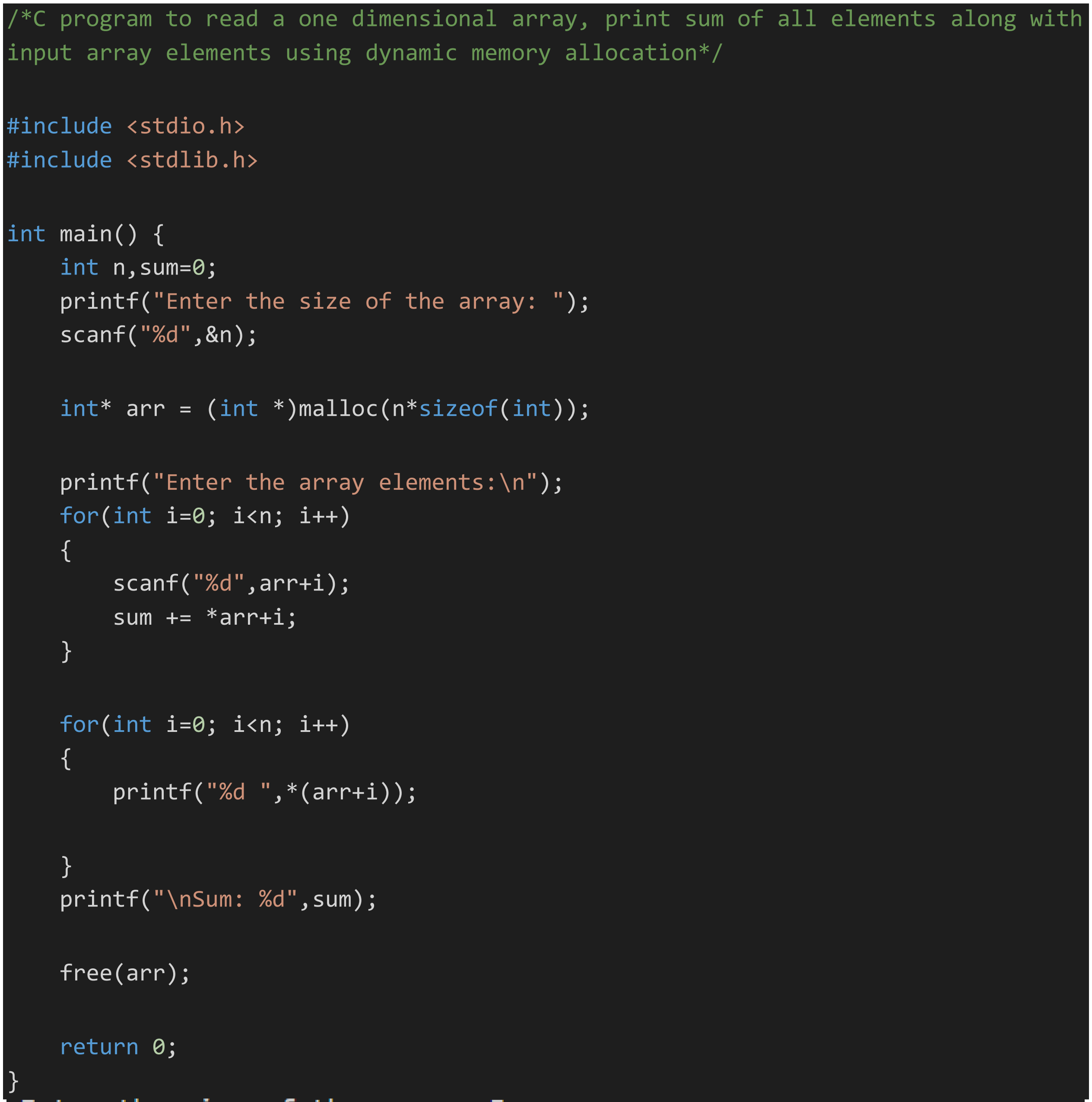
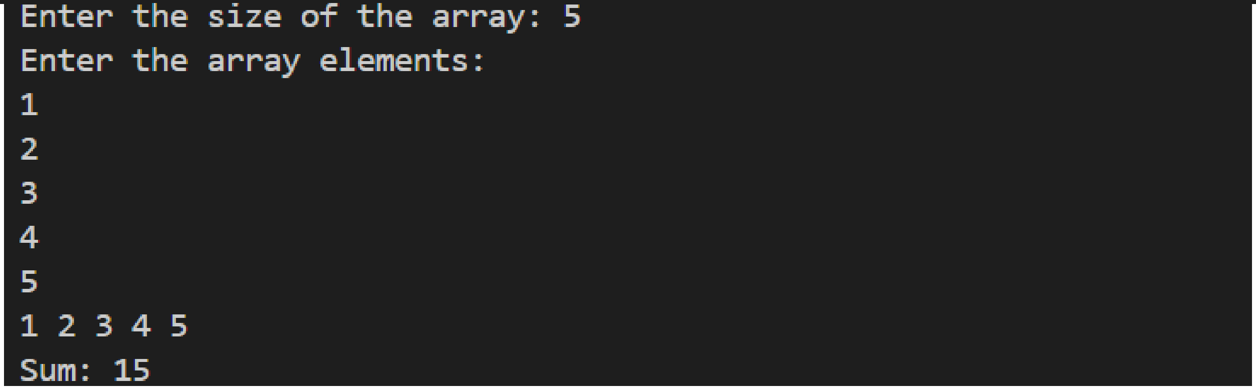
20.



