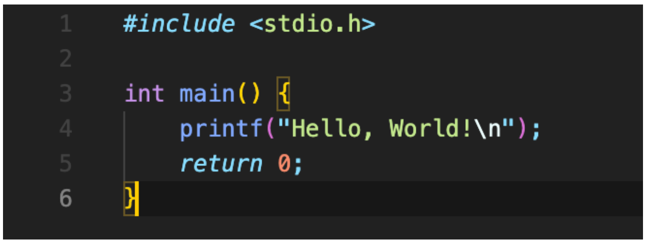
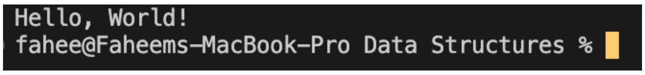
Practice of C-programming

1. Write a program to print "Hello, World!" to the console.

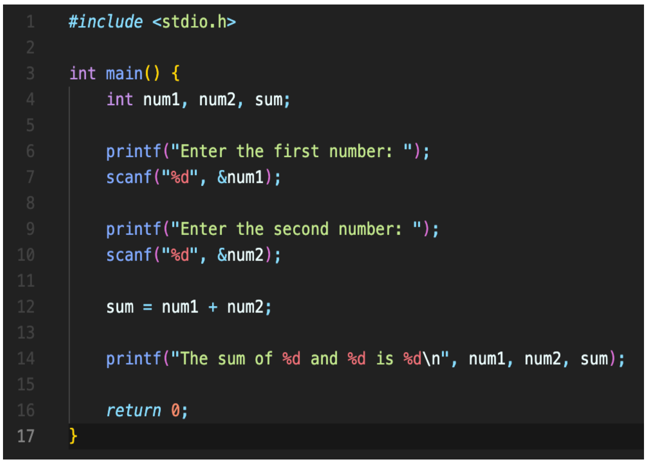


```c
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

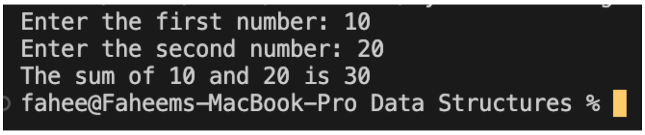


```
Hello, World!
fahee@Faheems-MacBook-Pro Data Structures %
```

2. Write a program to find the sum of two numbers.

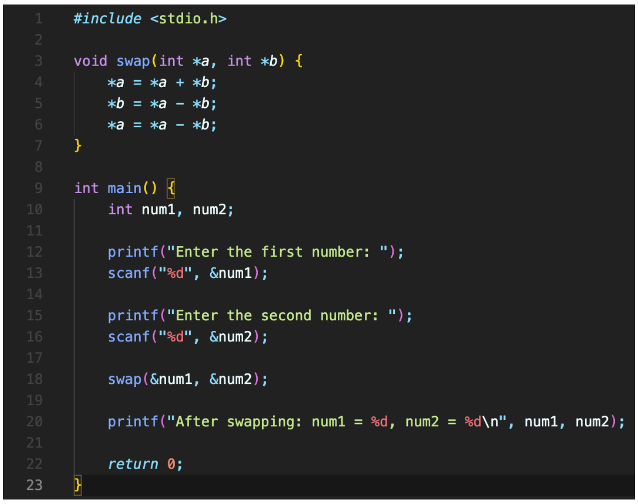


```c
#include <stdio.h>

int main() {
    int num1, num2, sum;

    printf("Enter the first number: ");
    scanf("%d", &num1);

    printf("Enter the second number: ");
    scanf("%d", &num2);

    sum = num1 + num2;

    printf("The sum of %d and %d is %d\n", num1, num2, sum);

    return 0;
}
```

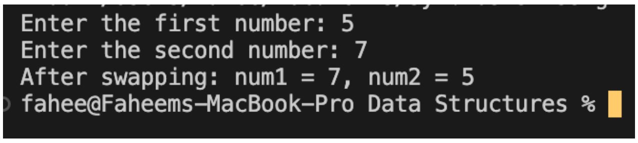


```
Enter the first number: 10
Enter the second number: 20
The sum of 10 and 20 is 30
fahee@Faheems-MacBook-Pro Data Structures %
```

3. Write a program to swap two numbers without using a temporary variable.

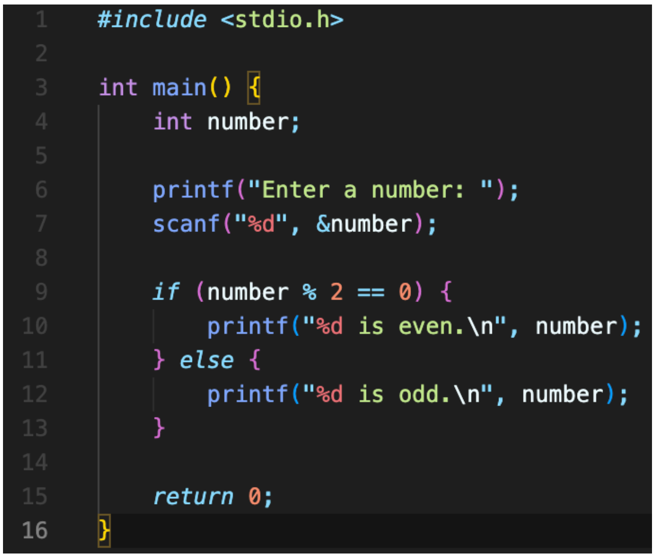


```c
#include <stdio.h>

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

int main() {
    int num1, num2;

    printf("Enter the first number: ");
    scanf("%d", &num1);

    printf("Enter the second number: ");
    scanf("%d", &num2);

    swap(&num1, &num2);

    printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);

    return 0;
}
```

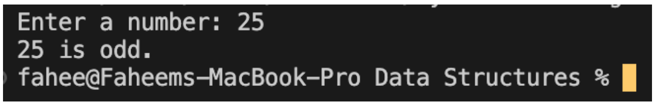


```
Enter the first number: 5
Enter the second number: 7
After swapping: num1 = 7, num2 = 5
fahee@Faheems-MacBook-Pro Data Structures %
```

4. Write a program to check if a given number is even or odd.

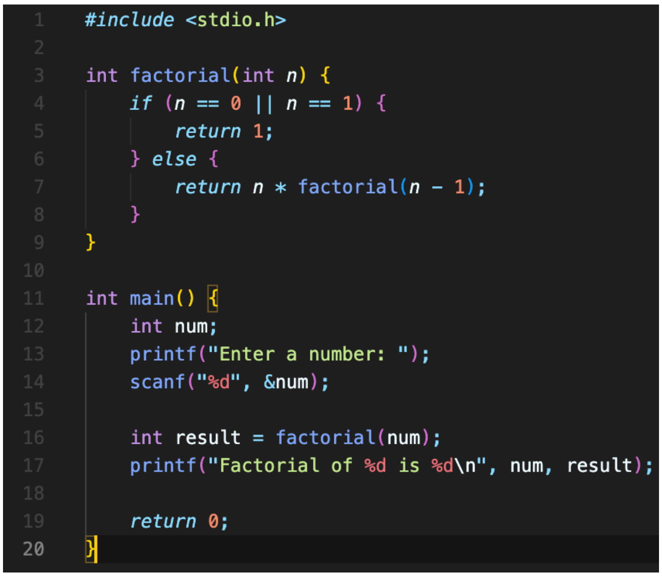


