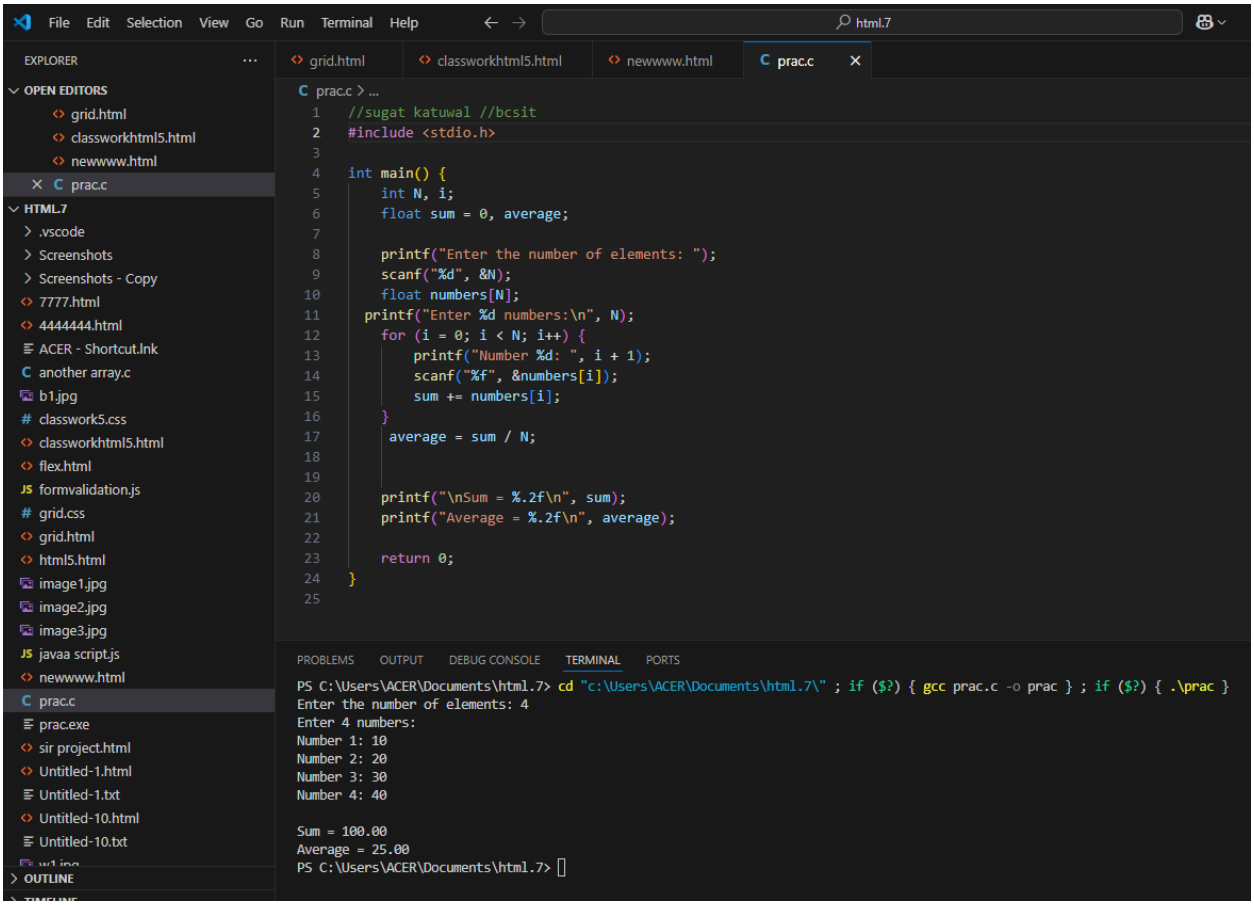
1. Write a C program to input **N** numbers in an array, calculate and display their **sum and average**.

```c
//sugat katuwal //bcsit
#include <stdio.h>

int main() {
    int N, i;
    float sum = 0, average;

    printf("Enter the number of elements: ");
    scanf("%d", &N);
    float numbers[N];
  printf("Enter %d numbers:\n", N);
    for (i = 0; i < N; i++) {
        printf("Number %d: ", i + 1);
        scanf("%f", &numbers[i]);
        sum += numbers[i];
    }
     average = sum / N;


    printf("\nSum = %.2f\n", sum);
    printf("Average = %.2f\n", average);

    return 0;
}
```