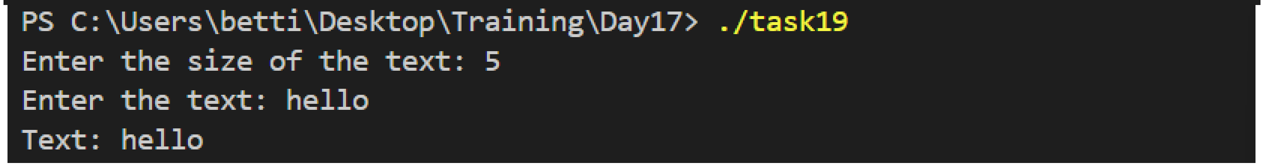
```c
/*C program to read a one dimensional array, print sum of all elements along with
input array elements using dynamic memory allocation*/

#include <stdio.h>
#include <stdlib.h>

int main() {
    int n,sum=0;
    printf("Enter the size of the array: ");
    scanf("%d",&n);

    int* arr = (int *)malloc(n*sizeof(int));

    printf("Enter the array elements:\n");
    for(int i=0; i<n; i++)
    {
        scanf("%d",arr+i);
        sum += *arr+i;
    }

    for(int i=0; i<n; i++)
    {
        printf("%d ",*(arr+i));

    }
    printf("\nSum: %d",sum);

    free(arr);

    return 0;
}
```



```
Enter the size of the array: 5
Enter the array elements:
1
2
3
4
5
1 2 3 4 5
