = 68
Min = 2
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

2. **Implement a C program to reverse the elements of an array.**

```
#include <stdio.h>
int main()
{
    int n, i;
    printf("Enter size: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter elements: \n");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("Reversed array:\n");
    for (i = n - 1; i >= 0; i--)
        printf("%d ", arr[i]);

    return 0;
}
```

Output

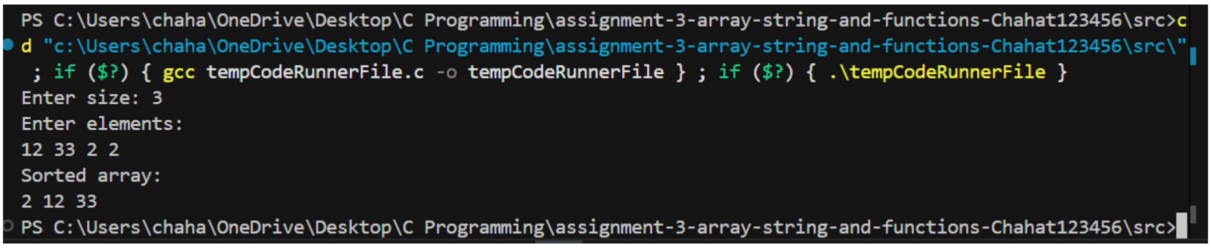


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc q2.c -o q2 } ; if ($?) { .\q2 }
Enter size: 4
Enter elements:
1 2 333 44421
Reversed array:
44421 333 2 1
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

3. **Write a C program to sort an array of integers in ascending order using a sorting algorithm of your choice (e.g., bubble sort, selection sort, insertion sort).**

```
#include <stdio.h>
int main()
{
    int n, i, j;
    printf("Enter size: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter elements:\n");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    for (i = 0; i < n - 1; i++)
        for (j = 0; j < n - i - 1; j++)
            if (arr[j] > arr[j + 1])
            {
                int t = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = t;
            }

    printf("Sorted array:\n");
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

Output

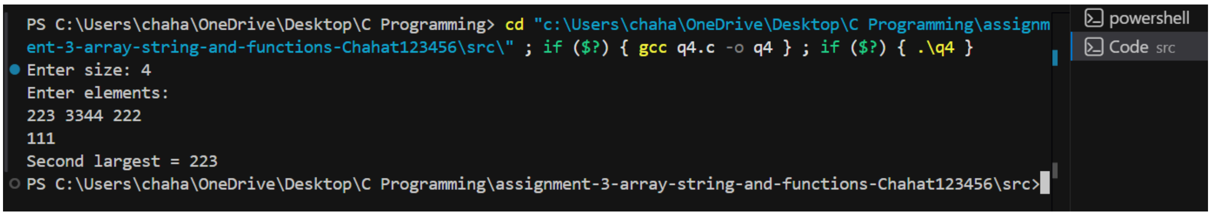


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter size: 3
Enter elements:
12 33 2 2
Sorted array:
2 12 33
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

4. Implement a C program to find the second largest element in an array.

```
#include <stdio.h>
int main()
{
    int n, i;
    printf("Enter size: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter elements:\n");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    int max = arr[0], second = -999999;
    for (i = 1; i < n; i++)
    {
        if (arr[i] > max)
        {
            second = max;
            max = arr[i];
        }
        else if (arr[i] > second && arr[i] != max)
        {
            second = arr[i];
        }
    }
    printf("Second largest = %d", second);
    return 0;
}
```

Output

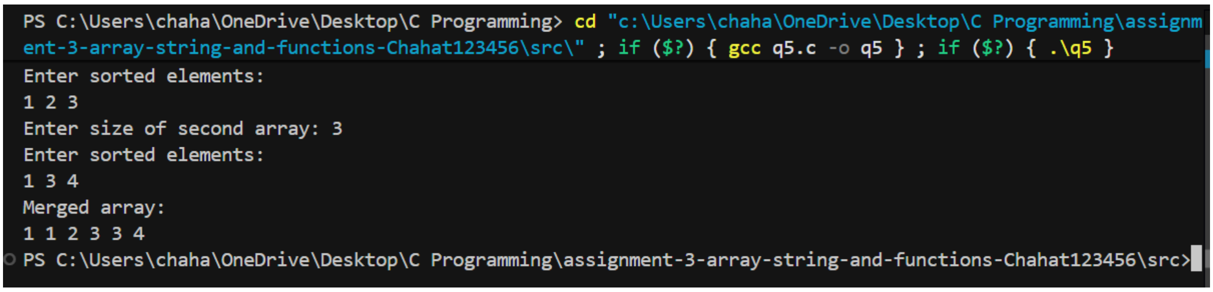


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming> cd "c:\Users\chaha\OneDrive\Desktop\C Programming\assignm
ent-3-array-string-and-functions-Chahat123456\src\" ; if ($?) { gcc q4.c -o q4 } ; if ($?) { .\q4 }
Enter size: 4
Enter elements:
223 3344 222
111
Second largest = 223
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

5. Write a C program to merge two sorted arrays into a single sorted array.