```c
#include <stdio.h>

int main() {
    int number;

    printf("Enter a number: ");
    scanf("%d", &number);

    if (number % 2 == 0) {
        printf("%d is even.\n", number);
    } else {
        printf("%d is odd.\n", number);
    }

    return 0;
}
```

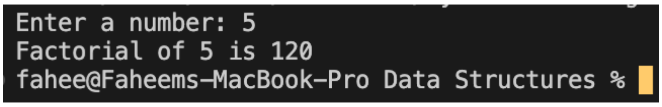


```
Enter a number: 25
25 is odd.
fahee@Faheems-MacBook-Pro Data Structures %
```

5. Write a program to find the factorial of a given number.

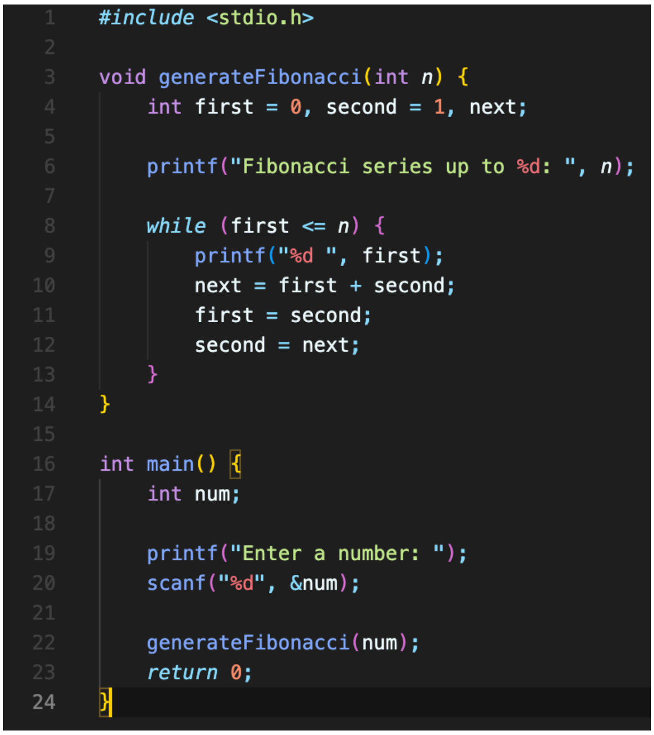


```c
#include <stdio.h>

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

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

    int result = factorial(num);
    printf("Factorial of %d is %d\n", num, result);

    return 0;
}
```

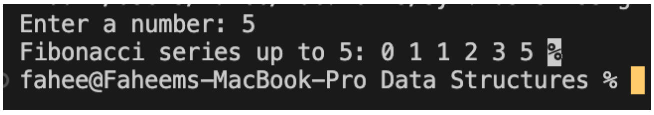


```
Enter a number: 5
Factorial of 5 is 120
fahee@Faheems-MacBook-Pro Data Structures %
```

6. Write a program to generate the Fibonacci series up to a given number.

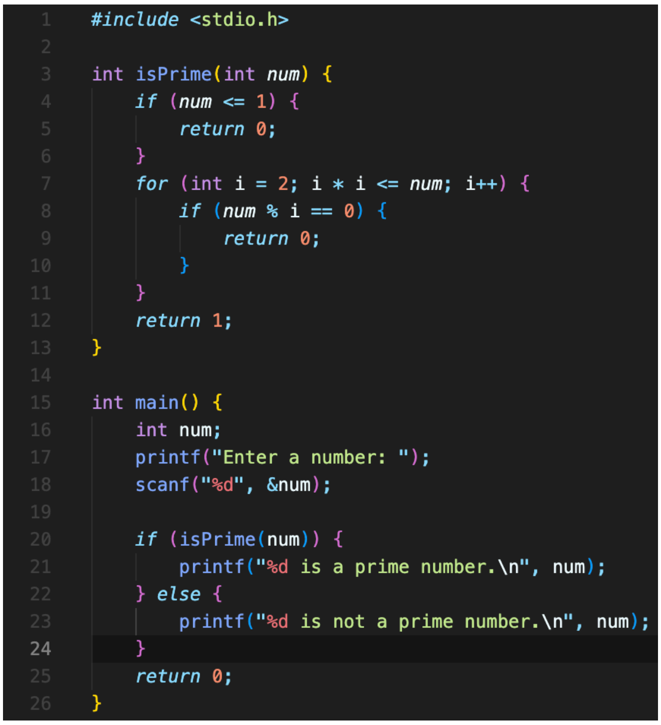


```c
#include <stdio.h>

void generateFibonacci(int n) {
    int first = 0, second = 1, next;

    printf("Fibonacci series up to %d: ", n);

    while (first <= n) {
        printf("%d ", first);
        next = first + second;
        first = second;
        second = next;
    }
}

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    generateFibonacci(num);
    return 0;
}
```

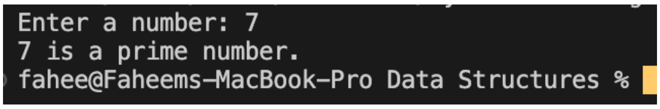


```
Enter a number: 5
Fibonacci series up to 5: 0 1 1 2 3 5 %
fahee@Faheems-MacBook-Pro Data Structures %
```

7. Write a program to check if a given number is prime.

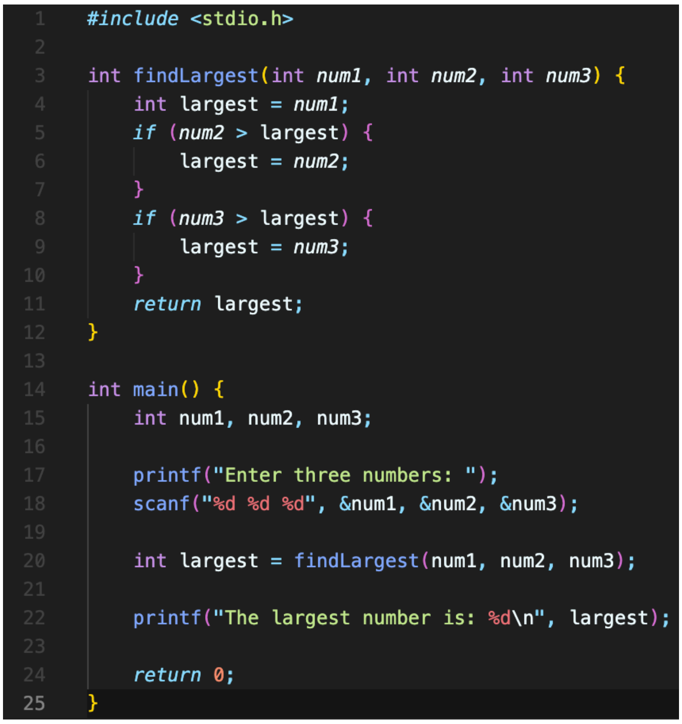


```c
#include <stdio.h>

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

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

    if (isPrime(num)) {
        printf("%d is a prime number.\n", num);
    } else {
        printf("%d is not a prime number.\n", num);
    }
    return 0;
}
```

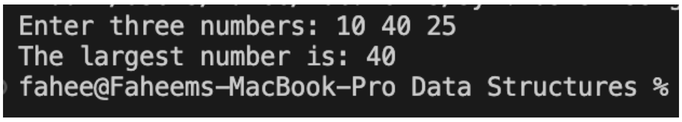


```
Enter a number: 7
7 is a prime number.
fahee@Faheems-MacBook-Pro Data Structures %
```

8. Write a program to find the largest of three numbers.

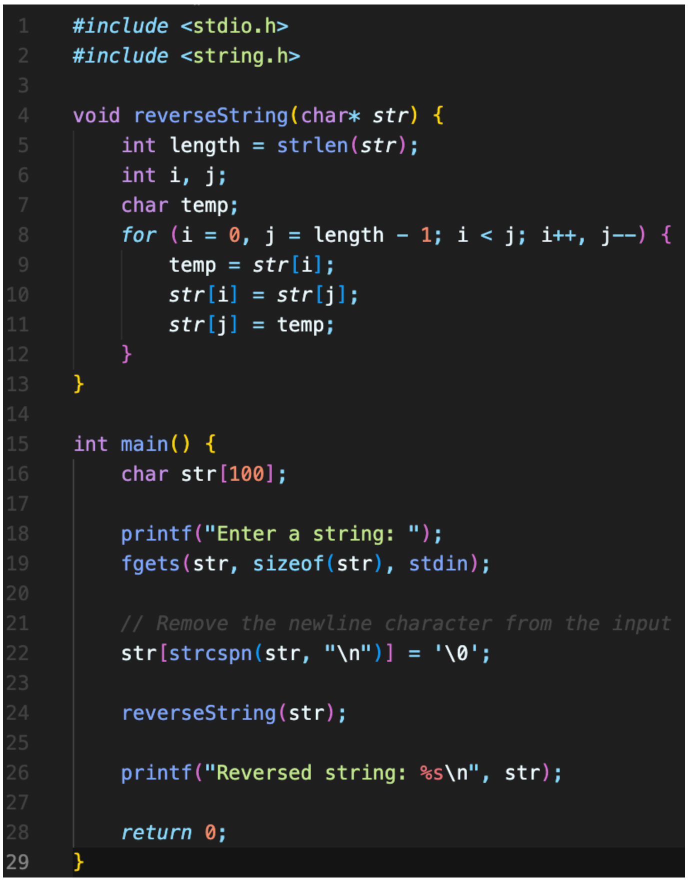