Sum: 15
```

19.

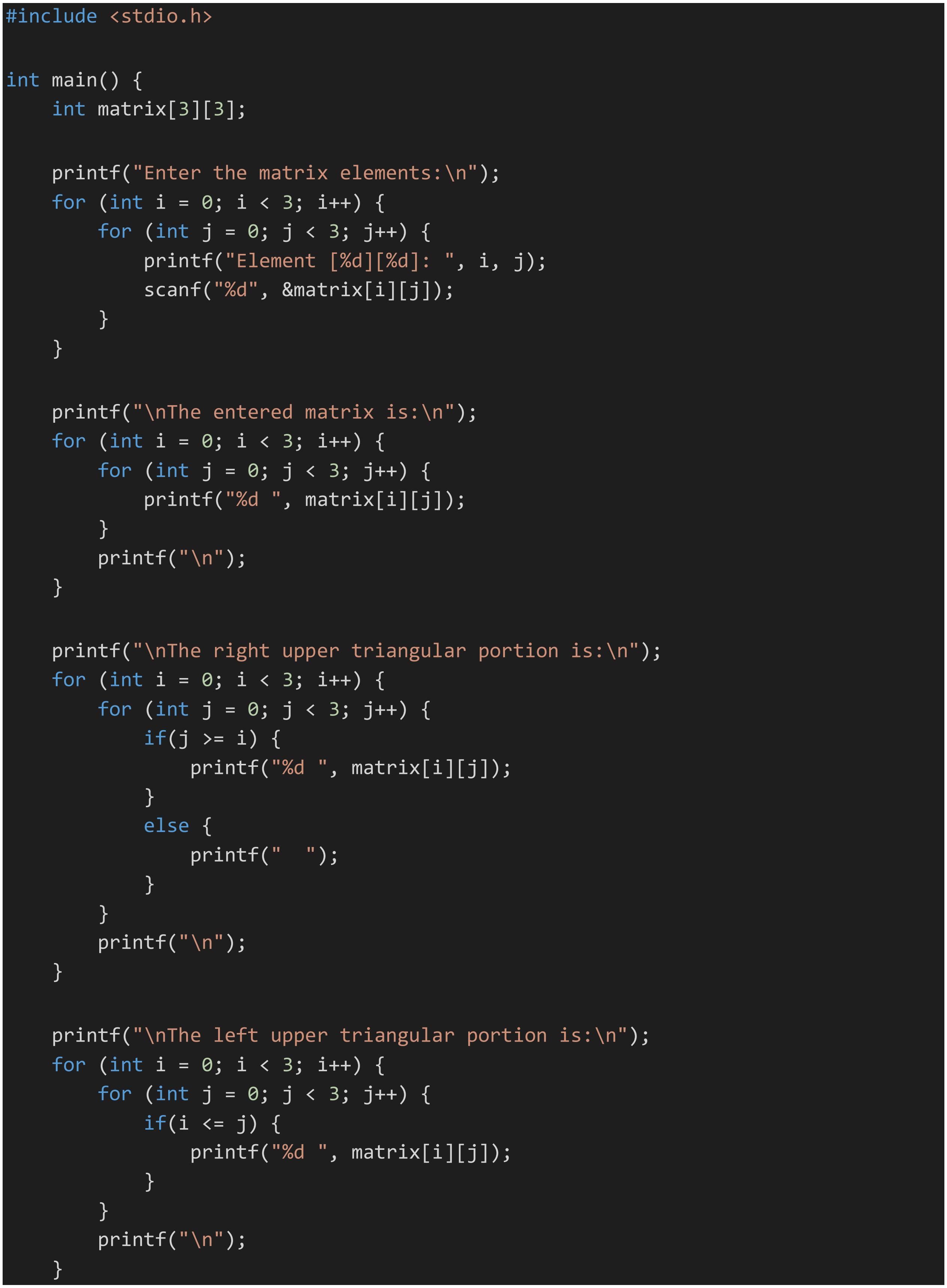
```c
/*C program to input and print text using dynamic memory allocation*/

#include <stdio.h>
#include <stdlib.h>

int main() {
    char *text;
    int size;

    // Get the size of the text
    printf("Enter the size of the text: ");
    scanf("%d", &size);

    // Dynamically allocate memory for the text
    text = (char *)malloc((size+1) * sizeof(char));

    if (text == NULL) {
        printf("Memory allocation failed.\n");
        return 1;
    }

    getchar();
    printf("Enter the text: ");
    fgets(text, size+1, stdin);

    printf("Text: %s", text);

    // Free the allocated memory
    free(text);

    return 0;
}
```



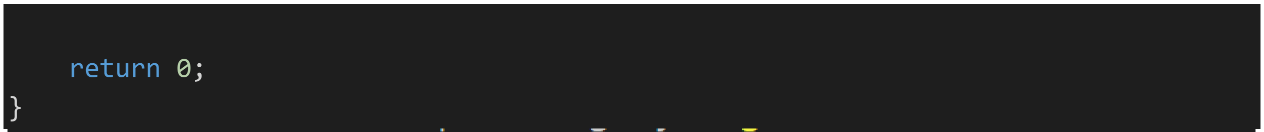