```
#include <stdio.h>

int main()
{
    int n1, n2, i = 0, j = 0, k = 0;

    printf("Enter size of first array: ");
    scanf("%d", &n1);
    int a[n1];
    printf("Enter sorted elements:\n");
    for (i = 0; i < n1; i++)
        scanf("%d", &a[i]);

    printf("Enter size of second array: ");
    scanf("%d", &n2);
    int b[n2];
    printf("Enter sorted elements:\n");
    for (i = 0; i < n2; i++)
        scanf("%d", &b[i]);

    int merged[n1 + n2];

    i = j = 0;
    while (i < n1  && j < n2)
        merged[k++] = (a[i] < b[j]) ? a[i++] : b[j++];

    while (i < n1)
        merged[k++] = a[i++];
    while (j < n2)
        merged[k++] = b[j++];

    printf("Merged array:\n");
    for (i = 0; i < n1 + n2; i++)
        printf("%d ", merged[i]);

    return 0;
}
```

Output

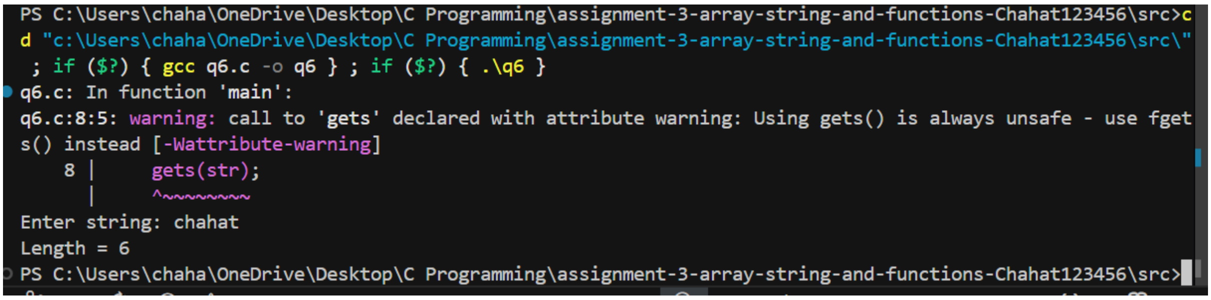


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming> cd "c:\Users\chaha\OneDrive\Desktop\C Programming\assignm
ent-3-array-string-and-functions-Chahat123456\src\" ; if ($?) { gcc q5.c -o q5 } ; if ($?) { .\q5 }
Enter sorted elements:
1 2 3
Enter size of second array: 3
Enter sorted elements:
1 3 4
Merged array:
1 1 2 3 3 4
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

6. **Write a C program to find the length of a string without using the built-in string functions.**

```
#include <stdio.h>
int main()
{
   char str[100];
   printf("Enter string: ");
   gets(str);

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

   printf("Length = %d", i);
   return 0;
}
```

Output



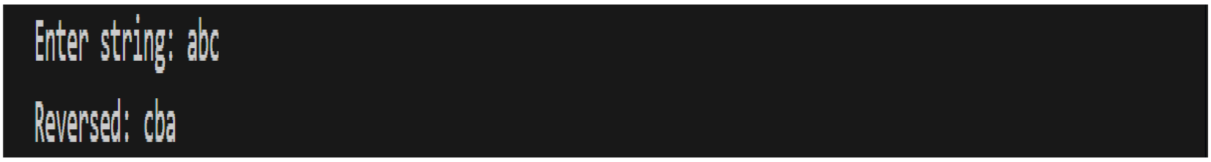
```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc q6.c -o q6 } ; if ($?) { .\q6 }
q6.c: In function 'main':
q6.c:8:5: warning: call to 'gets' declared with attribute warning: Using gets() is always unsafe - use fget
s() instead [-Wattribute-warning]
    8 |     gets(str);
      |     ^~~~~~~~~
Enter string: chahat
Length = 6
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

7. **Implement a C program to reverse a string.**

```
#include <stdio.h>

int main()
{
   char str[100];
   printf("Enter string: ");
   gets(str);

   int i = 0;
   while (str[i] != '\0')
```

```
        i++;

    printf("Reversed: ");
    for (i = i - 1; i >= 0; i--)
        printf("%c", str[i]);

    return 0;
}
```

