

## **Assignment 3**

Angelina Chaudhary

Westcliff University

C Programming

Profession Thakuri

November 22, 2025

## Assignment 3

**Q1. Write a C program to find the maximum and minimum elements in an array.**

```
assignment-3-array-string-and-functions-angelinacy > src > C q1.c > main()
1 // Write a C program to find the maximum and minimum elements in an array.
2 #include <stdio.h>
3 int main() {
4     int n;
5     printf("(char [3])%d of elements: ");
6     scanf("%d", &n);
7     int arr[n];
8     printf("Enter the elements:\n");
9     for(int i = 0; i < n; i++) {
10         scanf("%d", &arr[i]);
11     }
12
13     int max = arr[0];
14     int min = arr[0];
15     for(int i = 1; i < n; i++) {
16         if(arr[i] > max)
17             max = arr[i];
18         if(arr[i] < min)
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
19             min = arr[i];
20     }
21     printf("Maximum element = %d\n", max);
22     printf("Minimum element = %d\n", min);
23     return 0;
24 }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

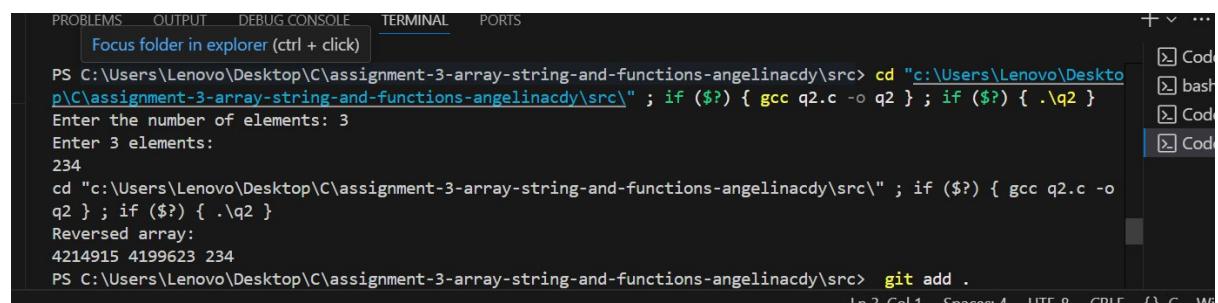
```
\C\assignment-3-array-string-and-functions-angelinacy\src\" ; if ($?) { gcc q1.c -o q1 } ; if ($
nter the number of elements: 3
nter the elements:
```

```
d "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src\" ; if ($?
1 } ; if ($?) { ./q1 }
aximum element = 4214915
inimum element = 2
```

```
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
[main 422716f] assignment completed
 2 files changed, 24 insertions(+), 1 deletion(-)
 create mode 100644 src/q1.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
```

## Q2. Implement a C program to reverse the elements of an array.

```
assignment-3-array-string-and-functions-angelinacy > src > C q2.c > main()
1 // Implement a C program to reverse the elements of an array.
2 #include <stdio.h>
3 int main() {
4     int n;
5     printf("Enter the number of elements: ");
6     scanf("%d", &n);
7     int arr[n];
8     printf("Enter %d elements:\n", n);
9     for (int i = 0; i < n; i++) {
10         scanf("%d", &arr[i]);
11     }
12     for (int i = 0; i < n / 2; i++) {
13         int temp = arr[i];
14         arr[i] = arr[n - 1 - i];
15         arr[n - 1 - i] = temp;
16     }
17     printf("Reversed array:\n");
18
19     for (int i = 0; i < n; i++) {
20         printf("%d ", arr[i]);
21     }
22 }
```



```

4214915 4199623 234
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
[main 96aeb8e] assignment completed
 2 files changed, 23 insertions(+), 1 deletion(-)
 create mode 100644 src/q2.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
Enumerating objects: 8, done.

Ln 3, Col 1  Spaces: 4  UTF-8  CRLF  {}  ⌂

```

**Q3. 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).**

```

C hello.c  ● C q1.c  ● C q2.c  ● C q3.c  X
assignment-3-array-string-and-functions-angelinacy > src > C q3.c > ...
1 // Write a C program to sort an array of integers in ascending order using a sorting algorithm of your choice
2
3 int main() {
4     int n;
5     printf("Enter the number of elements: ");
6     scanf("%d", &n);
7     int arr[n];
8     printf("Enter %d elements:\n", n);
9     for (int i = 0; i < n; i++) {
10         scanf("%d", &arr[i]);
11     }
12     for (int i = 0; i < n - 1; i++) {
13         for (int j = 0; j < n - i - 1; j++) {
14             if (arr[j] > arr[j + 1]) {
15                 int temp = arr[j];
16                 arr[j] = arr[j + 1];
17                 arr[j + 1] = temp;
18             }
19         }
20     }
21     int printf()
22     printf("Sorted array in ascending order:\n");
23     for (int i = 0; i < n; i++) {
24         printf("%d ", arr[i]);
25     }
26     return 0;
27 }

```

```

18 }
19
20     int printf()
21     printf("Sorted array in ascending order:\n");
22     for (int i = 0; i < n; i++) {
23         printf("%d ", arr[i]);
24     }
25     return 0;
26 }
27

```

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src"
PS C:\assignment-3-array-string-and-functions-angelinacdy\src> if ($?) { gcc q3.c -o q3 } ; if ($?) { ./q3 }
q3.c: In function 'main':
q3.c:5:5: warning: implicit declaration of function 'printf' [-Wimplicit-function-declaration]
    printf("Enter the number of elements: ");
    ^~~~~~
q3.c:5:5: warning: incompatible implicit declaration of built-in function 'printf'
q3.c:5:5: note: include '<stdio.h>' or provide a declaration of 'printf'
q3.c:6:5: warning: implicit declaration of function 'scanf' [-Wimplicit-function-declaration]
    scanf("%d", &n);
Ln 2, Col 1  Spaces: 4  UTF-8  CRLF  {}

```

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git add .
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
[main fc7d37a] assignment completed
 2 files changed, 26 insertions(+), 1 deletion(-)
 create mode 100644 src/q3.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads

```

#### Q4. Implement a C program to find the second largest element in an array.