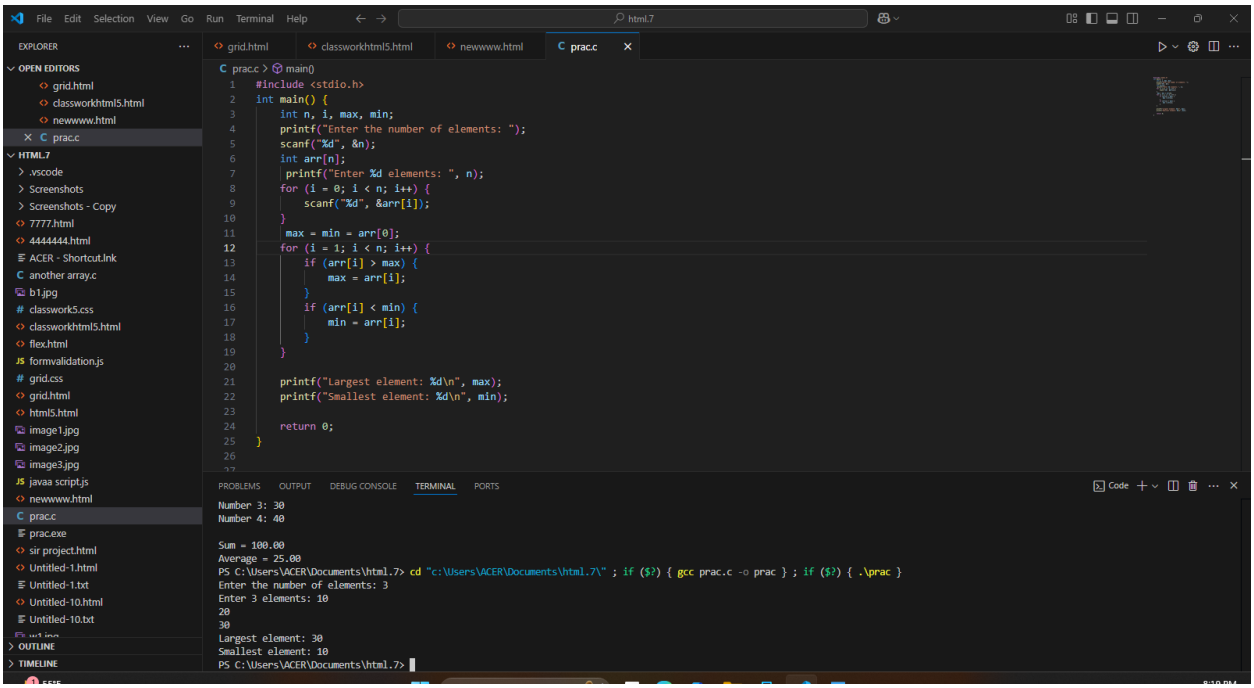


Write a C program to find the **largest and smallest** element in an array

```c
//sugat katuwal
#include <stdio.h>
int main() {
    int n, i, max, min;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];
     printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
     max = min = arr[0];
    for (i = 1; i < n; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
        if (arr[i] < min) {
```

```
            min = arr[i];
        }
    }

    printf("Largest element: %d\n", max);
    printf("Smallest element: %d\n", min);

    return 0;
}
```

\\



1. Write a C program to **reverse** the elements of an array.

//sugat katuwal

//bcsit

#include <stdio.h>

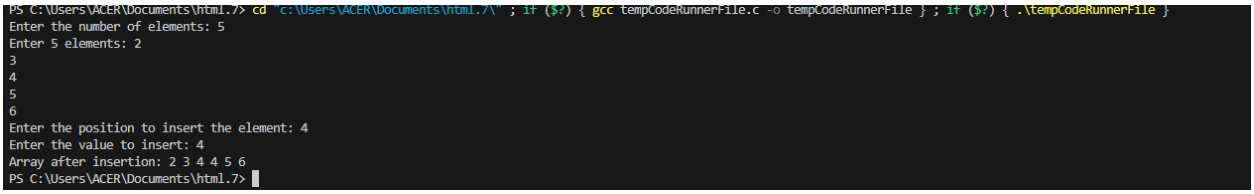
int main() {

```
    int n, i, temp;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];

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

    for (i = 0; i < n / 2; i++) {
        temp = arr[i];
        arr[i] = arr[n - i - 1];
        arr[n - i - 1] = temp;
    }

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

    return 0;
}
```

```
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
Enter the number of elements: 2
Enter 2 elements: 20
30
Reversed array: 30 20
PS C:\Users\ACER\Documents\html.7>
```