```c
#include <stdio.h>

int findLargest(int num1, int num2, int num3) {
    int largest = num1;
    if (num2 > largest) {
        largest = num2;
    }
    if (num3 > largest) {
        largest = num3;
    }
    return largest;
}

int main() {
    int num1, num2, num3;

    printf("Enter three numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);

    int largest = findLargest(num1, num2, num3);

    printf("The largest number is: %d\n", largest);

    return 0;
}
```

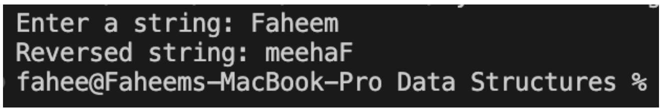


```
Enter three numbers: 10 40 25
The largest number is: 40
fahee@Faheems-MacBook-Pro Data Structures %
```

9. Write a program to reverse a given string.

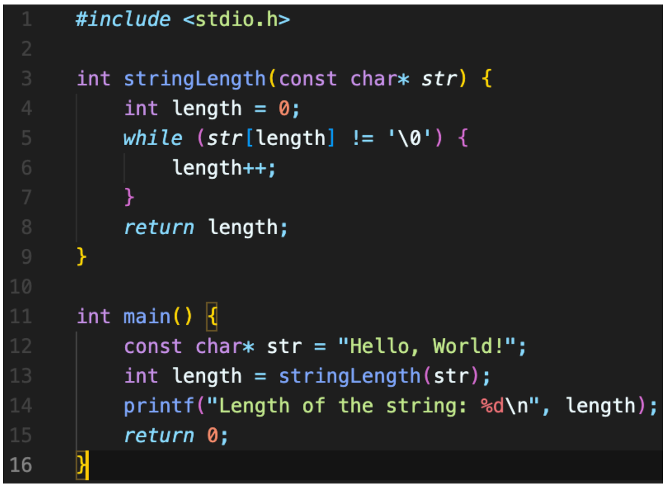


```c
#include <stdio.h>
#include <string.h>

void reverseString(char* str) {
    int length = strlen(str);
    int i, j;
    char temp;
    for (i = 0, j = length - 1; i < j; i++, j--) {
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
    }
}

int main() {
    char str[100];

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    // Remove the newline character from the input
    str[strcspn(str, "\n")] = '\0';

    reverseString(str);

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

    return 0;
}
```

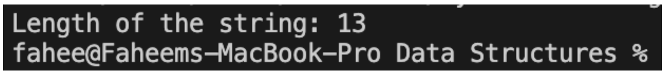


```
Enter a string: Faheem
Reversed string: meehaF
fahee@Faheems-MacBook-Pro Data Structures %
```

10. Write a program to find the length of a string without using the strlen function.

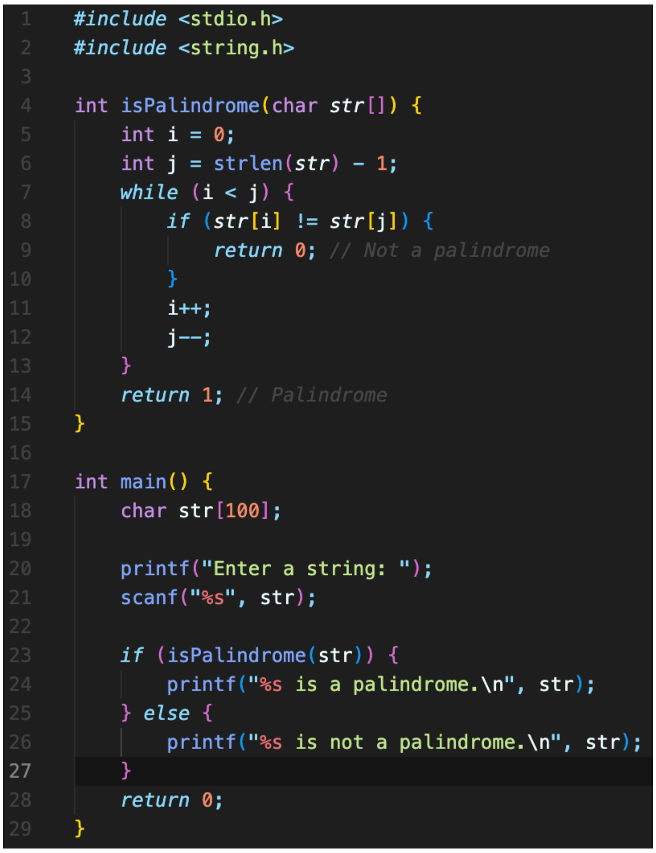


```c
#include <stdio.h>

int stringLength(const char* str) {
    int length = 0;
    while (str[length] != '\0') {
        length++;
    }
    return length;
}

int main() {
    const char* str = "Hello, World!";
    int length = stringLength(str);
    printf("Length of the string: %d\n", length);
    return 0;
}
```

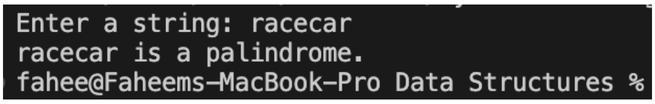


```
Length of the string: 13
fahee@Faheems-MacBook-Pro Data Structures %
```

11. Write a program to check if a given string is a palindrome.

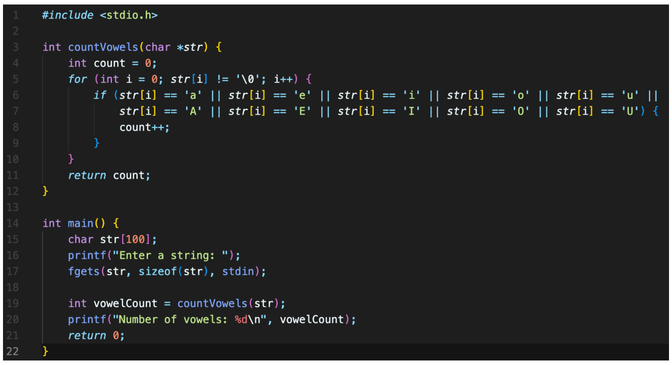


```c
#include <stdio.h>
#include <string.h>

int isPalindrome(char str[]) {
    int i = 0;
    int j = strlen(str) - 1;
    while (i < j) {
        if (str[i] != str[j]) {
            return 0; // Not a palindrome
        }
        i++;
        j--;
    }
    return 1; // Palindrome
}

int main() {
    char str[100];

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

    if (isPalindrome(str)) {
        printf("%s is a palindrome.\n", str);
    } else {
        printf("%s is not a palindrome.\n", str);
    }
    return 0;
}
```

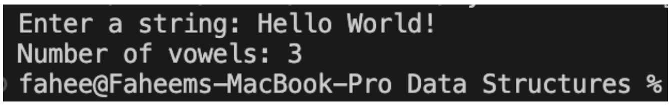


```
Enter a string: racecar
racecar is a palindrome.
fahee@Faheems-MacBook-Pro Data Structures %
```

12. Write a program to count the number of vowels in a given string.

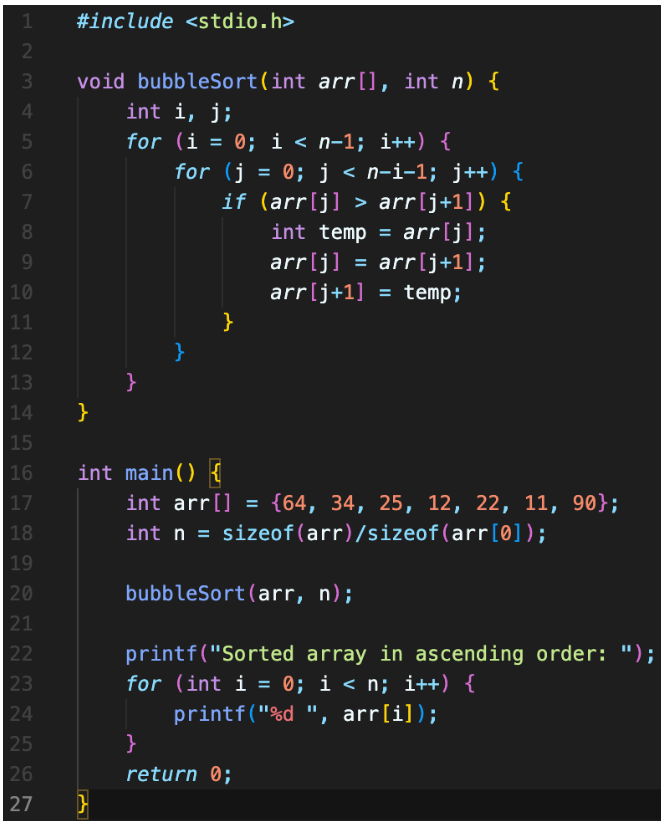