Output

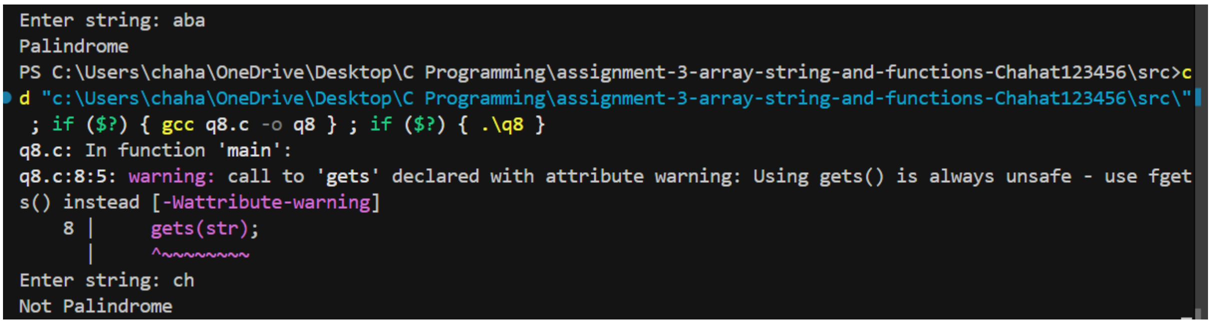


```
Enter string: abc
Reversed: cba
```

8. Write a C program to check if a given string is a palindrome.

```
#include <stdio.h>

int main()
{
    char str[100];
    printf("Enter string: ");
    gets(str);

    int i = 0, j;
    while (str[i] != '\0')
        i++;
    j = i - 1;

    int flag = 1;
    for (i = 0; i < j; i++, j--)
        if (str[i] != str[j])
            flag = 0;

    if (flag)
        printf("Palindrome");
    else
        printf("Not Palindrome");

    return 0;
}
```

Output

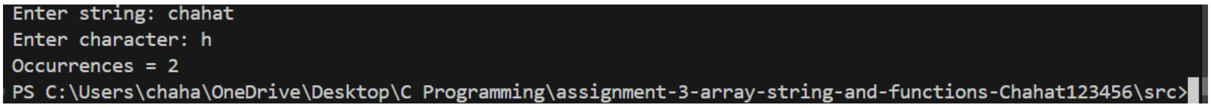


```
Enter string: aba
Palindrome
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc q8.c -o q8 } ; if ($?) { .\q8 }
q8.c: In function 'main':
q8.c:8:5: warning: call to 'gets' declared with attribute warning: Using gets() is always unsafe - use fget
s() instead [-Wattribute-warning]
    8 |     gets(str);
      |     ^~~~~~~~~
Enter string: ch
Not Palindrome
```

9. Implement a C program to count the occurrence of a specific character in a string.

```
#include <stdio.h>

int main()
{
   char str[100], ch;
   printf("Enter string: ");
   gets(str);

   printf("Enter character: ");
   scanf("%c", &ch);

   int count = 0;
   for (int i = 0; str[i] != '\0'; i++)
      if (str[i] == ch)
         count++;

   printf("Occurrences = %d", count);
   return 0;
}
```

Output



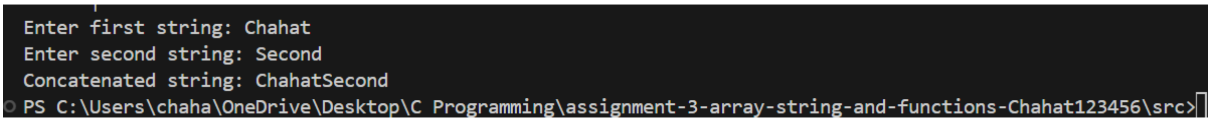
```
Enter string: chahat
Enter character: h
Occurrences = 2
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

10. Write a C program to concatenate two strings without using the built-in string functions.

```
#include <stdio.h>

int main()
{
   char a[100], b[100];
   printf("Enter first string: ");
   gets(a);
```

```
    printf("Enter second string: ");
    gets(b);

    int i = 0, j = 0;
    while (a[i] != '\0')
        i++;

    while (b[j] != '\0')
        a[i++] = b[j++];

    a[i] = '\0';

    printf("Concatenated string: %s", a);
    return 0;
}
```

Output

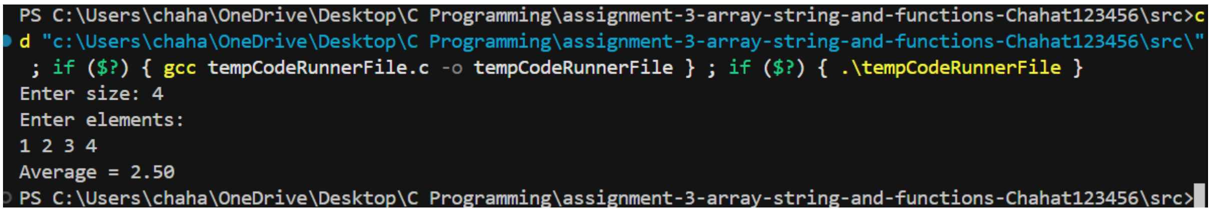