Write a C program to **insert an element** at a specific position in an array

```c
//sugat katuwal
int main() {
    int n, i, pos, value;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n + 1];

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

    printf("Enter the position to insert the element: ");
    scanf("%d", &pos);

    printf("Enter the value to insert: ");
    scanf("%d", &value);

    for (i = n; i >= pos; i--) {
        arr[i] = arr[i - 1];
    }
    arr[pos - 1] = value;
    n++;

    printf("Array after insertion: ");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```



Write a C program to **delete an element** from an array at a specific position

```c
//sugat katuwal
#include <stdio.h>

int main() {
    int n, i, pos;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];

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

    printf("Enter the position to delete the element: ");
    scanf("%d", &pos);

    if (pos < 1 || pos > n) {
        printf("Invalid position!\n");
        return 1;
    }

    for (i = pos - 1; i < n - 1; i++) {
        arr[i] = arr[i + 1];
    }
    n--;

    printf("Array after deletion: ");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
```

```
}
```

```
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter the number of elements: 3
Enter 3 elements: 1
2
3
Enter the position to delete the element: 2
Array after deletion: 1 3
PS C:\Users\ACER\Documents\html.7>
```

Write a C program to **add two matrices** of order **m × n**

```c
//sugat katuwal
//bcsit
#include <stdio.h>

int main() {
    int m, n;
    printf("Enter the number of rows and columns: ");
    scanf("%d %d", &m, &n);
    int a[m][n], b[m][n], sum[m][n];
    printf("Enter elements of first matrix:\n");
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &a[i][j]);
        }
    }
    printf("Enter elements of second matrix:\n");
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &b[i][j]);
        }
    }
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            sum[i][j] = a[i][j] + b[i][j];
        }
    }
    printf("Sum of the matrices:\n");
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            printf("%d ", sum[i][j]);
        }
        printf("\n");
    }
    return 0;
```

```
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
2
3
4
5
Enter elements of second matrix:
1
2
3
4
Sum of the matrices:
3 5
7 9
PS C:\Users\ACER\Documents\html.7>
```

Write a C program to **multiply two matrices**

```c
//sugat katuwal
#include <stdio.h>

void multiplyMatrices(int firstMatrix[][10], int secondMatrix[][10], int
result[][10], int row1, int col1, int row2, int col2) {
    for (int i = 0; i < row1; i++) {
        for (int j = 0; j < col2; j++) {
            result[i][j] = 0;
            for (int k = 0; k < col1; k++) {
                result[i][j] += firstMatrix[i][k] * secondMatrix[k][j];
            }
        }
    }
}

void inputMatrix(int matrix[][10], int row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            printf("Enter element [%d][%d]: ", i + 1, j + 1);
            scanf("%d", &matrix[i][j]);
        }
    }
}

void printMatrix(int matrix[][10], int row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
```

```
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }
}

int main() {
    int row1, col1, row2, col2;
    int firstMatrix[10][10], secondMatrix[10][10], result[10][10];

    printf("Enter rows and columns for first matrix: ");
    scanf("%d %d", &row1, &col1);
    printf("Enter rows and columns for second matrix: ");
    scanf("%d %d", &row2, &col2);

    if (col1 != row2) {
        printf("Error: Matrix multiplication not possible!\n");
        return 1;
    }

    printf("Enter elements of first matrix:\n");
    inputMatrix(firstMatrix, row1, col1);

    printf("Enter elements of second matrix:\n");
    inputMatrix(secondMatrix, row2, col2);

    multiplyMatrices(firstMatrix, secondMatrix, result, row1, col1, row2, col2);

    printf("Resultant matrix:\n");
    printMatrix(result, row1, col2);

    return 0;
}
```

```
Enter element [1][2]: 21
Enter element [1][3]: 22
Enter element [2][1]: 23
Enter element [2][2]: 24
Enter element [2][3]: 25
Enter element [3][1]: 26
Enter element [3][2]: 27
Enter element [3][3]: 28
Resultant matrix:
808     843     878
1041    1086    1131
1248    1302    1356
PS C:\Users\ACER\Documents\html.7>
```

Write a C program to find the transpose of a given matrix