```c
#include <stdio.h>

int countVowels(char *str) {
    int count = 0;
    for (int i = 0; str[i] != '\0'; i++) {
        if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u' ||
            str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U') {
            count++;
        }
    }
    return count;
}

int main() {
    char str[100];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    int vowelCount = countVowels(str);
    printf("Number of vowels: %d\n", vowelCount);
    return 0;
}
```

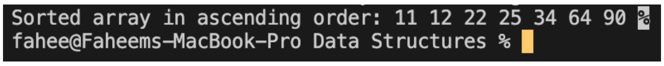


```
Enter a string: Hello World!
Number of vowels: 3
fahee@Faheems-MacBook-Pro Data Structures %
```

13. Write a program to sort an array of integers in ascending order.

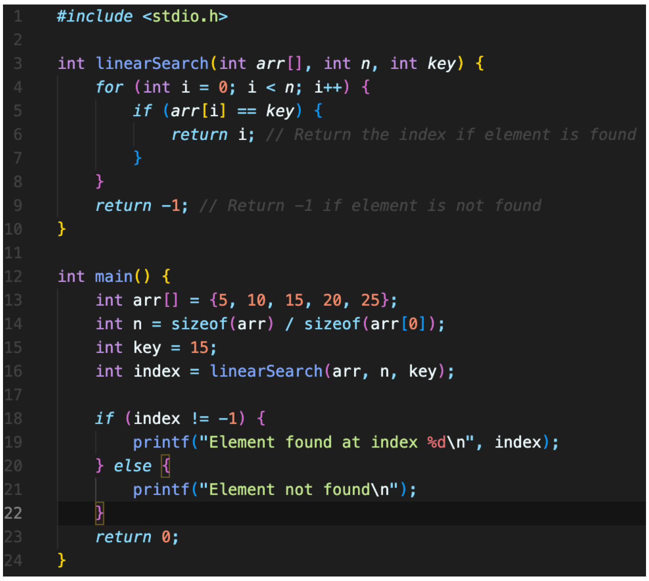


```c
#include <stdio.h>

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

int main() {
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);

    bubbleSort(arr, n);

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

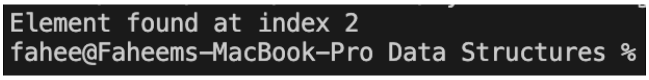


```
Sorted array in ascending order: 11 12 22 25 34 64 90 %
fahee@Faheems-MacBook-Pro Data Structures %
```

14. Write a program to search for an element in an array using linear search.

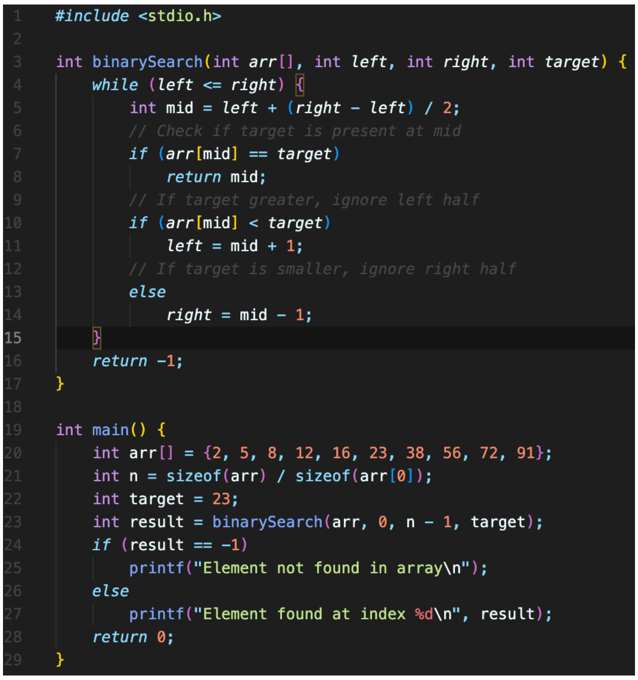


```c
#include <stdio.h>

int linearSearch(int arr[], int n, int key) {
    for (int i = 0; i < n; i++) {
        if (arr[i] == key) {
            return i; // Return the index if element is found
        }
    }
    return -1; // Return -1 if element is not found
}

int main() {
    int arr[] = {5, 10, 15, 20, 25};
    int n = sizeof(arr) / sizeof(arr[0]);
    int key = 15;
    int index = linearSearch(arr, n, key);

    if (index != -1) {
        printf("Element found at index %d\n", index);
    } else {
        printf("Element not found\n");
    }
    return 0;
}
```

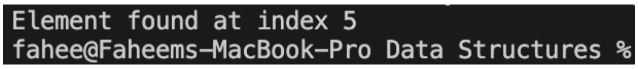


```
Element found at index 2
fahee@Faheems-MacBook-Pro Data Structures %
```

15. Write a program to search for an element in an array using binary search.

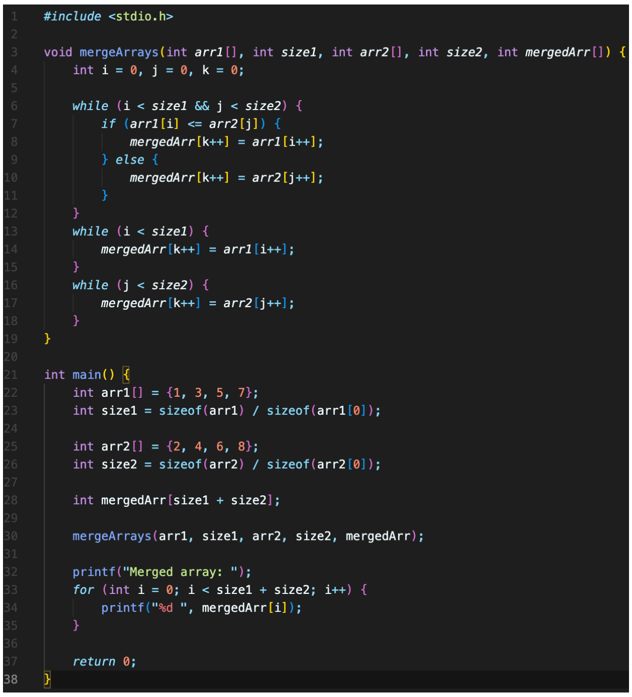


```c
#include <stdio.h>

int binarySearch(int arr[], int left, int right, int target) {
    while (left <= right) {
        int mid = left + (right - left) / 2;
        // Check if target is present at mid
        if (arr[mid] == target)
            return mid;
        // If target greater, ignore left half
        if (arr[mid] < target)
            left = mid + 1;
        // If target is smaller, ignore right half
        else
            right = mid - 1;
    }
    return -1;
}

int main() {
    int arr[] = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91};
    int n = sizeof(arr) / sizeof(arr[0]);
    int target = 23;
    int result = binarySearch(arr, 0, n - 1, target);
    if (result == -1)
        printf("Element not found in array\n");
    else
        printf("Element found at index %d\n", result);
    return 0;
}
```

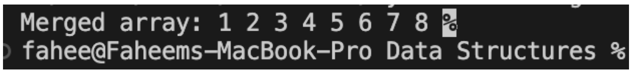


```
Element found at index 5
fahee@Faheems-MacBook-Pro Data Structures %
```