```

C hello.c ● C q1.c C q2.c ● C q3.c C q4.c ●
assignment-3-array-string-and-functions-angelinacdy > src > C q4.c > main()
1 // Implement a C program to find the second largest element in an array.
2 #include <stdio.h>
3 int main() {
4     int n;
5     printf("Enter number of elements: ");
6     scanf("%d", &n);
7     if (n < 2) {
8         printf("At least two elements are required.\n");
9         return 0;
10    }
11    int arr[n];
12    printf("Enter %d elements:\n", n);
13    for (int i = 0; i < n; i++) {
14        scanf("%d", &arr[i]);
15    }
16    for (int i = 0; i < n - 1; i++) {
17        for (int j = 0; j < n - i - 1; j++) {

```

```

16     for (int i = 0; i < n - 1; i++) {
17         for (int j = 0; j < n - i - 1; j++) {
18             if (arr[j] > arr[j + 1]) {
19                 int temp = arr[j];
20                 arr[j] = arr[j + 1];
21                 arr[j + 1] = temp;
22             }
23         }
24     }
25     printf("Second largest element: %d\n", arr[n - 2]);
26     return 0;
27 }

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```

pC\assignment-3-array-string-and-functions-angelinacy\src\" ; if ($?) { gcc q4.c -o q4 } ; if (?) { .\q4 }
Enter number of elements: 2
Enter 2 elements:
12
cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src\" ; if (?) { gcc q4.c -o
q4 } ; if (?) { .\q4 }
Second largest element: 12
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment
completed"

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment
completed"
[main 400f456] assignment completed
2 files changed, 27 insertions(+), 1 deletion(-)
 create mode 100644 src/q4.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads

```

Line 27 Col 2    Source: 4    UTF-8    CPU

**Q5. Write a C program to merge two sorted arrays into a single sorted array.**

```

C hello.c ● C q1.c C q2.c ● C q3.c C q4.c ● C q5.c 1 ●
assignment-3-array-string-and-functions-angelinacdy > src > C q5.c > ...
1 // Write a C program to merge two sorted arrays into a single sorted array.
2 #include <stdio.h>
3 int main() {
4     int n1, n2;
5     int a[50], b[50], c[100];
6     printf("Enter size of first sorted array: ");
7     scanf("%d", &n1);
8     printf("Enter size of second sorted array: ");
9     scanf("%d", &n2);
10    printf("Enter %d elements of first sorted array:\n", n1);
11    for (int i = 0; i < n1; i++) {
12        scanf("%d", &a[i]);
13    }
14    printf("Enter %d elements of second sorted array:\n", n2);
15    for (int i = 0; i < n2; i++) {
16        scanf("%d", &b[i]);
17    }

```

```

18 int i = 0, j = 0, k = 0;2
19 while (i < n1 && j < n2) {
20     if [a[i] < b[j]]
21         c[k++] = a[i++];
22     else
23         c[k++] = b[j++];
24 }
25
26     while (i < n1) c[k++] = a[i++];
27     while (j < n2) c[k++] = b[j++];
28
29     printf("Merged sorted array:\n");
30     for (int x = 0; x < n1 + n2; x++) {
31         printf("%d ", c[x]);
32     }
33

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

1     while (i < n1 && j < n2) {
2
3         while (i < n1) c[k++] = a[i++];
4         while (j < n2) c[k++] = b[j++];
5
6
7         printf("Merged sorted array:\n");
8         for (int x = 0; x < n1 + n2; x++) {
9             printf("%d ", c[x]);
10            }
11
12        printf("\n");
13
14    }
15
16
17

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src\" ; if ($?) { gcc q5.c -o q5 } ; if (?) { .\q5 }
Enter size of first sorted array: 2
Enter size of second sorted array: 1
Enter 2 elements of first sorted array:
12
cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src\" ; if ($?) { gcc q5.c -o q5 } ; if (?) { .\q5 }
Enter 1 elements of second sorted array:
Merged sorted array:

```

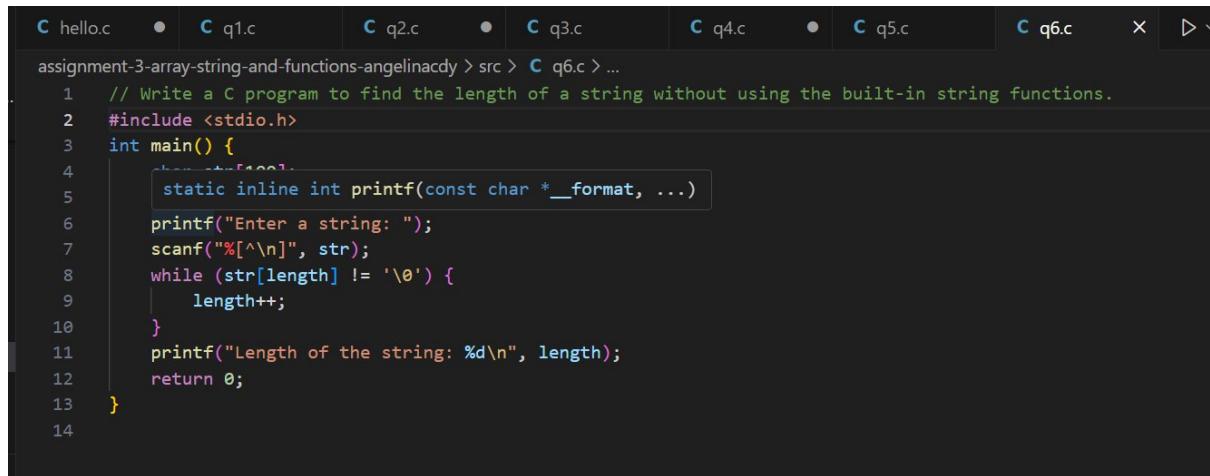
Ln 20. Col 25 Spaces: 4 UTF-8 CRLF

```

12 9437184 1974634114
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
[main 4d5aa63] assignment completed
2 files changed, 37 insertions(+), 1 deletion(-)
create mode 100644 src/q5.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.

```

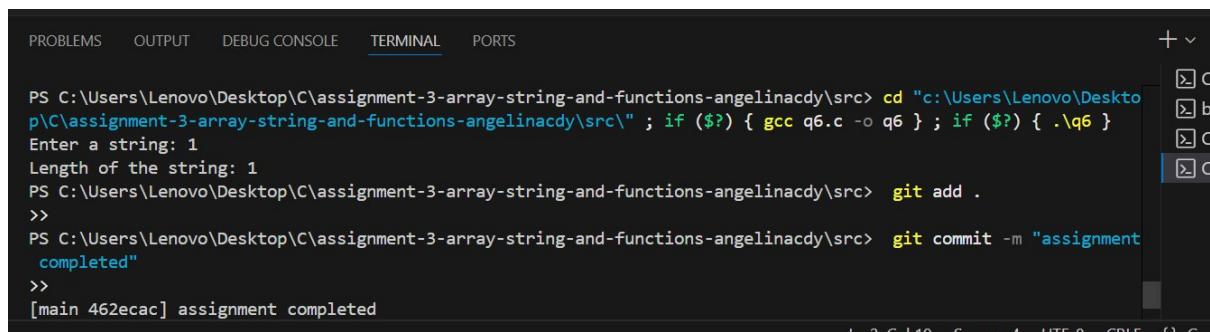
**Q6. Write a C program to find the length of a string without using the built-in string functions.**



```

C hello.c ● C q1.c C q2.c ● C q3.c C q4.c ● C q5.c C q6.c X ▶
assignment-3-array-string-and-functions-angelinacy > src > C q6.c > ...
1 // Write a C program to find the length of a string without using the built-in string functions.
2 #include <stdio.h>
3 int main() {
4     static inline int printf(const char * __format, ...)
5     printf("Enter a string: ");
6     scanf("%[^\\n]", str);
7     while (str[length] != '\\0') {
8         length++;
9     }
10    printf("Length of the string: %d\\n", length);
11    return 0;
12 }
13
14