```c
//sugat katuwal
//bcsit
#include <stdio.h>

#define MAX 10

int main() {
    int matrix[MAX][MAX], transpose[MAX][MAX];
    int rows, cols;

    printf("Enter rows and columns: ");
    scanf("%d %d", &rows, &cols);

    printf("Enter matrix elements:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            transpose[j][i] = matrix[i][j];
        }
    }

    printf("Transpose of the matrix:\n");
    for (int i = 0; i < cols; i++) {
        for (int j = 0; j < rows; j++) {
```

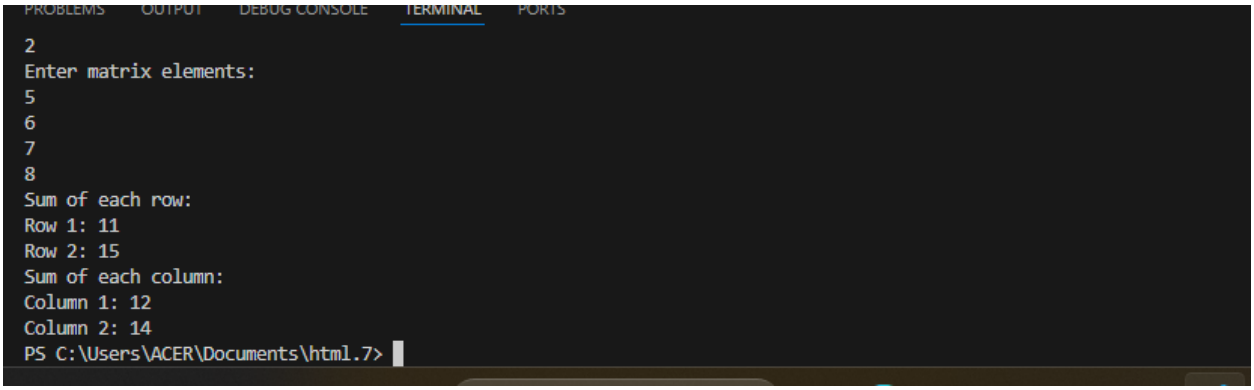
```
            printf("%d ", transpose[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
Enter rows and columns: 2
2
Enter matrix elements:
2
3
4
5
Transpose of the matrix:
2 4
3 5
PS C:\Users\ACER\Documents\html.7>
```

Write a C program to find the sum of each row and each column of a matrix

```
//sugat katuwal
//bcsit
#include <stdio.h>

#define MAX 10

int main() {
    int matrix[MAX][MAX], rowSum[MAX] = {0}, colSum[MAX] = {0};
    int rows, cols;

    printf("Enter rows and columns: ");
    scanf("%d %d", &rows, &cols);

    printf("Enter matrix elements:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &matrix[i][j]);
            rowSum[i] += matrix[i][j];
            colSum[j] += matrix[i][j];
        }
    }

    printf("Sum of each row:\n");
    for (int i = 0; i < rows; i++) {
```

```
        printf("Row %d: %d\n", i + 1, rowSum[i]);
    }

    printf("Sum of each column:\n");
    for (int j = 0; j < cols; j++) {
        printf("Column %d: %d\n", j + 1, colSum[j]);
    }

    return 0;
}
```



Write a C program to input **N** names and **display them**

```
//sugat katuwal
//bcsit
#include <stdio.h>

#define MAX 10

int main() {
    char names[MAX][50];
    int n;

    printf("Enter number of names: ");
    scanf("%d", &n);

    printf("Enter %d names:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%s", names[i]);
    }

    printf("\nThe entered names are:\n");
```

```
    for (int i = 0; i < n; i++) {
        printf("%s\n", names[i]);
    }

    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
Enter number of names: 3
Enter 3 names:
shyam
ram
laxman

The entered names are:
shyam
ram
laxman
PS C:\Users\ACER\Documents\html.7>
```

Write a program to **reverse a given string** without using string functions.