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

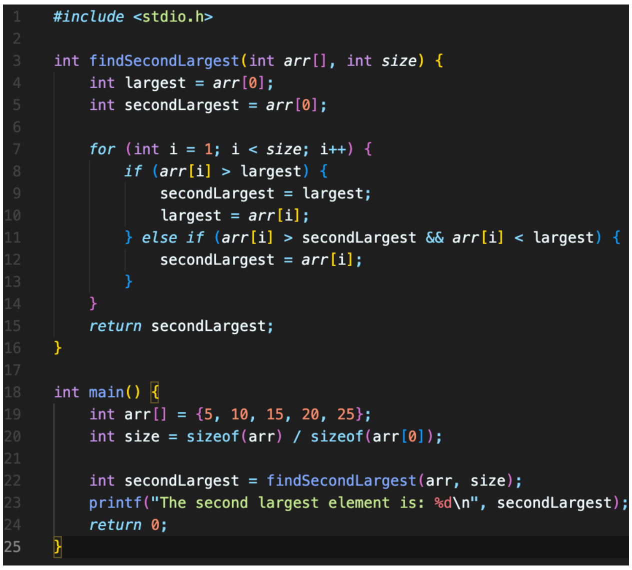


```c
#include <stdio.h>

void mergeArrays(int arr1[], int size1, int arr2[], int size2, int mergedArr[]) {
    int i = 0, j = 0, k = 0;

    while (i < size1 && j < size2) {
        if (arr1[i] <= arr2[j]) {
            mergedArr[k++] = arr1[i++];
        } else {
            mergedArr[k++] = arr2[j++];
        }
    }
    while (i < size1) {
        mergedArr[k++] = arr1[i++];
    }
    while (j < size2) {
        mergedArr[k++] = arr2[j++];
    }
}

int main() {
    int arr1[] = {1, 3, 5, 7};
    int size1 = sizeof(arr1) / sizeof(arr1[0]);

    int arr2[] = {2, 4, 6, 8};
    int size2 = sizeof(arr2) / sizeof(arr2[0]);

    int mergedArr[size1 + size2];

    mergeArrays(arr1, size1, arr2, size2, mergedArr);

    printf("Merged array: ");
    for (int i = 0; i < size1 + size2; i++) {
        printf("%d ", mergedArr[i]);
    }

    return 0;
}
```

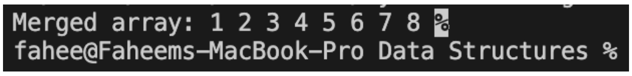


```
Merged array: 1 2 3 4 5 6 7 8 %
fahee@Faheems-MacBook-Pro Data Structures %
```

17. Write a program to find the second largest element in an array.

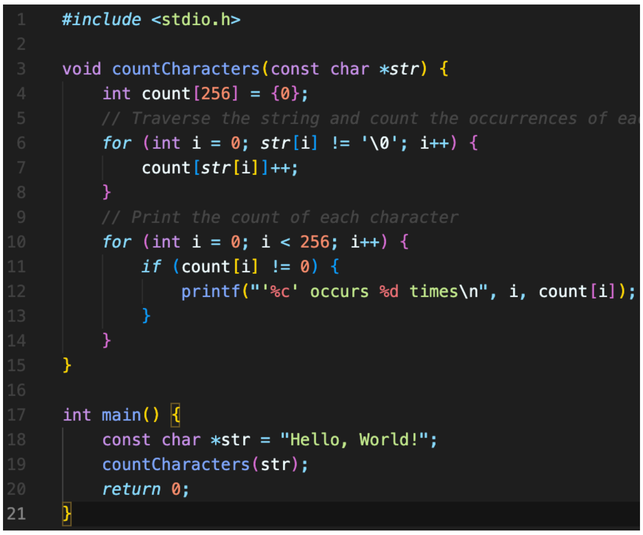


```c
#include <stdio.h>

int findSecondLargest(int arr[], int size) {
    int largest = arr[0];
    int secondLargest = arr[0];

    for (int i = 1; i < size; i++) {
        if (arr[i] > largest) {
            secondLargest = largest;
            largest = arr[i];
        } else if (arr[i] > secondLargest && arr[i] < largest) {
            secondLargest = arr[i];
        }
    }
    return secondLargest;
}

int main() {
    int arr[] = {5, 10, 15, 20, 25};
    int size = sizeof(arr) / sizeof(arr[0]);

    int secondLargest = findSecondLargest(arr, size);
    printf("The second largest element is: %d\n", secondLargest);
    return 0;
}
```

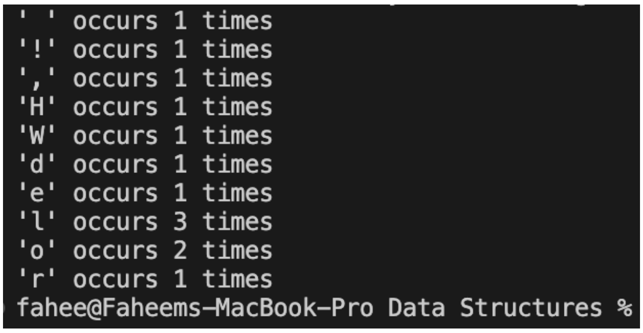


```
Merged array: 1 2 3 4 5 6 7 8 %
fahee@Faheems-MacBook-Pro Data Structures %
```

18. Write a program to count the number of occurrences of each character in a string.

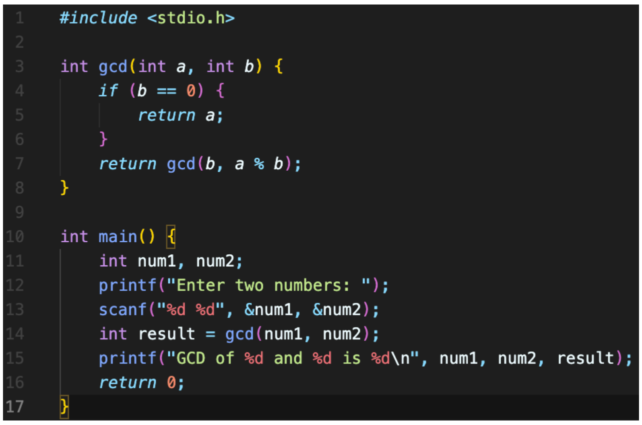


```c
#include <stdio.h>

void countCharacters(const char *str) {
    int count[256] = {0};
    // Traverse the string and count the occurrences of ea
    for (int i = 0; str[i] != '\0'; i++) {
        count[str[i]]++;
    }
    // Print the count of each character
    for (int i = 0; i < 256; i++) {
        if (count[i] != 0) {
            printf("'%c' occurs %d times\n", i, count[i]);
        }
    }
}

int main() {
    const char *str = "Hello, World!";
    countCharacters(str);
    return 0;
}
```

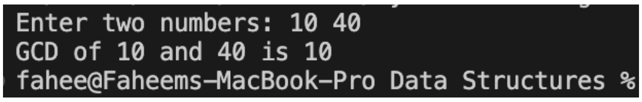


```
' ' occurs 1 times
'!' occurs 1 times
',' occurs 1 times
'H' occurs 1 times
'W' occurs 1 times
'd' occurs 1 times
'e' occurs 1 times
'l' occurs 3 times
'o' occurs 2 times
'r' occurs 1 times
fahee@Faheems-MacBook-Pro Data Structures %
```

19. Write a program to find the GCD (Greatest Common Divisor) of two numbers.

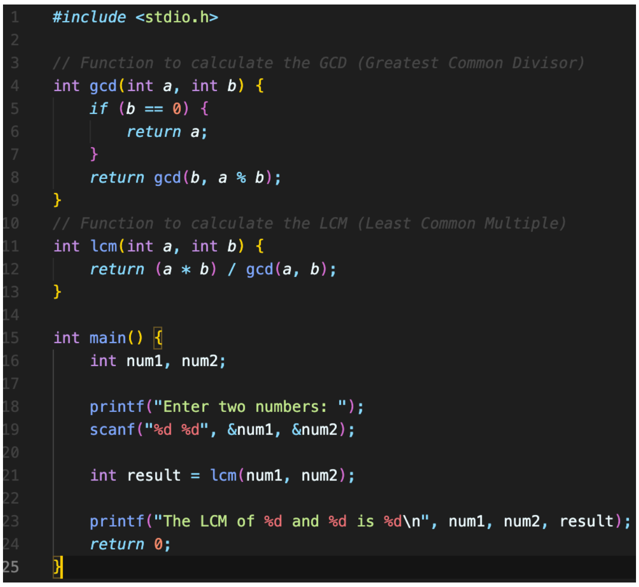


```c
#include <stdio.h>

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

int main() {
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    int result = gcd(num1, num2);
    printf("GCD of %d and %d is %d\n", num1, num2, result);
    return 0;
}
```