```
Enter first string: Chahat
Enter second string: Second
Concatenated string: ChahatSecond
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

11. Write a function named calculateAverage that takes an array of integers as input and returns the average of the numbers.

```
#include <stdio.h>

float calculateAverage(int arr[], int n)
{
    int sum = 0;
    for (int i = 0; i < n; i++)
        sum += arr[i];
    return (float)sum / n;
}

int main()
{
    int n;
    printf("Enter size: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter elements:\n");
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("Average = %.2f", calculateAverage(arr, n));
    return 0;
}
```

Output

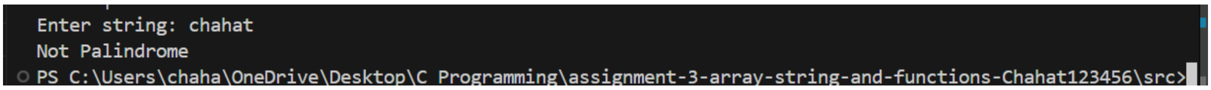


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter size: 4
Enter elements:
1 2 3 4
Average = 2.50
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

12. Write a function named isPalindrome that takes a string as input and returns 1 if it is a palindrome (reads the same forwards and backwards), and 0 otherwise.

```
#include <stdio.h>

int isPalindrome(char str[])
{
   int i = 0, j = 0;
   while (str[j] != '\0')
      j++;
   j--;

   while (i < j)
   {
      if (str[i] != str[j])
         return 0;
      i++;
      j--;
   }
   return 1;
}

int main()
{
   char str[100];
   printf("Enter string: ");
   gets(str);

   if (isPalindrome(str))
      printf("Palindrome");
   else
      printf("Not Palindrome");

   return 0;
}
```

Output

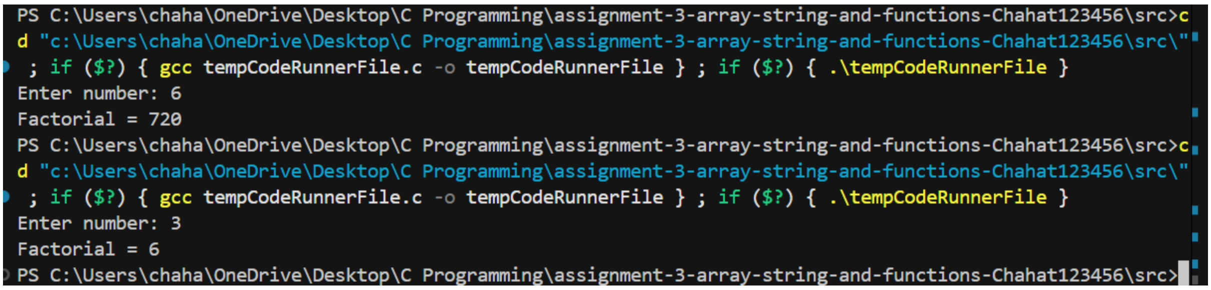


```
Enter string: chahat
Not Palindrome
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

13. Write a function named findFactorial that takes an integer as input and returns its factorial.

```
#include <stdio.h>

long long findFactorial(int n)
{
   long long fact = 1;
   for (int i = 1; i <= n; i++)
      fact *= i;
   return fact;
}

int main()
{
   int n;
   printf("Enter number: ");
   scanf("%d", &n);

   printf("Factorial = %lld", findFactorial(n));
   return 0;
}
```

Output

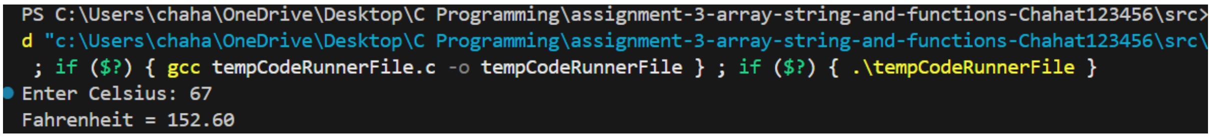


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter number: 6
Factorial = 720
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter number: 3
Factorial = 6
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

14. Write a function named convertTemperature that takes a temperature value in Celsius and converts it to Fahrenheit. The function should return the converted temperature.