```
//sugat katuwal
//bcsit
#include <stdio.h>

int main() {
    char str[100];
    int i, length = 0;
    char temp;

    printf("Enter a string: ");
    scanf("%s", str);

    while (str[length] != '\0') {
        length++;
    }

    for (i = 0; i < length / 2; i++) {
        temp = str[i];
        str[i] = str[length - i - 1];
        str[length - i - 1] = temp;
    }

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

```
    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
Enter a string: shyam
Reversed string: mayhs
PS C:\Users\ACER\Documents\html.7>
```

Write a program to check if a string is a **palindrome** (same forward & backward)

```c
//sugat katuwal
//bcsit
#include <stdio.h>
#include <string.h>

int main() {
    char str[100];
    int i, length;

    scanf("%s", str);
    length = strlen(str);

    for (i = 0; i < length / 2; i++) {
        if (str[i] != str[length - i - 1]) {
            printf("No\n");
            return 0;
        }
    }
    printf("Yes\n");
    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
ssss
Yes
PS C:\Users\ACER\Documents\html.7>
```

Write a C program to **count vowels and consonants** in a given string.

```c
//sugat katuwal
//bcsit
```

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

int main() {
    char str[100];
    int vowels = 0, consonants = 0, i;

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

    for (i = 0; str[i] != '\0'; i++) {
        char ch = tolower(str[i]);
        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
            vowels++;
        } else if ((ch >= 'a' && ch <= 'z')) {
            consonants++;
        }
    }

    printf("Vowels: %d\n", vowels);
    printf("Consonants: %d\n", consonants);

    return 0;
}
```

```
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
ssss
Yes
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
sugat
Vowels: 2
Consonants: 3
PS C:\Users\ACER\Documents\html.7>
```

Write a program to find the **length of a string** without using `strlen()`

```c
//sugat katuwal
//bcsit
#include <stdio.h>

int main() {
    char str[100];
    int length = 0;

    scanf("%s", str);
```

```
    while (str[length] != '\0') {
        length++;
    }

    printf("Length: %d\n", length);

    return 0;
}
```

```
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
sugat
Length: 5
PS C:\Users\ACER\Documents\html.7>
```

1. Write a program to **concatenate two strings** without using `strcat()`.

```
//sugat katuwal
//bcsit
#include <stdio.h>

int main() {
    char str1[100], str2[100];
    int i = 0, j = 0;

    scanf("%s", str1);
    scanf("%s", str2);

    // Move to the end of str1
    while (str1[i] != '\0') {
        i++;
    }

    // Append str2 to str1
    while (str2[j] != '\0') {
        str1[i] = str2[j];
        i++;
        j++;
    }

    // Null-terminate the concatenated string
    str1[i] = '\0';

    printf("Concatenated string: %s\n", str1);

    return 0;
```

```
31.}
32.
```

```
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
hello
word
Concatenated string: helloword
PS C:\Users\ACER\Documents\html.7>
```

Write a program to copy one string to another without using strcpy()

```c
//sugat katuwal
//bcsit
#include <stdio.h>

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

    printf("Enter a string: ");
    gets(str1);

    while (str1[i] != '\0') {
        str2[i] = str1[i];
        i++;
    }
    str2[i] = '\0';

    printf("Copied string: %s", str2);

    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
Enter a string: aaaa
Copied string: aaaa
PS C:\Users\ACER\Documents\html.7>
```

Write a function to calculate the **factorial** of a number using recursion

```c
//sugat katuwal
```

```c
//bcsit
#include <stdio.h>

int factorial(int n) {
    if (n == 0 || n == 1)
        return 1;
    return n * factorial(n - 1);
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    printf("Factorial of %d is %d", num, factorial(num));
    return 0;
}
```

```
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
Enter a number: 2
Factorial of 2 is 2
PS C:\Users\ACER\Documents\html.7>
```

Write a function that checks whether a given number is **prime** or not

```c
//sugat katuwal
//bcsit
#include <stdio.h>

int is_prime(int n) {
    if (n < 2) return 0;
    for (int i = 2; i * i <= n; i++) {
        if (n % i == 0) return 0;
    }
    return 1;
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    if (is_prime(num))
        printf("%d is a prime number", num);
```