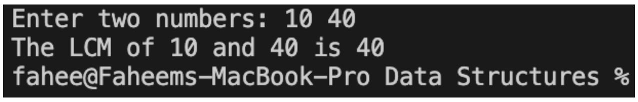


```
Enter two numbers: 10 40
GCD of 10 and 40 is 10
fahee@Faheems-MacBook-Pro Data Structures %
```

20. Write a program to find the LCM (Least Common Multiple) of two numbers.

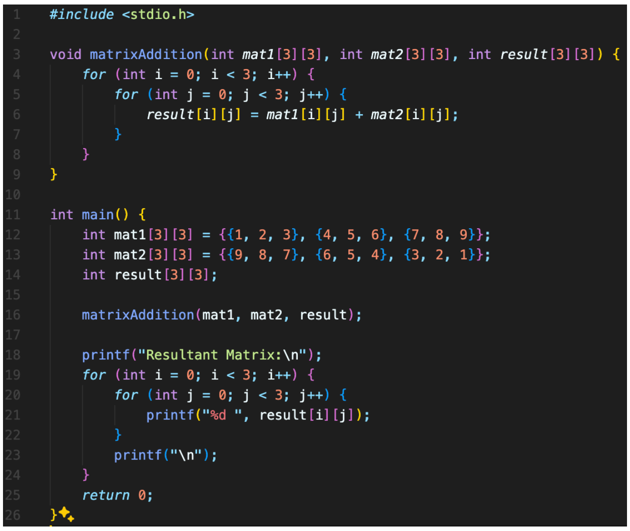


```c
#include <stdio.h>

// Function to calculate the GCD (Greatest Common Divisor)
int gcd(int a, int b) {
    if (b == 0) {
        return a;
    }
    return gcd(b, a % b);
}
// Function to calculate the LCM (Least Common Multiple)
int lcm(int a, int b) {
    return (a * b) / gcd(a, b);
}

int main() {
    int num1, num2;

    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    int result = lcm(num1, num2);

    printf("The LCM of %d and %d is %d\n", num1, num2, result);
    return 0;
}
```

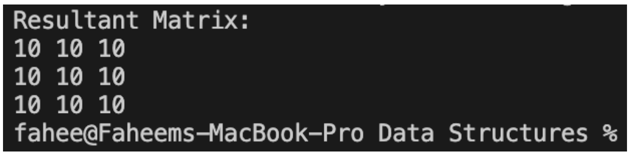


```
Enter two numbers: 10 40
The LCM of 10 and 40 is 40
fahee@Faheems-MacBook-Pro Data Structures %
```

21. Write a program to implement matrix addition.

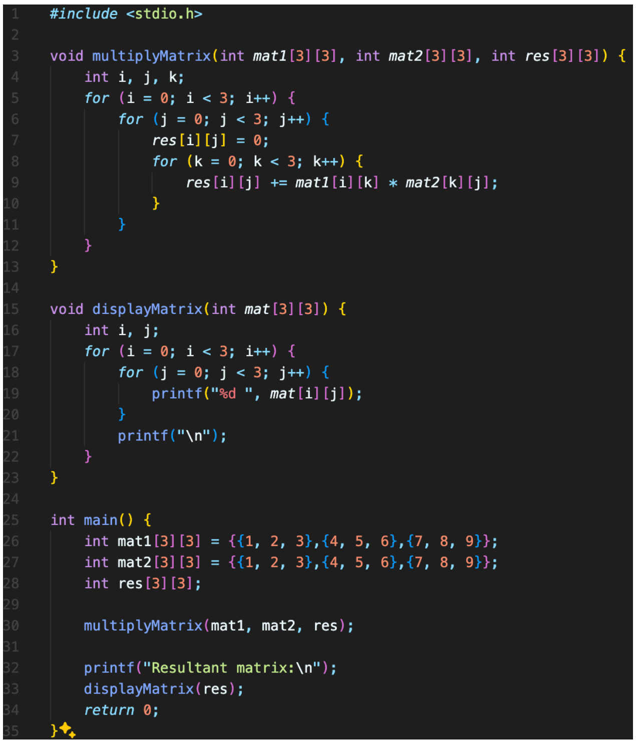


```c
#include <stdio.h>

void matrixAddition(int mat1[3][3], int mat2[3][3], int result[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            result[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
}

int main() {
    int mat1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int mat2[3][3] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
    int result[3][3];

    matrixAddition(mat1, mat2, result);

    printf("Resultant Matrix:\n");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ", result[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

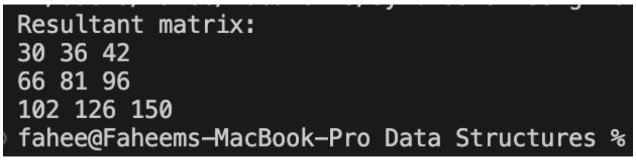


```
Resultant Matrix:
10 10 10
10 10 10
10 10 10
fahee@Faheems-MacBook-Pro Data Structures %
```

22. Write a program to implement matrix multiplication.

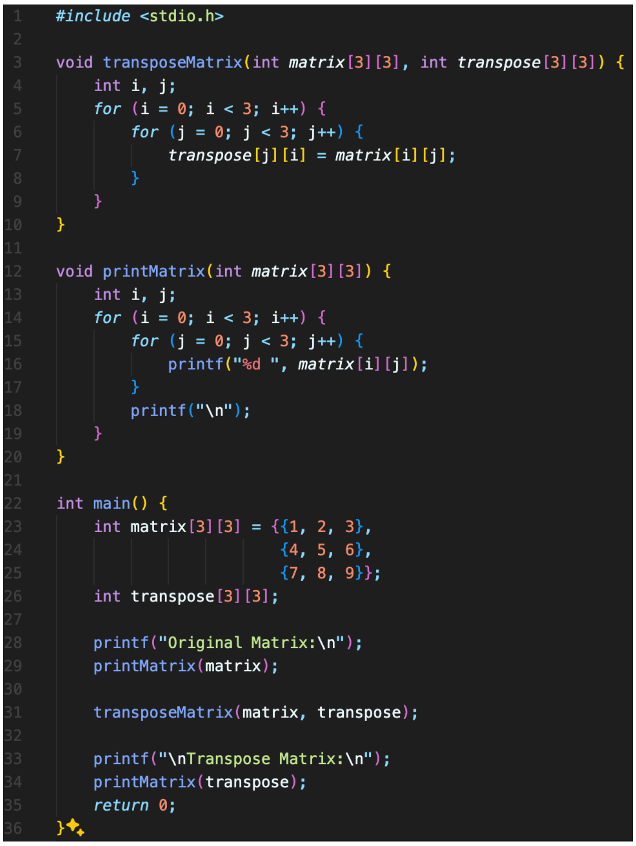


```c
#include <stdio.h>

void multiplyMatrix(int mat1[3][3], int mat2[3][3], int res[3][3]) {
    int i, j, k;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            res[i][j] = 0;
            for (k = 0; k < 3; k++) {
                res[i][j] += mat1[i][k] * mat2[k][j];
            }
        }
    }
}

void displayMatrix(int mat[3][3]) {
    int i, j;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }
}

int main() {
    int mat1[3][3] = {{1, 2, 3},{4, 5, 6},{7, 8, 9}};
    int mat2[3][3] = {{1, 2, 3},{4, 5, 6},{7, 8, 9}};
    int res[3][3];

    multiplyMatrix(mat1, mat2, res);

    printf("Resultant matrix:\n");
    displayMatrix(res);
    return 0;
}
```

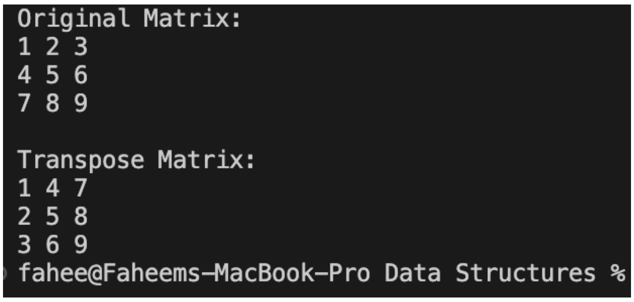