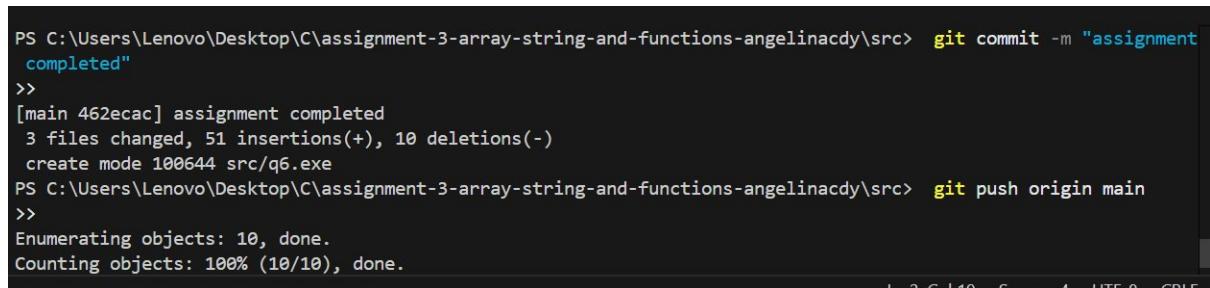
```



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + ▾
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src" ; if ($?) { gcc q6.c -o q6 } ; if ($?) { ./q6 }
Enter a string: 1
Length of the string: 1
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
[main 462ecac] assignment completed

```



```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
[main 462ecac] assignment completed
3 files changed, 51 insertions(+), 10 deletions(-)
create mode 100644 src/q6.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
>>
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.

```

## Q7. Implement a C program to reverse a string.

```

C q1.c      C q2.c      C q3.c      C q4.c      C q5.c
assignment-3-array-string-and-functions-angelinacy > src > C q7.c > ...
1 // Implement a C program to reverse a string.
2 #include <stdio.h>
3 #include <string.h>
4 int main() {
5     char str[100];
6     int leng (char [17])"Enter a string: "
7     printf("Enter a string: ");
8     scanf("%[^\\n]", str);
9     while (str[length] != '\0') {
10         length++;
11     }
12     for (int i = 0; i < length / 2; i++) {
13         char temp = str[i];
14         str[i] = str[length - 1 - i];
15         str[length - 1 - i] = temp;
16     }
17     printf("Reversed string: %s\\n", str);
18     return 0;

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

P\assignment-3-array-string-and-functions-angelinacy\src" ; if ($?) { gcc q7.c -o q7 } ; if ($?) { ./q7 }
Ent
Rev Focus folder in explorer (ctrl + click)
PS C:\Users\Lenovo\Desktop\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
[main 3367de2] assignment completed
2 files changed, 19 insertions(+), 1 deletion(-)

```

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\Lenovo\Desktop\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
[main 3367de2] assignment completed
2 files changed, 19 insertions(+), 1 deletion(-)
create mode 100644 src/q7.exe
PS C:\Users\Lenovo\Desktop\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.

```

**Q8. Write a C program to check if a given string is a palindrome.**

C q2.c ● C q3.c C q4.c ● C q5.c C q6.c C q7.c

```
assignment-3-array-string-and-functions-angelinacdy > src > C q8.c > ...
1 // Write a C program to check if a given string is a palindrome.
2 #include <stdio.h>
3 int main() {
4     char str[100];
5     int length = 0, isPalindrome = 1;
6     printf("Enter a string: ");
7     scanf("%[^\\n]", str);
8     while (str[length] != '\\0') {
9         length++;
10    }
11    for (int i = 0; i < length / 2; i++) {
12        if (str[i] != str[length - 1 - i]) {
13            isPalindrome = 0;
14            break;
15        }
16    }
17    if (isPalindrome) {
18        printf("The string is a palindrome.\n");
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
if (str[i] != str[length - 1 - i]) {
    isPalindrome = 0;
    break;
}
if (isPalindrome) (char [29])"The string is a palindrome.\n"
    printf("The string is a palindrome.\n");
} else {
    printf("The string is not a palindrome.\n");
}
return 0;
}
MS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src\" ; if ($?) { gcc q8.c -o q8 } ; if ($?) { ./q8 }
Enter a string: 1
The string is a palindrome.
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>
[main 5b1d0d2] assignment completed

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>
[main 5b1d0d2] assignment completed
2 files changed, 1 insertion(+), 1 deletion(-)
cr Focus folder in explorer (ctrl + click)
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.

```

### Q9. Implement a C program to count the occurrence of a specific character in a string.

```

q3.c q4.c q5.c q6.c q7.c q8.c

assignment-3-array-string-and-functions-angelinacdy > src > C q9.c > ...
1 // Implement a C program to count the occurrence of a specific character in a string.
2 #include <stdio.h>
3 int main() {
4     char str[100];
5     char ch;
6     int count = 0;
7     printf("Enter a string: ");
8     scanf("%[^\\n]", str);
9     printf("Enter the character to count: ");
10    scanf(" %c", &ch);
11
12    for (int i = 0; str[i] != '\\0'; i++) {
13        if (str[i] == ch) {
14            count++;
15        }
16    }
17    printf("The character '%c' occurs %d time(s) in the string.\n", ch, count);
18    return 0;

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src\" ; if ($?) { gcc q9.c -o q9 } ; if ($?) { ./q9 }
Enter a string: 1
Enter the character to count: 1
The character '1' occurs 1 time(s) in the string.
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + × ...  
  
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinadcy\src> git commit -m "assignment completed"  
->  
[main 06361d2] assignment completed  
 2 files changed, 19 insertions(+), 1 deletion(-)  
 create mode 100644 src/q9.exe  
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinadcy\src> git push origin main  
->  
Enumerating objects: 8, done.  
Counting objects: 100% (8/8), done.
```

**Q10. Write a C program to concatenate two strings without using the built-in string functions.**

```
assignment-3-array-string-and-functions-angelinacy > src > C q10.c > ...
1 // Write a C program to concatenate two strings without using the built-in string functions.
2 #include <stdio.h>
3 int main() {
4     char str1[100], str2[100];
5     int i = 0, j = 0;
6     printf("Enter first string: ");
7     scanf("%[^\\n]", str1);
8     printf("\n");
9     int __cdecl scanf(const char *, ...)
10    scanf("%[^\\n]", str2);
11    while (str1[i] != '\\0') {
12        i++;
13    }
14    while (str2[j] != '\\0') {
15        str1[i] = str2[j];
16        i++;
17        j++;
18    }
}
```

```
13     }
14     while (str2[j] != '\0') {
15         str1[i] = str2[j];
16         i++;
17         j++;
18     }
19     str1[i] = '\0';
20     printf("Concatenated string: %s\n", str1);
21     return 0;
22 }
23
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + ... [Code] [bash] [Code] [Code]

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src" ; if ($?) { gcc q10.c -o q10 } ; if (?) { .\q10 }

Enter first string: 1
Enter second string: 2
Concatenated string: 12
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} C Win?