```
    else
        printf("%d is not a prime number", num);
    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
Enter a number: 4
4 is not a prime number
PS C:\Users\ACER\Documents\html.7>
```

Write a function to **swap two numbers** using **call by reference**.

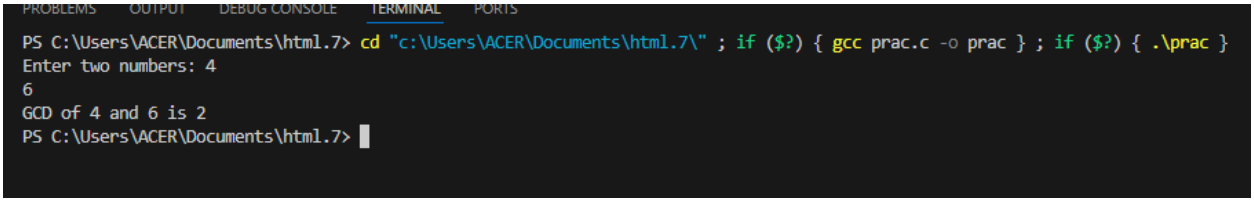
```
//sugat katuwal
//bcsit
#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    int x, y;
    printf("Enter two numbers: ");
    scanf("%d %d", &x, &y);
    printf("Before swapping: x = %d, y = %d\n", x, y);
    swap(&x, &y);
    printf("After swapping: x = %d, y = %d\n", x, y);
    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
Enter two numbers: 2
3
Before swapping: x = 2, y = 3
After swapping: x = 3, y = 2
PS C:\Users\ACER\Documents\html.7>
```

Write a function to find the **GCD (Greatest Common Divisor)** of two numbers

```
//sugat katuwal
//bcsit
#include <stdio.h>
```

```c
int gcd(int a, int b) {
    if (b == 0)
        return a;
    return gcd(b, a % b);
}

int main() {
    int x, y;
    printf("Enter two numbers: ");
    scanf("%d %d", &x, &y);
    printf("GCD of %d and %d is %d\n", x, y, gcd(x, y));
    return 0;
}
```



Write a function to find the **LCM (Least Common Multiple)** of two numbers

```c
//sugat katuwal
//bcsit
#include <stdio.h>

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

int lcm(int a, int b) {
    return (a * b) / gcd(a, b);
}

int main() {
    int x, y;
    printf("Enter two numbers: ");
    scanf("%d %d", &x, &y);
    printf("GCD of %d and %d is %d\n", x, y, gcd(x, y));
    printf("LCM of %d and %d is %d\n", x, y, lcm(x, y));
```

```
    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
Enter two numbers: 7
9
GCD of 7 and 9 is 1
LCM of 7 and 9 is 63
PS C:\Users\ACER\Documents\html.7> \
```

Write a recursive function to generate the **Fibonacci series** up to N terms

```c
//sugat katuwal
//bcsit
#include <stdio.h>

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

int main() {
    int n;
    printf("Enter the number of terms: ");
    scanf("%d", &n);
    printf("Fibonacci Series: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", fibonacci(i));
    }
    printf("\n");
    return 0;
}
```

```
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
Enter the number of terms: 10
Fibonacci Series: 0 1 1 2 3 5 8 13 21 34
PS C:\Users\ACER\Documents\html.7>
```

Write a function to **find the sum of digits** of a number using recursion

```c
//sugat katuwal
//bcsit
#include <stdio.h>
```

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

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    printf("Sum of digits of %d is %d\n", num, sum_of_digits(num));
    return 0;
}
```

```
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
Enter a number: 2
Sum of digits of 2 is 2
PS C:\Users\ACER\Documents\html.7>
```

Write a function to **find the sum of all elements in a 1D array**

```c
//sugat katuwal
//bcsit
#include <stdio.h>

int sum_of_array(int arr[], int size) {
    int sum = 0;
    for (int i = 0; i < size; i++) {
        sum += arr[i];
    }
    return sum;
}

int main() {
    int array[] = {1, 2, 3, 4, 5};
    int size = sizeof(array) / sizeof(array[0]);
    printf("%d\n", sum_of_array(array, size));
    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc prac.c -o prac } ; if ($?) { .\prac }
15
PS C:\Users\ACER\Documents\html.7>
```

Write a function to **count the number of words** in a given string.

```c
//sugat katuwal
//bcsit
#include <stdio.h>
#include <ctype.h>

int count_words(const char *str) {
    int count = 0, in_word = 0;

    while (*str) {
        if (isspace(*str)) {
            in_word = 0;
        } else if (!in_word) {
            in_word = 1;
            count++;
        }
        str++;
    }

    return count;
}

int main() {
    char str[] = "Hello, how are you today?";
    printf("Word count: %d\n", count_words(str));
    return 0;
}
```

```
PS C:\Users\ACER\Documents\html.7> cd "c:\Users\ACER\Documents\html.7\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Word count: 5
PS C:\Users\ACER\Documents\html.7>
```