```
PS C:\Users\betti\Desktop\Training\Day17> ./task19
Enter the size of the text: 5
Enter the text: hello
Text: hello
```

18.

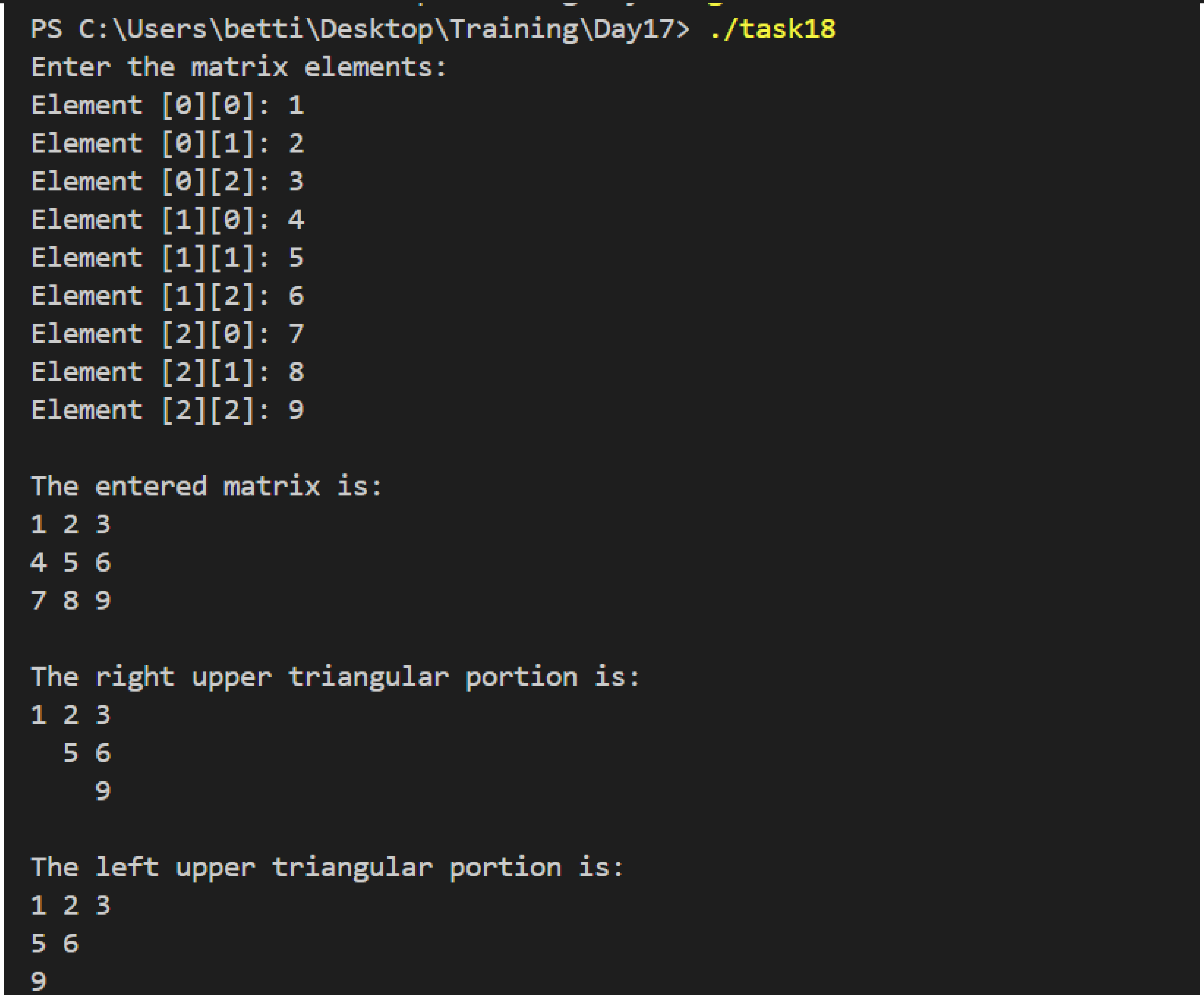
```c
/*C program to print the upper triangular portion of 3x3 matrix*/

```



```c
#include <stdio.h>

int main() {
    int matrix[3][3];

    printf("Enter the matrix elements:\n");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &matrix[i][j]);
        }
    }

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

    printf("\nThe right upper triangular portion is:\n");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if(j >= i) {
                printf("%d ", matrix[i][j]);
            }
            else {
                printf("  ");