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + ... [Code] [Code] [Code] [Code]

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
[main e64dd5f] assignment completed
2 files changed, 22 insertions(+), 1 deletion(-)
create mode 100644 src/q10.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {}

```

**Q11. Write a function named calculate Average that takes an array of integers as input and returns the average of the numbers.**

```

q5.c q6.c q7.c q8.c q9.c ● q10.c q11.c X ▶ ...
assignment-3-array-string-and-functions-angelinacy > src > C q11.c > ...
1 // Write a function named calculateAverage that takes an array of integers as input and returns the average
2 #include <stdio.h>
3 double calculateAverage(int arr[], int n) {
4     int sum = 0;
5     for (int i = 0; i < n; i++) {
6         sum += arr[i];
7     }
8     return (double)sum / n;
9 }
10 int main() {
11     int n; (char [31])"Enter the number of elements: "
12     printf("Enter the number of elements: ");
13     scanf("%d", &n);
14     if (n <= 0) {
15         printf("Array size must be positive.\n");
16     }
17 }

```

```
15     return 0;
16 }
17
18 int arr[n];
19 printf("Enter %d elements:\n", n);
20 for (int i = 0; i < n; i++) {
21     scanf("%d", &arr[i]);
22 }
23 double avg = calculateAverage(arr, n);
24 printf("Average of the array: %.1lf\n", avg);
25
26 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + ⌂ ⌂ ⌂ ⌂ ⌂

```
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src" ; if ($?) { gcc q11.c -o q11 } ; if ($?) { ./q11 }

Enter the number of elements: 2
Enter 2 elements:
12
cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src" ; if ($?) { gcc q11.c -o q11 } ; if ($?) { ./q11 }
Average of the array: 2107463.50
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS +  
  
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"  
>>  
[main fcfdfd0] assignment completed  
 2 files changed, 26 insertions(+), 1 deletion(-)  
 create mode 100644 src/q11.exe  
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main  
>>  
Enumerating objects: 8, done.  
Counting objects: 100% (8/8), done.
```

**Q12.** Write a function named `is Palindrome` that takes a string as input and returns 1 if it is a palindrome (reads the same forwards and backwards), and 0 otherwise.

```
assignment-3-array-string-and-functions-angelinacy > src > C q12.c > ...
1 // Write a function named isPalindrome that takes a string as input and returns 1 if it is a palindrome (...
2 #include <stdio.h>
3 int isPalindrome(char str[]) {
4     int length = 0;
5     while (str[length] != '\0') {
6         length++;
7     }
8     for (int i = 0; i < length / 2; i++) {
9         if (str[i] != str[length - 1 - i]) {
10             return 0;
11         }
12     }
13     return 1;
14 }
15 int main() {
16     char str[100];
17     printf("Enter a string: ");
```

```

16     char str[100];
17     printf("Enter a string: ");
18     scanf("%[^\\n]", str);
19     if (isPalindrome(str)) {
20         printf("The string is a palindrome.\n");
21     } else {
22         printf("The string is not a palindrome.\n");
23     }
24     return 0;
25 }
26

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    **TERMINAL**    PORTS

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src\" ; if ($?) { gcc q12.c -o q12 } ; if ($?) { ./q12 }

Enter a string: 1
The string is a palindrome.
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>

```

In 2 Col 19 Spaces: 4 UFT-8 CR LF {} C

PROBLEMS    OUTPUT    DEBUG CONSOLE    **TERMINAL**    PORTS

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>
[main f392bc9] assignment completed
2 files changed, 25 insertions(+), 1 deletion(-)
create mode 100644 src/q12.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.

```

In 2 Col 19 Spaces: 4 UFT-8 CR LF {} C\_Win

**Q13. Write a function named findFactorial that takes an integer as input and returns its factorial.**

```

C q7.c   C q8.c   C q9.c   ● C q10.c   C q11.c   C q12.c   C q13.c   X ▶
assignment-3-array-string-and-functions-angelinacdy > src > C q13.c > ...
1 // Write a function named findFactorial that takes an integer as input and returns its factorial.
2 #include <stdio.h>
3 unsigned long long findFactorial(int n) {
4     if (n < 0) {
5         return 0;
6     }
7     unsigned long long fact = 1;
8     for (int i = 1; i <= n; i++) {
9         fact *= i;
10    }
11    return fact;
12 }
13 int main() {
14     int num;
15     printf("Enter a non-negative integer: ");
16     scanf("%d", &num);
17     if (num < 0) {
18         printf("Factorial is not defined for negative numbers.\n");
19     } else {
20         unsigned long long result = findFactorial(num);
21         printf("Factorial of %d is %llu\n", num, result);
22     }
23     return 0;
24 }
25
26

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src\" ; if ($?) { gcc q13.c -o q13 } ; if ($?) { .\q13 }

Enter a non-negative integer: 1
Factorial of 1 is 1
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>
[main f70a4e0] assignment completed
2 files changed, 25 insertions(+), 1 deletion(-)
create mode 100644 src/q13.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
```

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} C Wi

**Q14. Write a function named convert Temperature that takes a temperature value in Celsius and converts it to Fahrenheit. The function should return the converted temperature.**

```

C q8.c      C q9.c      ● C q10.c      C q11.c      C q12.c      C q13.c      C q14.c
assignment-3-array-string-and-functions-angelinacdy > src > C q14.c > ...
1 // Write a function named convertTemperature that takes a temperature value in Celsius and conver...
2 #include <stdio.h>
3 double convertTemperature(double celsius) {
4     return (celsius * 9 / 5) + 32;
5 }
6 int main() {
7     double celsius;
8     printf("Enter temperature in Celsius: ");
9     scanf("%lf", &celsius);
10    double fahrenheit = convertTemperature(celsius);
11    printf("%.2lf Celsius = %.2lf Fahrenheit\n", celsius, fahrenheit);
12    return 0;
13 }
14

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> cd "c:/Users/Lenovo/Desktop/C/assignment-3-array-string-and-functions-angelinacdy/src"
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> if ($?) { gcc q14.c -o q14 } ; if ($?) { ./q14 }

Enter temperature in Celsius: 2
2.00 Celsius = 35.60 Fahrenheit
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>

```

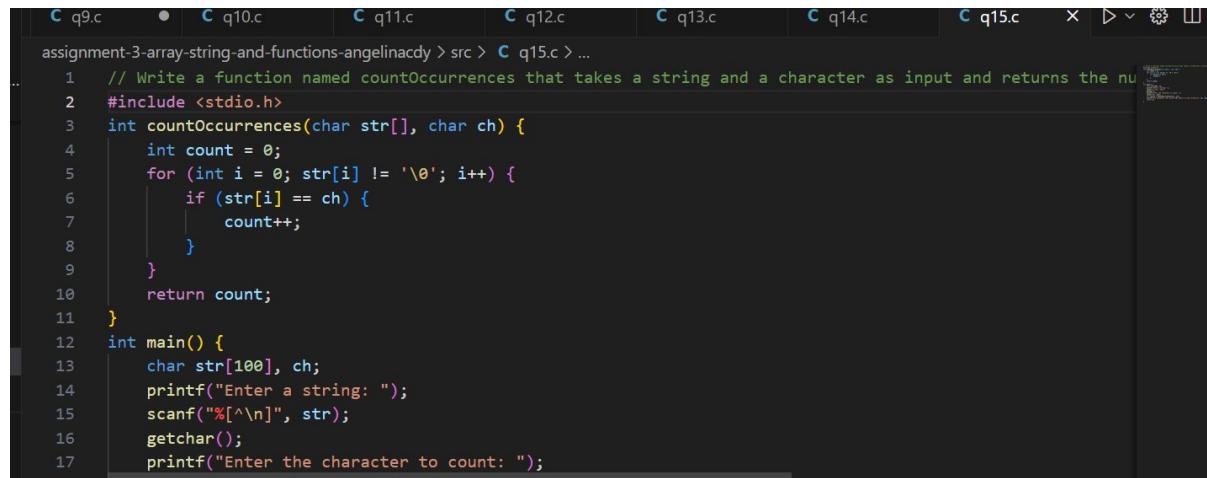
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git push origin main
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.