```
#include <stdio.h>

float convertTemperature(float c)
{
   return (c * 9 / 5) + 32;
}

int main()
{
   float c;
   printf("Enter Celsius: ");
   scanf("%f", &c);

   printf("Fahrenheit = %.2f", convertTemperature(c));
   return 0;
}
```

Output

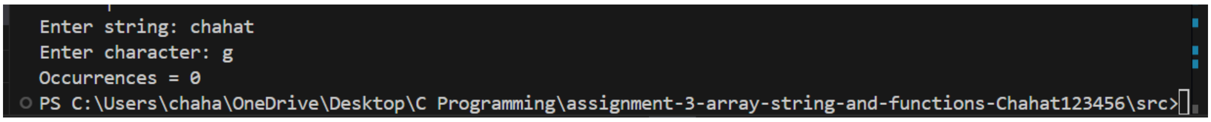


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter Celsius: 67
Fahrenheit = 152.60
```

15. Write a function named countOccurrences that takes a string and a character as input and returns the number of times the character appears in the string.

```
#include <stdio.h>

int countOccurrences(char str[], char ch)
{
    int count = 0;
    for (int i = 0; str[i] != '\0'; i++)
        if (str[i] == ch)
            count++;
    return count;
}

int main()
{
    char str[100], ch;
    printf("Enter string: ");
    gets(str);

    printf("Enter character: ");
    scanf("%c", &ch);

    printf("Occurrences = %d", countOccurrences(str, ch));
    return 0;
}
```

Output



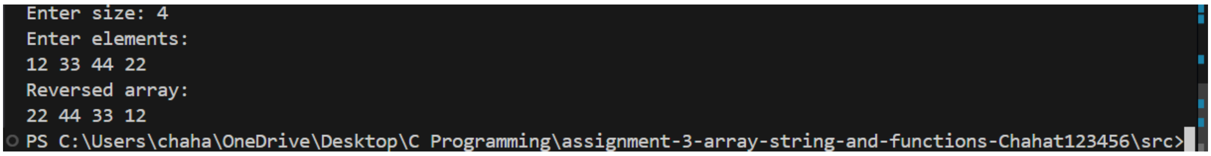
```
Enter string: chahat
Enter character: g
Occurrences = 0
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

16. Write a function named reverseArray that takes an array of integers as input and reverses the order of the elements in the array.

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

```
void reverseArray(int arr[], int n)
{
    int i = 0, j = n - 1, t;
    while (i < j)
    {
        t = arr[i];
        arr[i] = arr[j];
        arr[j] = t;
        i++;
        j--;
    }
}

int main()
{
    int n;
    printf("Enter size: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter elements:\n");
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    reverseArray(arr, n);

    printf("Reversed array:\n");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);

    return 0;
}
```

Output

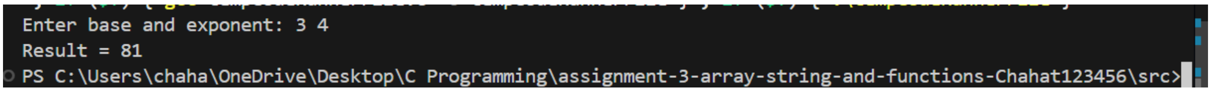


```
Enter size: 4
Enter elements:
12 33 44 22
Reversed array:
22 44 33 12
○ PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

17. **Write a function named calculatePower that takes two integers, base and exponent, as input and returns the result of raising the base to the exponent.**

```
#include <stdio.h>

long long calculatePower(int base, int exp)
{
    long long result = 1;
    for (int i = 0; i < exp; i++)
        result *= base;
    return result;
}

int main()
{
    int b, e;
    printf("Enter base and exponent: ");
    scanf("%d %d", &b, &e);

    printf("Result = %lld", calculatePower(b, e));
    return 0;
}
```

Output

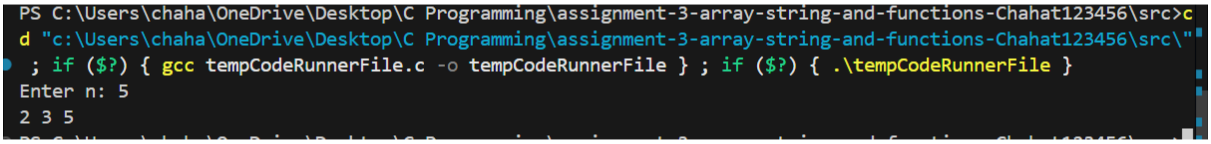


```
Enter base and exponent: 3 4
Result = 81
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

18. Write a function named findPrimeNumbers that takes an integer n as input and prints all prime numbers from 1 to n.