            }
        }
        printf("\n");
    }

    printf("\nThe left upper triangular portion is:\n");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if(i <= j) {
                printf("%d ", matrix[i][j]);
            }
        }
        printf("\n");
    }
```

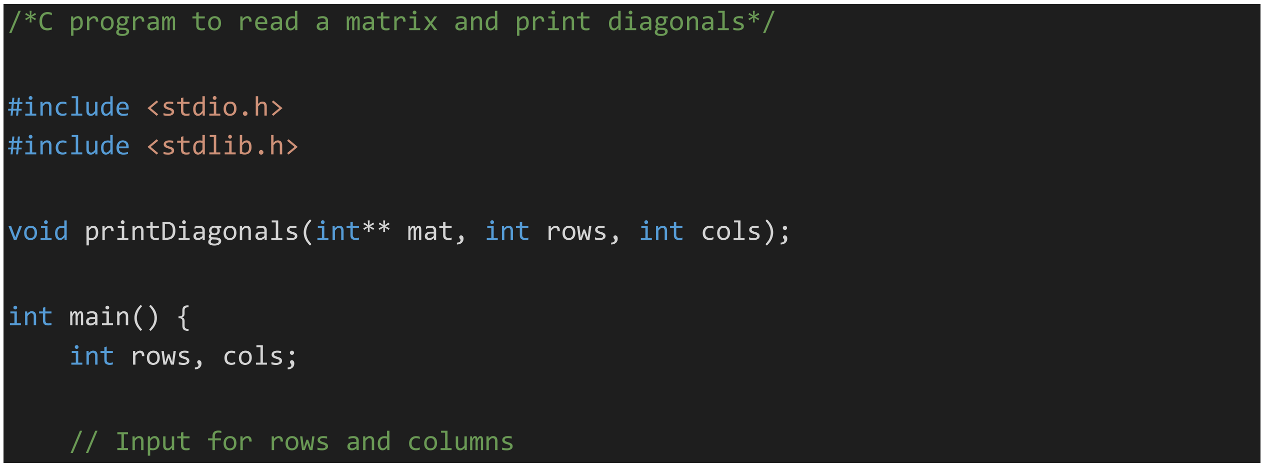


```
    return 0;
}
```

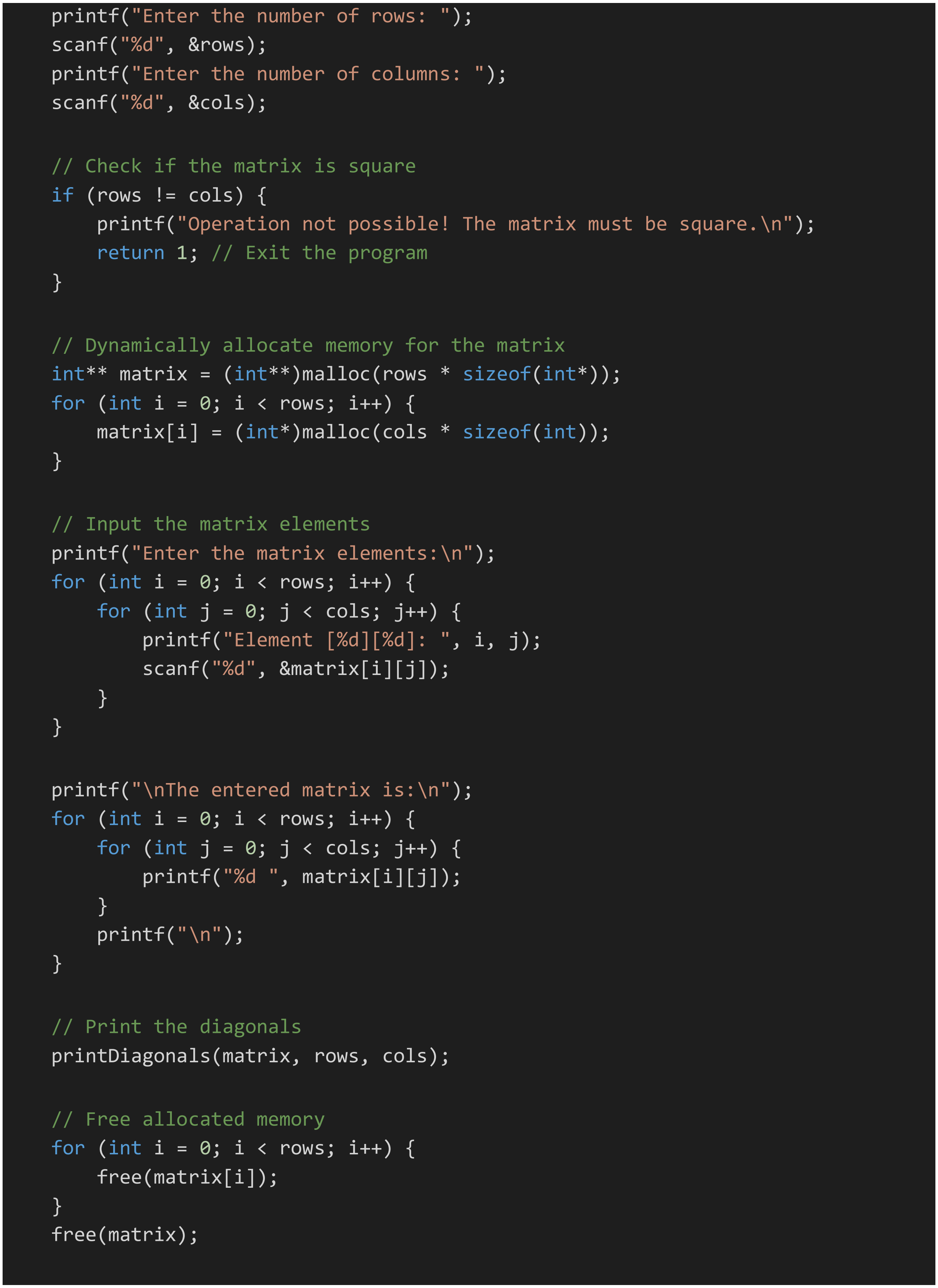


```
PS C:\Users\betti\Desktop\Training\Day17> ./task18
Enter the matrix elements:
Element [0][0]: 1
Element [0][1]: 2
Element [0][2]: 3
Element [1][0]: 4
Element [1][1]: 5
Element [1][2]: 6
Element [2][0]: 7
Element [2][1]: 8
Element [2][2]: 9

The entered matrix is:
1 2 3
4 5 6
7 8 9

The right upper triangular portion is:
1 2 3
  5 6
    9

The left upper triangular portion is:
1 2 3
5 6
9
```

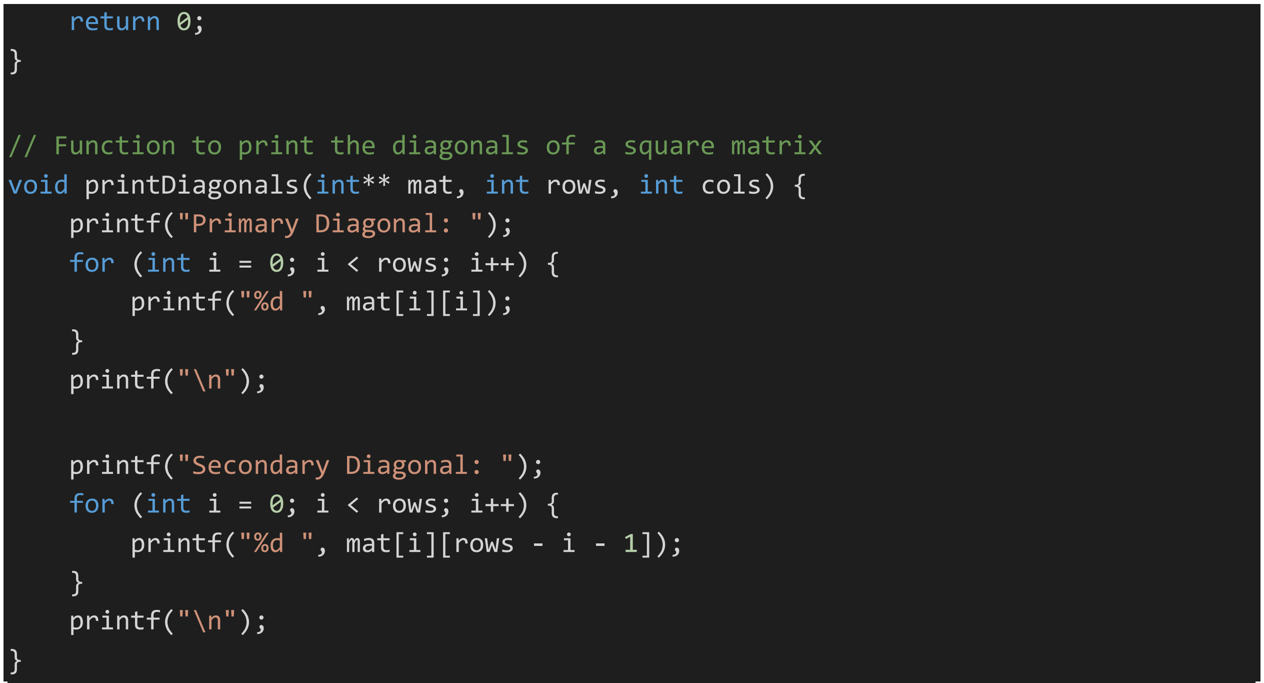
17.



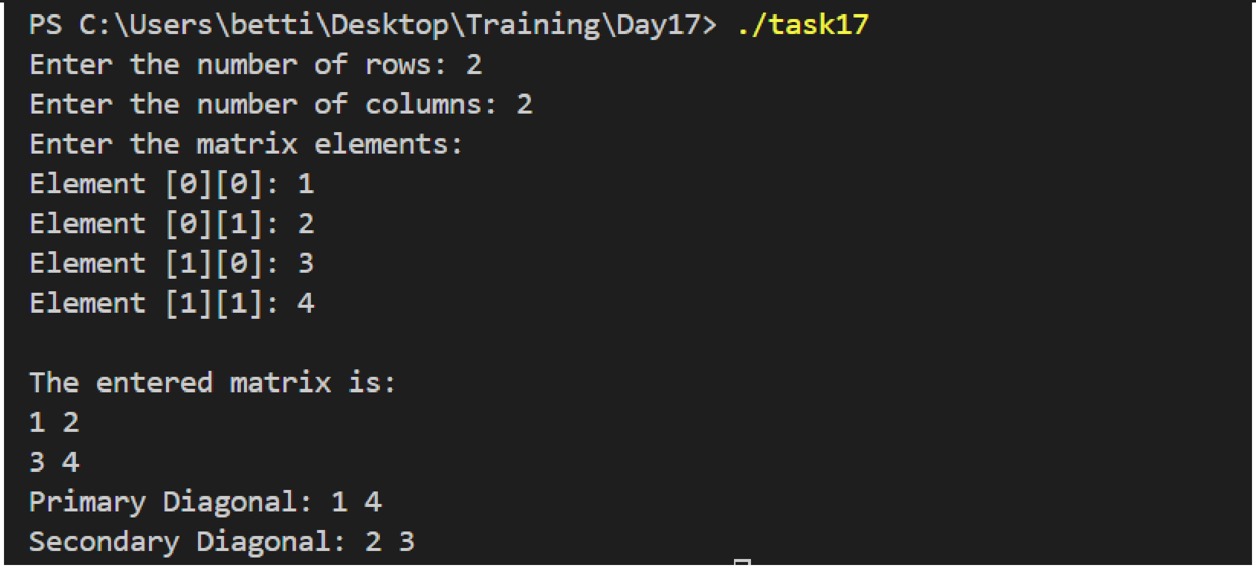
```
/*C program to read a matrix and print diagonals*/

#include <stdio.h>
#include <stdlib.h>

void printDiagonals(int** mat, int rows, int cols);

int main() {
    int rows, cols;

    // Input for rows and columns
```



```c
    printf("Enter the number of rows: ");
    scanf("%d", &rows);
    printf("Enter the number of columns: ");
    scanf("%d", &cols);

    // Check if the matrix is square
    if (rows != cols) {
        printf("Operation not possible! The matrix must be square.\n");
        return 1; // Exit the program
    }