```

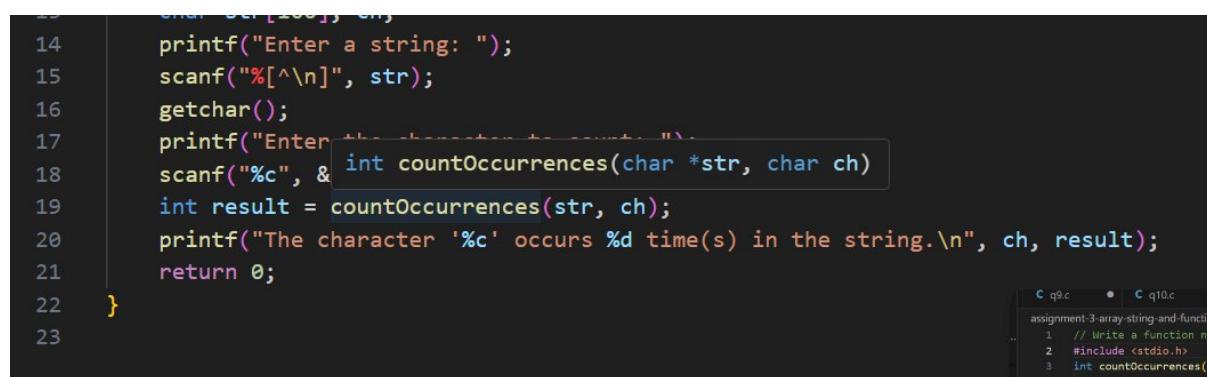
**Q15. Write a function named count Occurrences that takes a string and a character as input and returns the number of times the character appears in the string.**



```

1 // Write a function named countOccurrences that takes a string and a character as input and returns the number of occurrences of the character in the string.
2 #include <stdio.h>
3 int countOccurrences(char str[], char ch) {
4     int count = 0;
5     for (int i = 0; str[i] != '\0'; i++) {
6         if (str[i] == ch) {
7             count++;
8         }
9     }
10    return count;
11 }
12 int main() {
13     char str[100], ch;
14     printf("Enter a string: ");
15     scanf("%[^\\n]", str);
16     getchar();
17     printf("Enter the character to count: ");
18     scanf("%c", &ch);
19     int result = countOccurrences(str, ch);
20     printf("The character '%c' occurs %d time(s) in the string.\n", ch, result);
21     return 0;
22 }
23

```



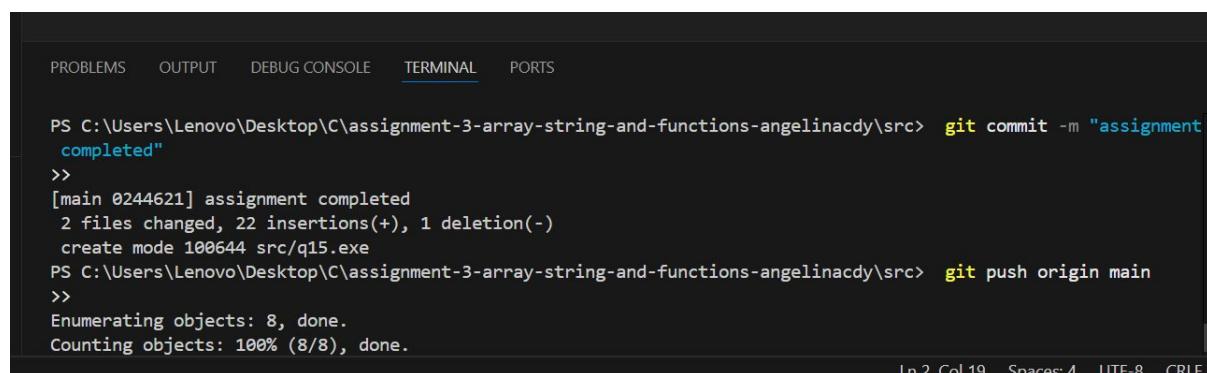

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src\" ; if ($?) { gcc q15.c -o q15 } ; if ($?) { ./q15 }

Enter a string: 1
Enter the character to count: 2
The character '2' occurs 0 time(s) in the string.
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"

```

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} C



```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
[main 0244621] assignment completed
2 files changed, 22 insertions(+), 1 deletion(-)
create mode 100644 src/q15.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.

```

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF

**Q16. Write a function named reverse Array that takes an array of integers as input and reverses the order of the elements in the array.**

```
assignment-3-array-string-and-functions-angelinacdy > src > C q16.c > ...
1 // Write a function named reverseArray that takes an array of integers as input and reverses the order of
2 #include <stdio.h>
3 void reverseArray(int arr[], int n) {
4     int temp;
5     for (int i = 0; i < n / 2; i++) {
6         temp = arr[i];
7         arr[i] = arr[n - 1 - i];
8         arr[n - 1 - i] = temp;
9     }
10 }
11 int main() {
12     int n;
13     printf("Enter number of elements: ");
14     scanf("%d", &n);
15     if (n <= 0) {
16         printf("Array size must be positive.\n");
17         return 0;
18     }
19     int arr[n];
20     printf("Enter %d elements:\n", n);
21     for (int i = 0; i < n; i++) {
22         scanf("%d", &arr[i]);
23     }
24     reverseArray(arr, n);
25     printf("Reversed array:\n");
26     for (int i = 0; i < n; i++) {
27         printf("%d ", arr[i]);
28     }
29     printf("\n");
30     return 0;
31 }
```

```
17     return 0;
18 }
19     int arr[n];
20     printf("Enter %d elements:\n", n);
21     for (int i = 0; i < n; i++) {
22         scanf("%d", &arr[i]);
23     }
24     reverseArray(arr, n);
25     printf("Reversed array:\n");
26     for (int i = 0; i < n; i++) {
27         printf("%d ", arr[i]);
28     }
29     printf("\n");
30     return 0;
31 }
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src\" ; if (?) { gcc q16.c -o q16 } ; if (?) { .\q16 }

Enter number of elements: 2
Enter 2 elements:
12
cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src\" ; if (?) { gcc q16.c -o q16 } ; if (?) { .\q16 }
Reversed array:
4199777 12
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS +  
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .  
>>  
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"  
>>  
[main 03f9a54] assignment completed  
2 files changed, 31 insertions(+), 1 deletion(-)  
create mode 100644 src/q16.exe  
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main  
>>
```