```
Resultant matrix:
30 36 42
66 81 96
102 126 150
fahee@Faheems-MacBook-Pro Data Structures %
```

23. Write a program to find the transpose of a matrix.

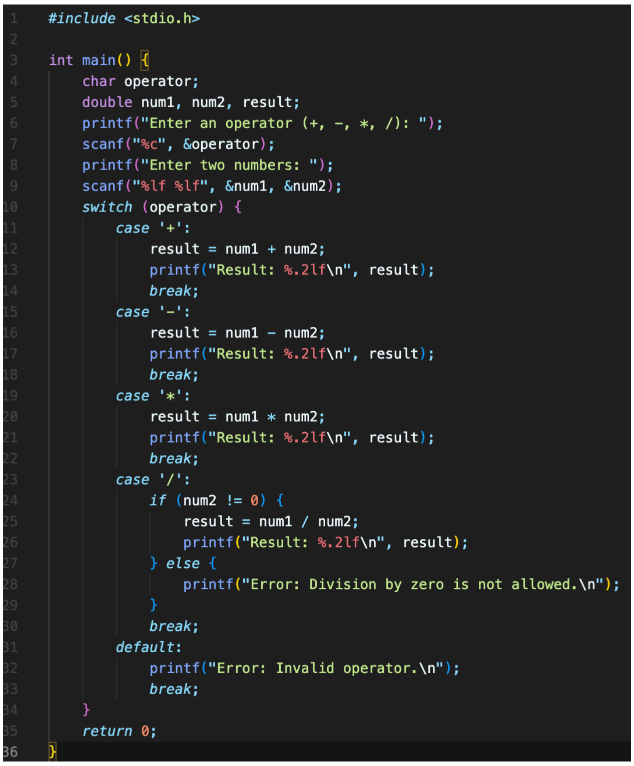


```c
#include <stdio.h>

void transposeMatrix(int matrix[3][3], int transpose[3][3]) {
    int i, j;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            transpose[j][i] = matrix[i][j];
        }
    }
}

void printMatrix(int matrix[3][3]) {
    int i, j;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
}

int main() {
    int matrix[3][3] = {{1, 2, 3},
                        {4, 5, 6},
                        {7, 8, 9}};
    int transpose[3][3];

    printf("Original Matrix:\n");
    printMatrix(matrix);

    transposeMatrix(matrix, transpose);

    printf("\nTranspose Matrix:\n");
    printMatrix(transpose);
    return 0;
}
```

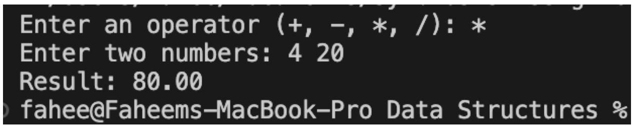


```
Original Matrix:
1 2 3
4 5 6
7 8 9

Transpose Matrix:
1 4 7
2 5 8
3 6 9
fahee@Faheems-MacBook-Pro Data Structures %
```

24. Write a program to implement a simple calculator using switch case.

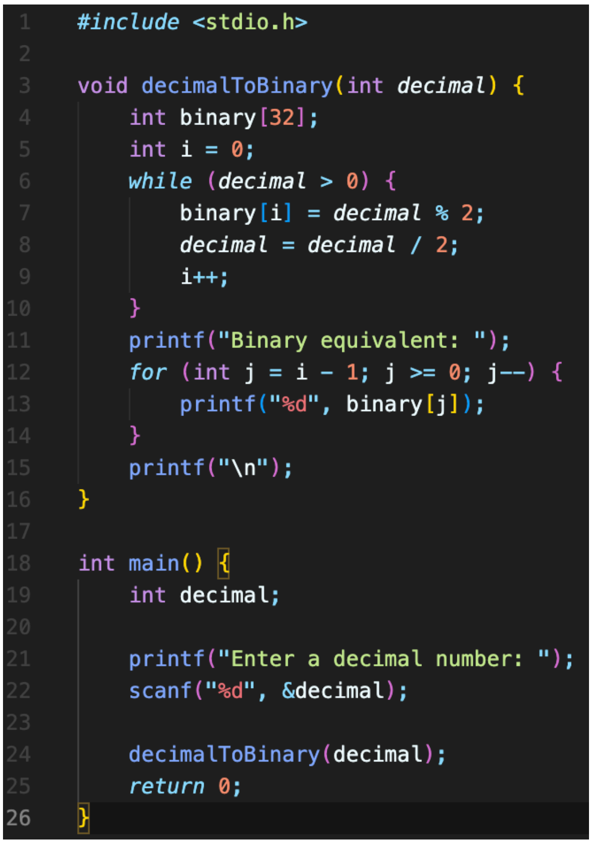


```c
#include <stdio.h>

int main() {
    char operator;
    double num1, num2, result;
    printf("Enter an operator (+, -, *, /): ");
    scanf("%c", &operator);
    printf("Enter two numbers: ");
    scanf("%lf %lf", &num1, &num2);
    switch (operator) {
        case '+':
            result = num1 + num2;
            printf("Result: %.2lf\n", result);
            break;
        case '-':
            result = num1 - num2;
            printf("Result: %.2lf\n", result);
            break;
        case '*':
            result = num1 * num2;
            printf("Result: %.2lf\n", result);
            break;
        case '/':
            if (num2 != 0) {
                result = num1 / num2;
                printf("Result: %.2lf\n", result);
            } else {
                printf("Error: Division by zero is not allowed.\n");
            }
            break;
        default:
            printf("Error: Invalid operator.\n");
            break;
    }
    return 0;
}
```

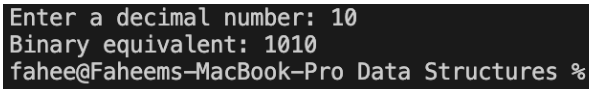


```
Enter an operator (+, -, *, /): *
Enter two numbers: 4 20
Result: 80.00
fahee@Faheems-MacBook-Pro Data Structures %
```

25. Write a program to convert a decimal number to its binary equivalent.

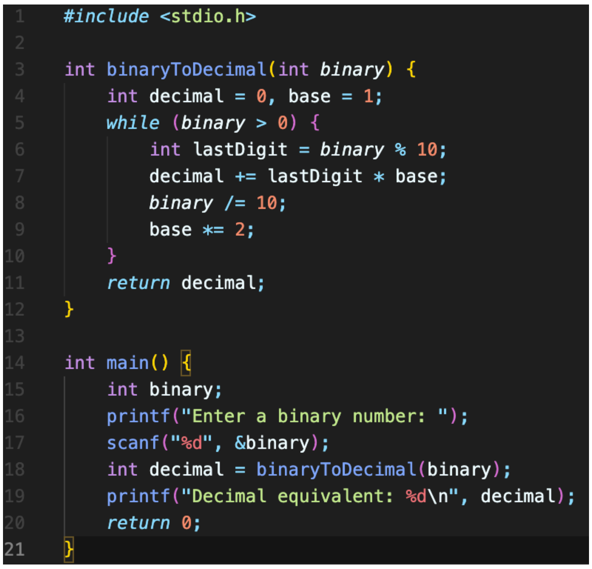


```c
#include <stdio.h>

void decimalToBinary(int decimal) {
    int binary[32];
    int i = 0;
    while (decimal > 0) {
        binary[i] = decimal % 2;
        decimal = decimal / 2;
        i++;
    }
    printf("Binary equivalent: ");
    for (int j = i - 1; j >= 0; j--) {
        printf("%d", binary[j]);
    }
    printf("\n");
}

int main() {
    int decimal;

    printf("Enter a decimal number: ");
    scanf("%d", &decimal);

    decimalToBinary(decimal);
    return 0;
}
```

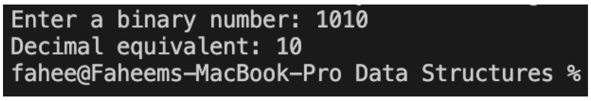