```
#include <stdio.h>

void findPrimeNumbers(int n)
{
    for (int i = 2; i <= n; i++)
    {
        int flag = 1;
        for (int j = 2; j <= i / 2; j++)
            if (i % j == 0)
            {
                flag = 0;
                break;
            }
        if (flag)
            printf("%d ", i);
    }
}

int main()
{
    int n;
    printf("Enter n: ");
    scanf("%d", &n);
    findPrimeNumbers(n);
    return 0;
}
```

Output

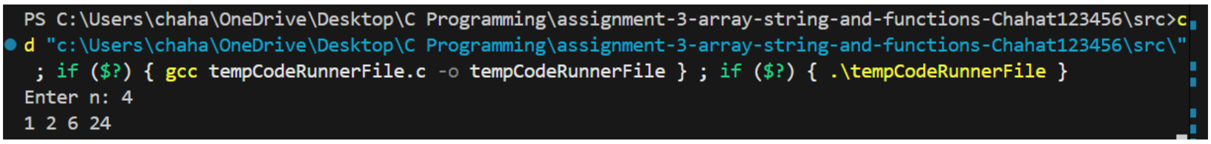


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter n: 5
2 3 5
```

19. Write a function named calculateFactorialSeries that takes an integer n as input and prints the factorial series up to n.

```
#include <stdio.h>

void calculateFactorialSeries(int n)
{
    long long fact = 1;
    for (int i = 1; i <= n; i++)
    {
        fact *= i;
        printf("%lld ", fact);
    }
}

int main()
{
    int n;
    printf("Enter n: ");
    scanf("%d", &n);

    calculateFactorialSeries(n);
    return 0;
}
```

Output

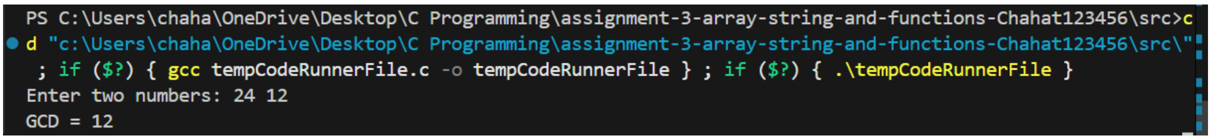


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter n: 4
1 2 6 24
```

20. Write a function named calculateGCD that takes two integers as input and returns their greatest common divisor (GCD).

```
#include <stdio.h>

int calculateGCD(int a, int b)
{
    while (b != 0)
    {
        int t = b;
        b = a % b;
        a = t;
    }
    return a;
}

int main()
{
    int a, b;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);

    printf("GCD = %d", calculateGCD(a, b));
    return 0;
}
```

Output

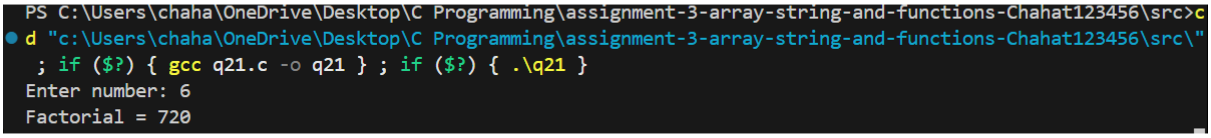


```
 PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
● d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
  ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
 Enter two numbers: 24 12
 GCD = 12
```

21. Write a recursive function named calculateFactorial that takes an integer n as input and returns its factorial.

```
#include <stdio.h>

long long calculateFactorial(int n)
{
    if (n == 0)
        return 1;
    return n * calculateFactorial(n - 1);
}

int main()
{
    int n;
    printf("Enter number: ");
    scanf("%d", &n);

    printf("Factorial = %lld", calculateFactorial(n));
    return 0;
}
```

Output

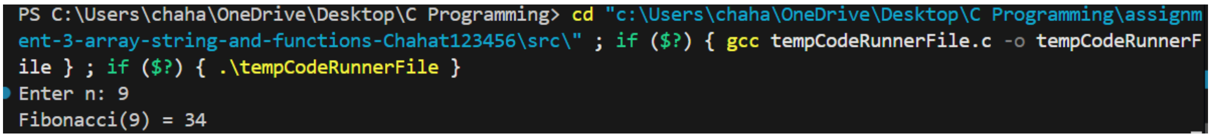


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc q21.c -o q21 } ; if ($?) { .\q21 }
Enter number: 6
Factorial = 720
```

22. Write a recursive function named calculateFibonacci that takes an integer n as input and returns the nth Fibonacci number. The Fibonacci sequence starts with 0 and 1, and each subsequent number is the sum of the two preceding numbers.

```
#include <stdio.h>

int calculateFibonacci(int n)
{
   if (n <= 1)
      return n;
   return calculateFibonacci(n - 1) + calculateFibonacci(n - 2);
}

int main()
{
   int n;
   printf("Enter n: ");
   scanf("%d", &n);

   printf("Fibonacci(%d) = %d", n, calculateFibonacci(n));
   return 0;
}
```