```
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"  
>>  
[main 03f9a54] assignment completed  
2 files changed, 31 insertions(+), 1 deletion(-)  
create mode 100644 src/q16.exe  
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main  
>>  
Enumerating objects: 8, done.  
Counting objects: 100% (8/8), done.
```

In 2 Col 19 Spaces: 4 UTE-8 CR LF

**Q17. Write a function named calculate Power that takes two integers, base and exponent, as input and returns the result of raising the base to the exponent.**

```
C q11.c C q12.c C q13.c C q14.c C q15.c C q16.c C q17.c X D v  
assignment-3-array-string-and-functions-angelinacy > src > C q17.c > ...  
1 // Write a function named calculatePower that takes two integers, base and exponent, as input and returns  
2 #include <stdio.h>  
3 long long calculatePower(int base, int exponent) {  
4     long long result = 1;  
5     if (exponent < 0) {  
6         printf("Negative exponents are not supported in this integer version.\n");  
7         return 0;  
8     }  
9     for (int i = 0; i < exponent; i++) {  
10         result *= base;  
11     }  
12     return result;  
13 }  
14 int main() {  
15     int base, exponent;  
16     printf("Enter base: ");  
17     scanf("%d", &base);
```

```
18     printf("Enter exponent (non-negative integer): ");  
19     scanf("%d", &exponent);  
20     long long power = calculatePower(base, exponent);  
21     printf("%d raised to the power %d is: %lld\n", base, exponent, power);  
22     return 0;  
23 }  
24
```

Problems (Ctrl+Shift+M)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src" ; if ($?) { gcc q17.c -o q17 } ; if ($?) { ./q17 }

Enter base: 1
Enter exponent (non-negative integer): 2
1 raised to the power 2 is: 1
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .

>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
```

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} C W

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
[main 025e9a2] assignment completed
 2 files changed, 23 insertions(+), 1 deletion(-)
  create mode 100644 src/q17.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
```

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} C W

**Q18. Write a function named find Prime Numbers that takes an integer n as input and prints all prime numbers from 1 to n.**

```
assignment-3-array-string-and-functions-angelinacy > src > C q18.c > ...
1 // Write a function named findPrimeNumbers that takes an integer n as input and prints all prime numbers f
2 #include <stdio.h>
3 void findPrimeNumbers(int n) {
4     if (n < 2) {
5         printf("No prime numbers in this range.\n");
6         return;
7     }
8     printf("Prime numbers from 1 to %d:\n", n);
9
10    for (int i = 2; i <= n; i++) {
11        int isPrime = 1;
12        for (int j = 2; j * j <= i; j++) {
13            if (i % j == 0) {
14                isPrime = 0;
15                break;
16            }
17        }
```

```
18
19     if (isPrime) {
20         printf("%d ", i);
21     }
22 }
23
24 printf("\n");
25 }
26
27 int main() {
28     int n;
29
30     printf("Enter a positive integer: ");
31     scanf("%d", &n);
```

```
28     int n;
29
30     printf("Enter a positive integer: ");
31     scanf("%d", &n);
32
33     findPrimeNumbers(n);
34
35     return 0;
36 }
37
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + ... ^ x

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src\" ; if ($?) { gcc q18.c -o q18 } ; if ($?) { ./q18 }

Enter a positive integer: 1
No prime numbers in this range.
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + × ⋮ ^

```
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacd\src> git commit -m "assignment completed"
>>
[main d3314b6] assignment completed
 2 files changed, 36 insertions(+), 1 deletion(-)
  create mode 100644 src/q18.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacd\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
```

**Q19.** Write a function named calculate Factorial Series that takes an integer n as input and prints the factorial series up to n.

```
q13.c q14.c q15.c q16.c q17.c q18.c q19.c X D v E S M ...  
assignment-3-array-string-and-functions-angelinacyd > src > C q19.c > ...  
1 // Write a function named calculateFactorialSeries that takes an integer n as input and prints the factor  
2 #include <stdio.h>  
3 void calculateFactorialSeries(int n) {  
4     if (n < 0) {  
5         printf("Factorial is not defined for negative numbers.\n");  
6         return;  
7     }  
8     unsigned long long fact = 1;  
9     printf("Factorial series up to %d:\n", n);  
10    for (int i = 0; i <= n; i++) {  
11        if (i == 0) {  
12            fact = 1; // 0! = 1  
13        } else {  
14            fact *= i;  
15        }  
16        printf("%d! = %llu\n", i, fact);  
17    }  
18 }  
19  
20 int main() {  
21     int n;  
22     printf("Enter a non-negative integer: ");  
23     scanf("%d", &n);  
24     calculateFactorialSeries(n);  
25     return 0;  
26 }  
27
```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinadcy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinadcy\src" ; if (?) { gcc q19.c -o q19 } ; if (?) { .\q19 }

Enter a non-negative integer: 1  
Factorial series up to 1:  
0! = 1  
1! = 1

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinadcy\src> git add .  
>>  
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinadcy\src> git commit -m "assignment

```
completed"
>>
[main 5a2ed5f] assignment completed
 2 files changed, 26 insertions(+), 1 deletion(-)
 create mode 100644 src/q19.exe
PS C:\Users\Lenovo\Desktop\assignment-3-array-string-and-functions-angelinacdy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
```

**Q20.** Write a function named calculate GCD that takes two integers as input and returns their greatest common divisor (GCD).

```

q14.c q15.c q16.c q17.c q18.c q19.c q20.c ...
assignment-3-array-string-and-functions-angelinacdy > src > C q20.c > ...
1 // Write a function named calculateGCD that takes two integers as input and returns their greatest common
2 #include <stdio.h>
3 int calculateGCD(int a, int b) {
4     while (b != 0) {
5         int temp = b;
6         b = a % b;
7         a = temp;
8     }
9     return a;
10 }
11 int main() {
12     int num1, num2;
13     printf("Enter two integers: ");
14     scanf("%d %d", &num1, &num2);
15     if (num1 == 0 && num2 == 0) {
16         printf("GCD is not defined for both numbers being zero.\n");
17     } else {
18         int gcd = calculateGCD(num1, num2);
19         printf("GCD of %d and %d is: %d\n", num1, num2, gcd);
20     }
21     return 0;
22 }
23

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src\" ; if ($?) { gcc q20.c -o q20 } ; if ($?) { ./q20 }

Enter two integers: 2
12
GCD of 2 and 12 is: 2
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"

```

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} C W

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>
[main a0d75ea] assignment completed
2 files changed, 22 insertions(+), 1 deletion(-)
create mode 100644 src/q20.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.

```

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {}

**Q21. Write a recursive function named calculateFactorial that takes an integer n as input and returns its factorial.**

```

C q15.c   C q16.c   C q17.c   C q18.c   C q19.c   C q20.c   C q21.c   X D 
assignment-3-array-string-and-functions-angelinacdy > src > C q21.c > ...
1 // Write a recursive function named calculateFactorial that takes an integer n as input and returns its fa
2 #include <stdio.h>
3 unsigned long long calculateFactorial(int n) {
4     if (n < 0) {
5         return 0;
6     }
7     if (n == 0 || n == 1) {
8         return 1;
9     }
10    return n * calculateFactorial(n - 1);
11 }
12
13 int main() {
14     int num;
15
16     printf("Enter a non-negative integer: ");
17     scanf("%d", &num);

```

```

18
19     if (num < 0) {
20         printf("Factorial is not defined for negative numbers.\n");
21     } else {
22         unsigned long long result = calculateFactorial(num);
23         printf("Factorial of %d is: %llu\n", num, result);
24     }
25
26     return 0;
27 }
28

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + ... 
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src" ; if ($?) { gcc q21.c -o q21 } ; if ($?) { ./q21 }
W Enter a non-negative integer: 1
CS Factorial of 1 is: 1
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git add .
KS >>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>
Ln 2, Col 19  Spaces: 4  UTF-8  CRLF  {} C  Win3

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

completed"
>>
[main e43e145] assignment completed
  2 files changed, 27 insertions(+), 1 deletion(-)
  create mode 100644 src/q21.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads

```

Ln 2, Col 19 Spaces: 4 UTF-8

**Q22.** Write a recursive function named calculate Fibonacci 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.