```
Enter a decimal number: 10
Binary equivalent: 1010
fahee@Faheems-MacBook-Pro Data Structures %
```

26. Write a program to convert a binary number to its decimal equivalent.

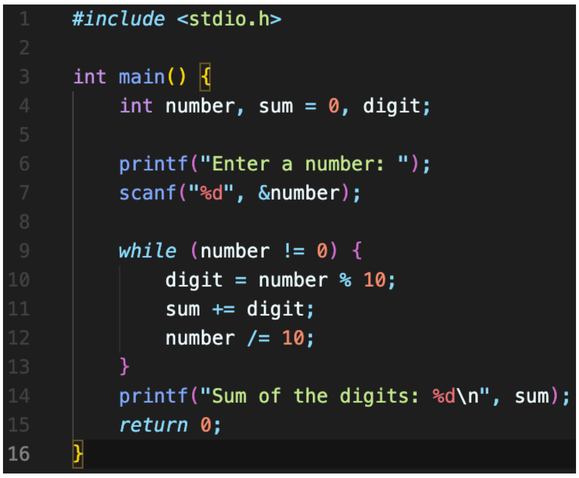


```c
#include <stdio.h>

int binaryToDecimal(int binary) {
    int decimal = 0, base = 1;
    while (binary > 0) {
        int lastDigit = binary % 10;
        decimal += lastDigit * base;
        binary /= 10;
        base *= 2;
    }
    return decimal;
}

int main() {
    int binary;
    printf("Enter a binary number: ");
    scanf("%d", &binary);
    int decimal = binaryToDecimal(binary);
    printf("Decimal equivalent: %d\n", decimal);
    return 0;
}
```

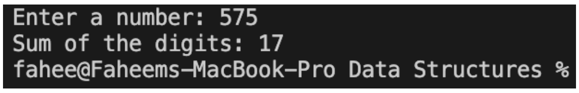


```
Enter a binary number: 1010
Decimal equivalent: 10
fahee@Faheems-MacBook-Pro Data Structures %
```

27. Write a program to find the sum of the digits of a given number.

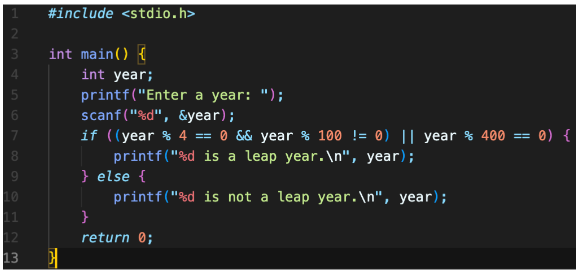


```c
#include <stdio.h>

int main() {
    int number, sum = 0, digit;

    printf("Enter a number: ");
    scanf("%d", &number);

    while (number != 0) {
        digit = number % 10;
        sum += digit;
        number /= 10;
    }
    printf("Sum of the digits: %d\n", sum);
    return 0;
}
```

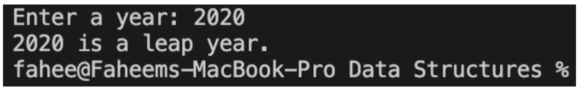


```
Enter a number: 575
Sum of the digits: 17
fahee@Faheems-MacBook-Pro Data Structures %
```

28. Write a program to check if a given year is a leap year.

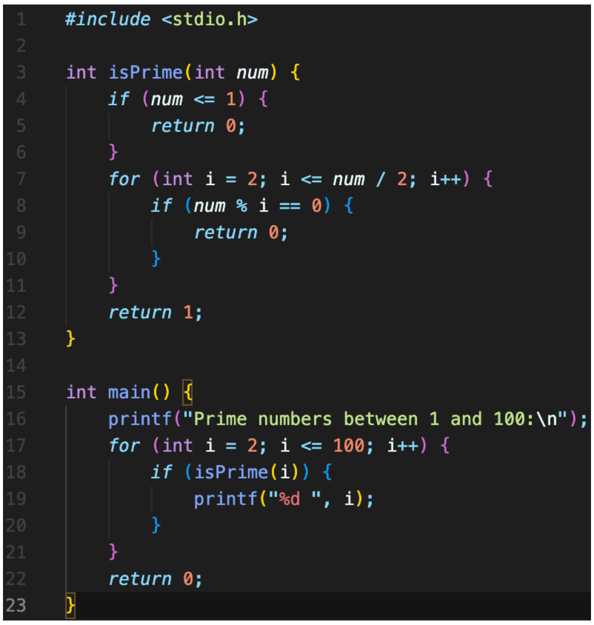


```c
#include <stdio.h>

int main() {
    int year;
    printf("Enter a year: ");
    scanf("%d", &year);
    if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
        printf("%d is a leap year.\n", year);
    } else {
        printf("%d is not a leap year.\n", year);
    }
    return 0;
}
```

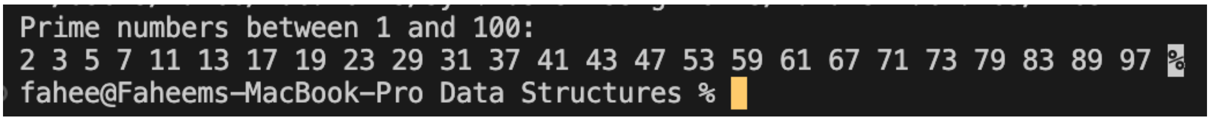


```
Enter a year: 2020
2020 is a leap year.
fahee@Faheems-MacBook-Pro Data Structures %
```

29. Write a program to print the prime numbers between 1 and 100.

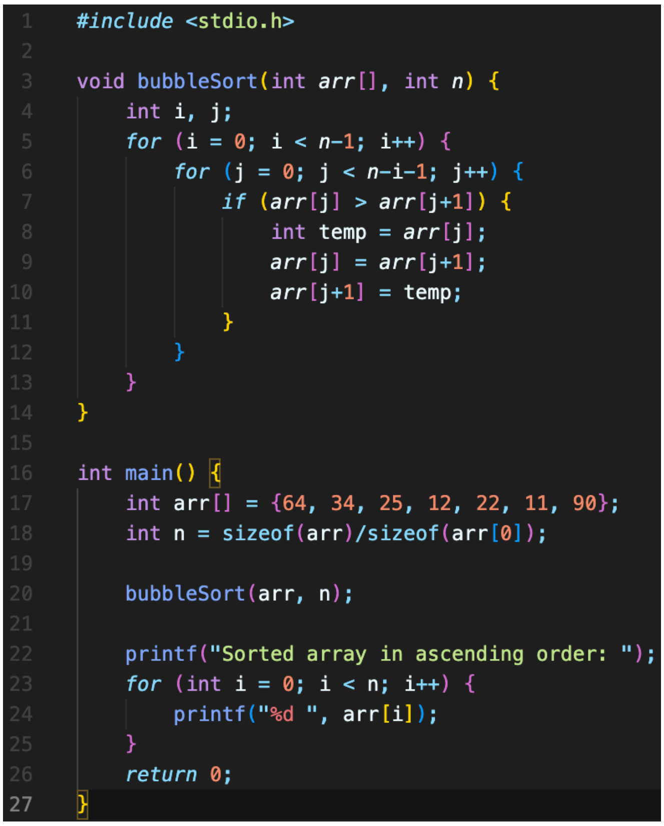


```c
#include <stdio.h>

int isPrime(int num) {
    if (num <= 1) {
        return 0;
    }
    for (int i = 2; i <= num / 2; i++) {
        if (num % i == 0) {
            return 0;
        }
    }
    return 1;
}

int main() {
    printf("Prime numbers between 1 and 100:\n");
    for (int i = 2; i <= 100; i++) {
        if (isPrime(i)) {
            printf("%d ", i);
        }
    }
    return 0;
}
```

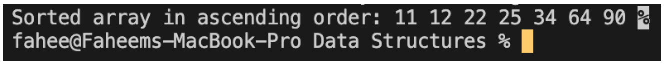


```
Prime numbers between 1 and 100:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 %
fahee@Faheems-MacBook-Pro Data Structures %
```

30. Write a program to implement bubble sort.



```c
#include <stdio.h>

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

int main() {
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);

    bubbleSort(arr, n);

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



```
Sorted array in ascending order: 11 12 22 25 34 64 90 %
fahee@Faheems-MacBook-Pro Data Structures %
```