Output



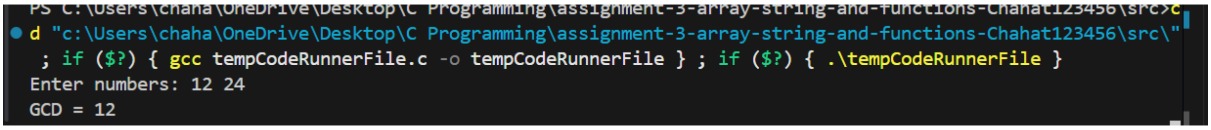
```
PS C:\Users\chaha\OneDrive\Desktop\C Programming> cd "c:\Users\chaha\OneDrive\Desktop\C Programming\assignm
ent-3-array-string-and-functions-Chahat123456\src\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerF
ile } ; if ($?) { .\tempCodeRunnerFile }
Enter n: 9
Fibonacci(9) = 34
```

23. Write a recursive function named calculateGCD that takes two integers a and b as input and returns their greatest common divisor (GCD).

```
#include <stdio.h>

int calculateGCD(int a, int b)
{
   if (b == 0)
      return a;
   return calculateGCD(b, a % b);
}

int main()
{
   int a, b;
   printf("Enter numbers: ");
```

```
    scanf("%d %d", &a, &b);

    printf("GCD = %d", calculateGCD(a, b));
    return 0;
}
```

Output

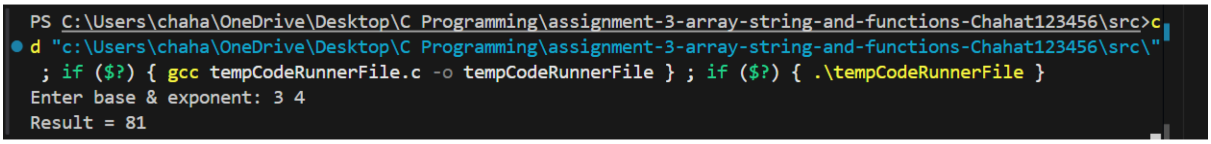


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter numbers: 12 24
GCD = 12
```

24. Write a recursive function named calculatePower that takes two integers base and exponent as input and returns the result of raising the base to the exponent.

```
#include <stdio.h>

long long calculatePower(int base, int exp)
{
    if (exp == 0)
        return 1;
    return base * calculatePower(base, exp - 1);
}

int main()
{
    int b, e;
    printf("Enter base & exponent: ");
    scanf("%d %d", &b, &e);

    printf("Result = %lld", calculatePower(b, e));
    return 0;
}
```

Output



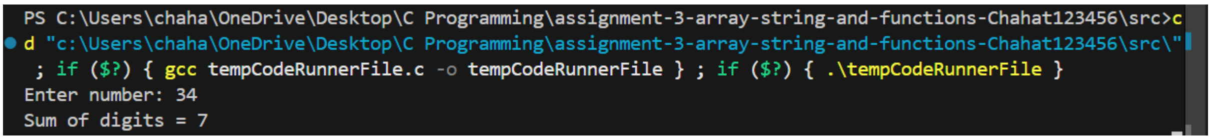
```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter base & exponent: 3 4
Result = 81
```

25. Write a recursive function named calculateSumOfDigits that takes an integer n as input and returns the sum of its digits.

```
#include <stdio.h>

int calculateSumOfDigits(int n)
{
    if (n == 0)
        return 0;
    return (n % 10) + calculateSumOfDigits(n / 10);
}
```

```
int main()
{
    int n;
    printf("Enter number: ");
    scanf("%d", &n);

    printf("Sum of digits = %d", calculateSumOfDigits(n));
    return 0;
}
```

Output

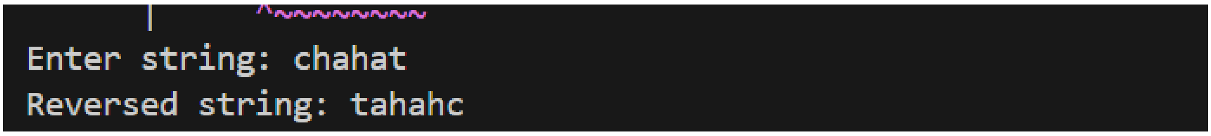


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter number: 34
Sum of digits = 7
```