```

assignment-3-array-string-and-functions-angelinacdy > src > C q22.c > main()
1 // Write a recursive function named calculateFibonacci that takes an integer n as input and returns th
2 #include <stdio.h>
3 unsigned long long calculateFibonacci(int n) {
4     if (n < 0) {
5         return 0;
6     }
7     if (n == 0) return 0;
8     if (n == 1) return 1;
9     return calculateFibonacci(n - 1) + calculateFibonacci(n - 2);
10 }
11 int main() {
12     int n;
13     printf("Enter a non-negative integer n: ");
14     scanf("%d", &n);
15
16     if (n < 0) {
17         printf("Invalid input. n must be non-negative.\n");
18     } else {
19         unsigned long long fib = calculateFibonacci(n);
20         printf("The %dth Fibonacci number is: %llu\n", n, fib);
21     }
22
23     return 0;
24 }
25

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src\" ; if ($?) { gcc q22.c -o q22 } ; if ($?) { .\q22 }

Enter a non-negative integer n: 1
The 1th Fibonacci number is: 1
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

completed"
>>
[main 6bf9656] assignment completed
 2 files changed, 24 insertions(+), 1 deletion(-)
 create mode 100644 src/q22.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads

```

Ln 12, Col 11 Spaces: 4 UTF-8 CRLF {} C

**Q23. Write a recursive function named calculate GCD that takes two integers a and b as input and returns their greatest common divisor (GCD).**

```

assignment-3-array-string-and-functions-angelinacy > src > C q23.c > ...
1 // Write a recursive function named calculateGCD that takes two integers a and b as input and returns their
2 #include <stdio.h>
3 int calculateGCD(int a, int b) {
4     if (b == 0)
5         return a;
6     return calculateGCD(b, a % b);
7 }
8
9 int main() {
10     int num1, num2;
11
12     printf("Enter two integers: ");
13     scanf("%d %d", &num1, &num2);
14
15     if (num1 == 0 && num2 == 0) {
16         printf("GCD is not defined for both numbers being zero.\n");
17     } else {
18
19         int gcd = calculateGCD(num1, num2);
20         printf("GCD of %d and %d is: %d\n", num1, num2, gcd);
21
22     }
23 }
24

```

```

15     if (num1 == 0 && num2 == 0) {
16         printf("GCD is not defined for both numbers being zero.\n");
17     } else {
18         int gcd = calculateGCD(num1, num2);
19         printf("GCD of %d and %d is: %d\n", num1, num2, gcd);
20     }
21
22 }
23
24

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src\" ; if ($?) { gcc q23.c -o q23 } ; if ($?) { ./q23 }

Enter two integers: 12
cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src\" ; if ($?) { gcc q23.c -o q23 } ; if ($?) { ./q23 }
GCD of 12 and 12922584 is: 12
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> ^C

```

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} C

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>
[main d5c32aa] assignment completed
 2 files changed, 23 insertions(+), 1 deletion(-)
   create mode 100644 src/q23.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.

```

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF

**Q24. 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.**

```

assignment-3-array-string-and-functions-angelinacdy > src > C q24.c > ...
1 // Write a recursive function named calculatePower that takes two integers base and exponent as input and
2 #include <stdio.h>
3 long long calculatePower(int base, int exponent) {
4     if (exponent < 0) {
5         printf("Negative exponents are not supported in this integer version.\n");
6         return 0;
7     }
8     if (exponent == 0) {
9         return 1;
10    }
11    return base * calculatePower(base, exponent - 1);
12 }
13 int main() {
14     int base, exponent;
15     printf("Enter base: ");
16     scanf("%d", &base);
17     printf("Enter exponent (non-negative integer): ");
18     scanf("%d", &exponent);
19     long long power = calculatePower(base, exponent);
20     printf("%d raised to the power %d is: %lld\n", base, exponent, power);
21     return 0;
22 }
23

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src\" ; if ($?) { gcc q24.c -o q24 } ; if (?) { .\q24 }

Enter base: 1
Enter exponent (non-negative integer): 2
1 raised to the power 2 is: 1
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"

```

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
[main acfc96c] assignment completed
 2 files changed, 2 insertions(+), 6 deletions(-)
 create mode 100644 src/q24.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.

```

**Q25. Write a recursive function named calculate Sum Of Digits that takes an integer n**

**as input and returns the sum of its digits.**

```

assignment-3-array-string-and-functions-angelinacy > src > C q25.c > ...
1 // Write a recursive function named calculateSumOfDigits that takes an integer n as input and returns the
2 #include <stdio.h>
3 int calculateSumOfDigits(int n) {
4     if (n < 0) {
5         n = -n;
6     }
7     if (n == 0) {
8         return 0;
9     }
10    return (n % 10) + calculateSumOfDigits(n / 10);
11 }
12
13 int main() {
14     int num;
15
16     printf("Enter an integer: ");
17     scanf("%d", &num);
18

```

```

15     printf("Enter an integer: ");
16     scanf("%d", &num);
17
18     int sum = calculateSumOfDigits(num);
19     printf("Sum of digits of %d is: %d\n", num, sum);
20
21     return 0;
22 }
23
24

```

The screenshot shows a terminal window with the following content:

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src" ; if ($?) { gcc q25.c -o q25 } ; if ($?) { ./q25 }
Enter an integer: 2
Sum of digits of 2 is: 2
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>

```

Terminal status bar: Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} C Win32

The screenshot shows a terminal window with the following content:

```

>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
[main 1d52219] assignment completed
 2 files changed, 23 insertions(+), 1 deletion(-)
 create mode 100644 src/q25.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
>>
Enumerating objects: 8, done.