    // Dynamically allocate memory for the matrix
    int** matrix = (int**)malloc(rows * sizeof(int*));
    for (int i = 0; i < rows; i++) {
        matrix[i] = (int*)malloc(cols * sizeof(int));
    }

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

    printf("\nThe entered matrix is:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }

    // Print the diagonals
    printDiagonals(matrix, rows, cols);

    // Free allocated memory
    for (int i = 0; i < rows; i++) {
        free(matrix[i]);
    }
    free(matrix);
```

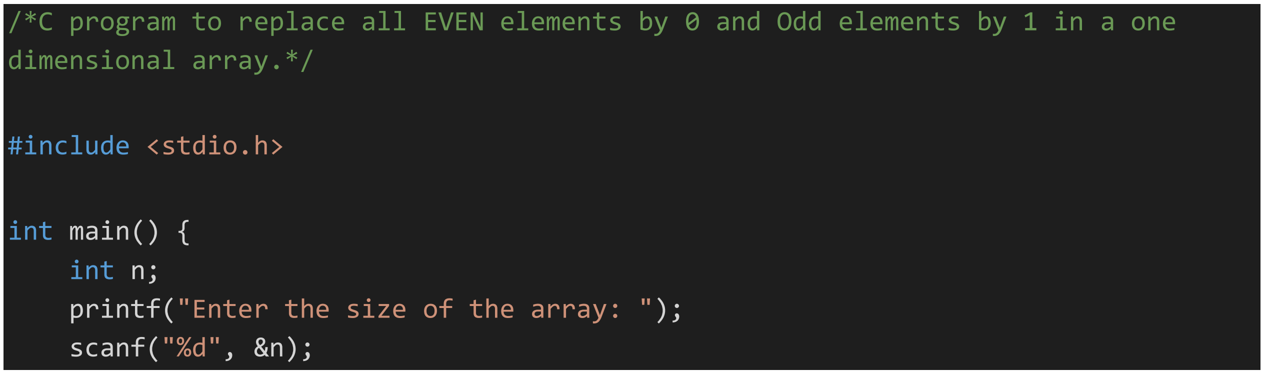


```c
    return 0;
}

// Function to print the diagonals of a square matrix
void printDiagonals(int** mat, int rows, int cols) {
    printf("Primary Diagonal: ");