26. Write a recursive function named reverseString that takes a string as input and returns the reversed string.

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

void reverseString(char str[], int start, int end)
{
    if (start >= end)
        return;

    char temp = str[start];
    str[start] = str[end];
    str[end] = temp;

    reverseString(str, start + 1, end - 1);
}

int main()
{
    char str[100];
    printf("Enter string: ");
    gets(str);

    reverseString(str, 0, strlen(str) - 1);

    printf("Reversed string: %s", str);
    return 0;
}
```

Output

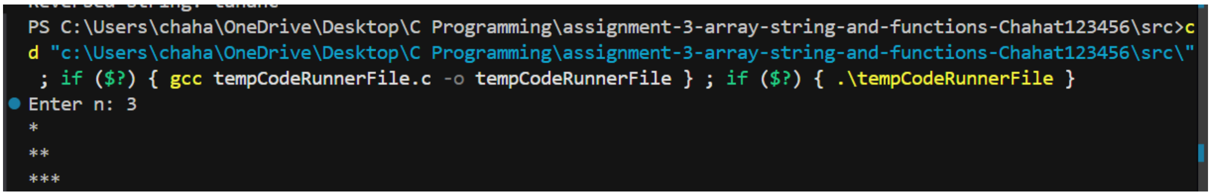


```
Enter string: chahat
Reversed string: tahahc
```

27. Write a recursive function named printTriangle that takes an integer n as input and prints a triangle of asterisks (*) with n rows.

```
#include <stdio.h>

void printTriangle(int n)
{
   if (n == 0)
      return;

   printTriangle(n - 1);

   for (int i = 0; i < n; i++)
      printf("*");
   printf("\n");
}

int main()
{
   int n;
   printf("Enter n: ");
   scanf("%d", &n);

   printTriangle(n);
   return 0;
}
```

Output



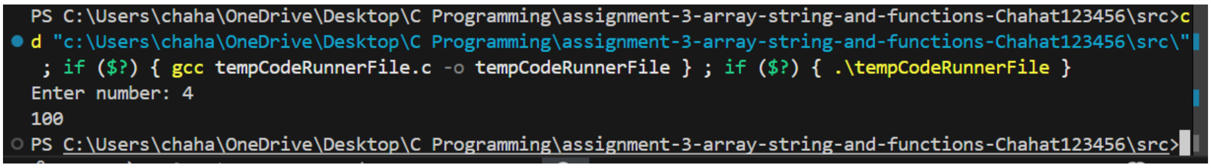
```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter n: 3
*
**
***
```

28. Write a recursive function named calculateBinary that takes an integer n as input and returns its binary representation as a string.

```
#include <stdio.h>

void calculateBinary(int n)
{
   if (n > 1)
      calculateBinary(n / 2);
```

```
    printf("%d", n % 2);
}

int main()
{
    int n;
    printf("Enter number: ");
    scanf("%d", &n);

    calculateBinary(n);
    return 0;
}
```

Output

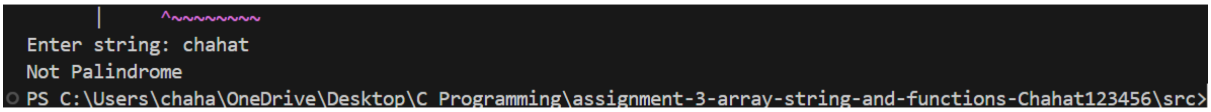


```
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>c
d "c:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src\"
 ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter number: 4
100
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

29. Write a recursive function named isPalindrome that takes a string as input and returns 1 if it is a palindrome (reads the same forwards and backwards), and 0 otherwise.

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

int isPalindrome(char str[], int start, int end)
{
    if (start >= end)
        return 1;
    if (str[start] != str[end])
        return 0;
    return isPalindrome(str, start + 1, end - 1);
}

int main()
{
    char str[100];
    printf("Enter string: ");
    gets(str);

    if (isPalindrome(str, 0, strlen(str) - 1))
        printf("Palindrome");
    else
        printf("Not Palindrome");
    return 0;
}
```

Output

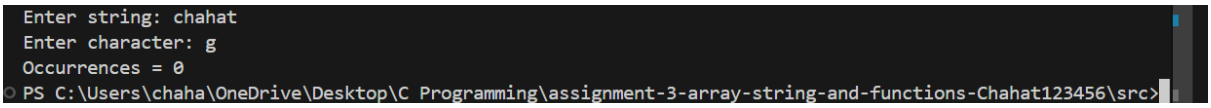


```
        |       ^~~~~~~~~
Enter string: chahat
Not Palindrome
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```

30. Write a recursive function named countOccurrences that takes a string and a character as input and returns the number of times the character appears in the string.

```
#include <stdio.h>

int countOccurrences(char str[], char ch, int i)
{
    if (str[i] == '\0')
        return 0;
    return (str[i] == ch) + countOccurrences(str, ch, i + 1);
}

int main()
{
    char str[100], ch;
    printf("Enter string: ");
    gets(str);

    printf("Enter character: ");
    scanf("%c", &ch);

    printf("Occurrences = %d", countOccurrences(str, ch, 0));
    return 0;
}
```

Output



```
Enter string: chahat
Enter character: g
Occurrences = 0
PS C:\Users\chaha\OneDrive\Desktop\C Programming\assignment-3-array-string-and-functions-Chahat123456\src>
```