```

Terminal status bar: Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} C

**Q26. Write a recursive function named reverse String that takes a string as input and returns the reversed string.**

The screenshot shows a code editor with the following C code:

```

assignment-3-array-string-and-functions-angelinacy > src > C q26.c > ...
1 // Write a recursive function named reverseString that takes a string as input and returns the reversed st
2 #include <stdio.h>
3 #include <string.h>
4 void reverseString(char str[], int start, int end) {
5     if (start >= end) {
6         return;
7     }
8     char temp = str[start];
9     str[start] = str[end];
10    str[end] = temp;
11    reverseString(str, start + 1, end - 1);
12 }
13 int main() {
14     char str[100];
15     printf("Enter a string: ");
16     scanf("%[^\\n]", str);
17     int length = 0;
18     while (str[length] != '\\n') f

```

```

18     while (str[length] != '\0') {
19         length++;
20     }
21     reverseString(str, 0, length - 1);
22     printf("Reversed string: %s\n", str);
23     return 0;
24 }
25

```

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src\" ; if ($?) { gcc q26.c -o q26 } ; if ($?) { ./q26 }

Enter a string: 1
Reversed string: 1
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>

```

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git commit -m "assignment completed"
>>
[main dc41eb3] assignment completed
 2 files changed, 24 insertions(+), 1 deletion(-)
 create mode 100644 src/q26.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacdy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.

```

**Q27. Write a recursive function named print Triangle that takes an integer n as input and prints a triangle of asterisks (\*) with n rows.**

```

assignment-3-array-string-and-functions-angelinacdy > src > C q27.c > ...
1 // Write a recursive function named printTriangle that takes an integer n as input and prints a triangle of asterisks (*)
2 #include <stdio.h>
3 void printTriangle(int n) {
4     if (n == 0) {
5         return;
6     }
7
8     printTriangle(n - 1);
9     for (int i = 0; i < n; i++) {
10         printf("*");
11     }
12     printf("\n");
13 }
14 int main() {
15     int n;
16     printf("Enter the number of rows: ");
17     scanf("%d", &n);
18     printTriangle(n);

```

```

14 int main() {
15     int n;
16     printf("Enter the number of rows: ");
17     scanf("%d", &n);
18     printTriangle(n);
19     return 0;
20 }
21

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src\" ; if (\$?) { gcc q27.c -o q27 } ; if (\$?) { .\q27 }

Enter the number of rows: 1

\* PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .

>>

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"

>>

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main

>>

Enumerating objects: 8, done.

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} C Win

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"

>>

[main 863c283] assignment completed

2 files changed, 20 insertions(+), 1 deletion(-)

create mode 100644 src/q27.exe

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main

>>

Enumerating objects: 8, done.

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} C Win

**Q28. Write a recursive function named calculate Binary that takes an integer n as input and returns its binary representation as a string.**

```

assignment-3-array-string-and-functions-angelinacy > src > C q28.c > ...
1 // Write a recursive function named calculateBinary that takes an integer n as input and returns its binary representation as a string.
2 #include <stdio.h>
3 #include <string.h>
4 void calculateBinary(int n) {
5     if (n == 0)
6         return;
7     calculateBinary(n / 2);
8     printf("%d", n % 2);
9 }
10 int main() {
11     int n;
12     printf("Enter a non-negative integer: ");
13     scanf("%d", &n);
14     if (n == 0) {
15         printf("Binary: 0\n");
16     } else {
17         printf("Binary: ");

```

```

14     if (n == 0) {
15         printf("Binary: 0\n");
16     } else {
17         printf("Binary: ");
18         calculateBinary(n);
19         printf("\n");
20     }
21     return 0;
22 }
23

```

The terminal window shows the following sequence of commands:

```

^
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src"
PS C:\assignment-3-array-string-and-functions-angelinacy\src> if ($?) { gcc q28.c -o q28 } ; if ($?) { ./q28 }

Enter a non-negative integer: 1
Binary: 1
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"

```

At the bottom right of the terminal window, there is a sidebar with several checkboxes labeled "Code", "bash", and "Code".

The terminal window shows the following sequence of commands:

```

>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
[main 95c3fcb] assignment completed
 2 files changed, 22 insertions(+), 1 deletion(-)
 create mode 100644 src/q28.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
>>
Enumerating objects: 8, done.

```

At the bottom right of the terminal window, there is a sidebar with several checkboxes labeled "Code", "bash", and "Code".

**Q29.** Write a recursive function named `is Palindrome` that takes a string as input and returns 1 if it is a palindrome (reads the same forwards and backwards), and 0 otherwise.

```

C q22.c      C q23.c      C q25.c      C q26.c      C q27.c      C q28.c      C q29.c      X D v E S M
assignment-3-array-string-and-functions-angelinacy > src > C q29.c > ...
1 // Write a recursive function named isPalindrome that takes a string as input and returns 1 if it is a pal
2 #include <stdio.h>
3 #include <string.h>
4 int isPalindrome(char str[], int start, int end) {
5     if (start >= end) {
6         return 1;
7     }
8     if (str[start] != str[end]) {
9         return 0;
10    int isPalindrome(char *str, int start, int end)
11    return isPalindrome(str, start + 1, end - 1);
12 }
13 int main() {
14     char str[100];
15     int length = 0;
16     printf("Enter a string: ");
17     scanf("%[^\\n]", str);

18     while (str[length] != '\0') {
19         length++;
20     }
21     if (isPalindrome(str, 0, length - 1)) {
22         printf("The string is a palindrome.\n");
23     } else {
24         printf("The string is not a palindrome.\n");
25     }
26     return 0;
27 }
28

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src" ; if ($?) { gcc q29.c -o q29 } ; if ($?) { ./q29 }

Enter a string: 1
The string is a palindrome.
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>

```

Ln 3, Col 20 Spaces: 4 UTF-8 CRLF {} C Win32

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"
>>
[main dce4acf] assignment completed
2 files changed, 27 insertions(+), 1 deletion(-)
create mode 100644 src/q29.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.

```

Ln 3, Col 20 Spaces: 4 UTF-8 CRLF {} C

**Q30. Write a recursive function named count Occurrences that takes a string and a character as input and returns the number of times the character appears in the string.**

```

assignment-3-array-string-and-functions-angelinacy > src > C q30.c > ...
1 // Write a recursive function named countOccurrences that takes a string and a character as input and returns the number of times the character occurs in the string.
2 #include <stdio.h>
3 int countOccurrences(char str[], char ch, int index) {
4     if (str[index] == '\0') {
5         return 0;
6     }
7     int count = (str[index] == ch) ? 1 : 0;
8     return count + countOccurrences(str, ch, index + 1);
9 }
10 int main() {
11     char str[100], ch;
12     printf("Enter a string: ");
13     scanf("%[^\\n]", str);
14     getchar();
15     printf("Enter the character to count: ");
16     scanf("%c", &ch);
17     int result = countOccurrences(str, ch, 0);
18     printf("The character '%c' occurs %d time(s) in the string.\n", ch, result);
19 }
20
21

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```

PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> cd "c:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src" ; if ($?) { gcc q30.c -o q30 } ; if ($?) { ./q30 }
Enter a string: 1
Enter the character to count: 2
The character '2' occurs 0 time(s) in the string.
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git add .
>>
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git commit -m "assignment completed"

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```

completed"
>>
[main a1ec68f] assignment completed
2 files changed, 20 insertions(+), 1 deletion(-)
create mode 100644 src/q30.exe
PS C:\Users\Lenovo\Desktop\C\assignment-3-array-string-and-functions-angelinacy\src> git push origin main
>>
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads

```