    for (int i = 0; i < rows; i++) {
        printf("%d ", mat[i][i]);
    }
    printf("\n");

    printf("Secondary Diagonal: ");
    for (int i = 0; i < rows; i++) {
        printf("%d ", mat[i][rows - i - 1]);
    }
    printf("\n");
}
```

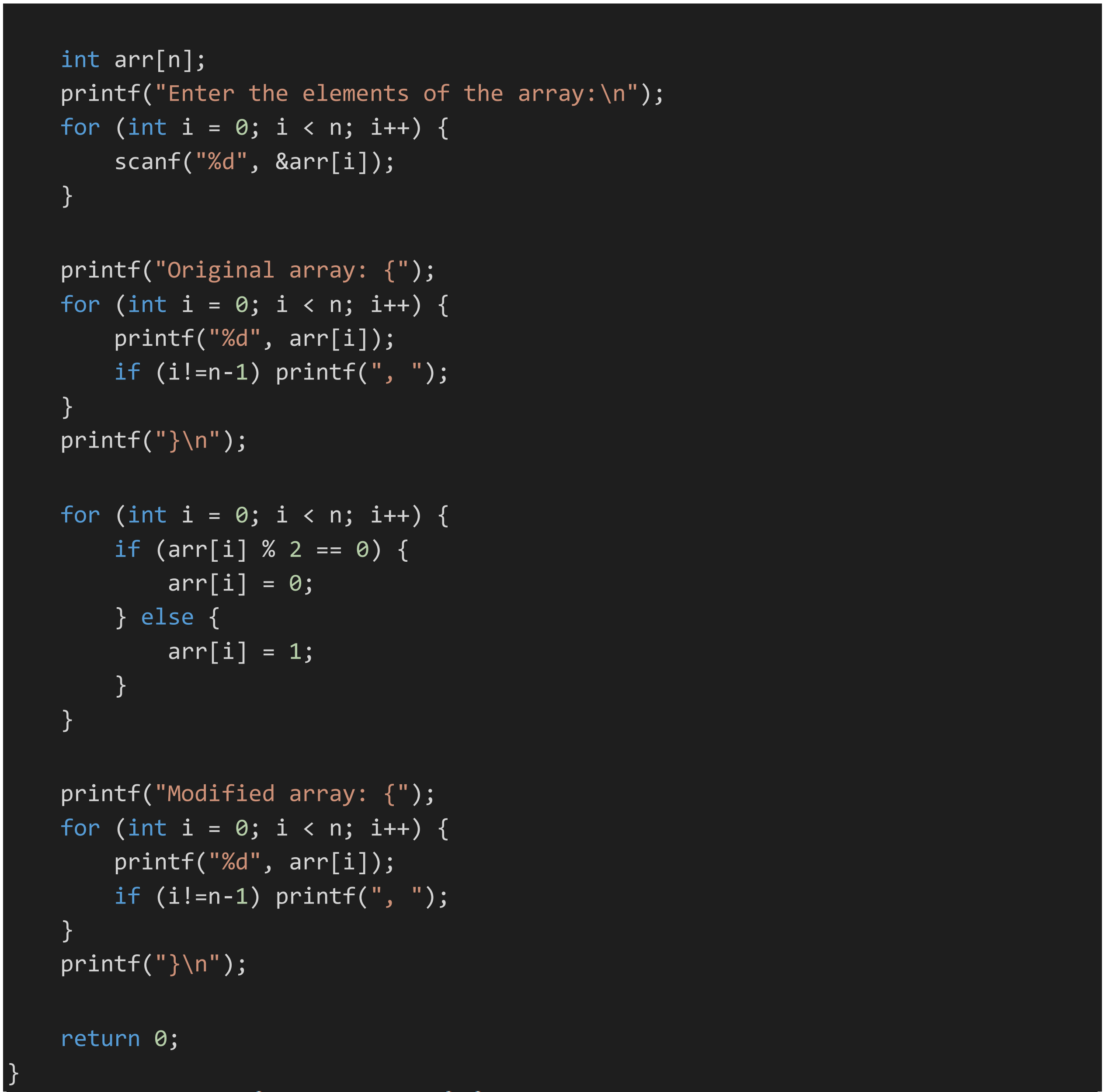


```
PS C:\Users\betti\Desktop\Training\Day17> ./task17
Enter the number of rows: 2
Enter the number of columns: 2
Enter the matrix elements:
Element [0][0]: 1
Element [0][1]: 2
Element [1][0]: 3
Element [1][1]: 4

The entered matrix is:
1 2
3 4
Primary Diagonal: 1 4
Secondary Diagonal: 2 3
```

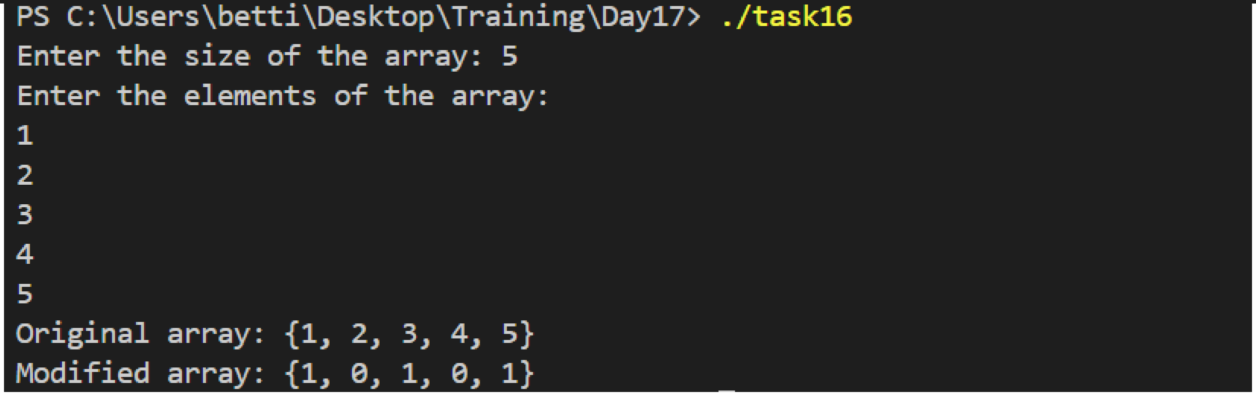
16.



```c
/*C program to replace all EVEN elements by 0 and Odd elements by 1 in a one
dimensional array.*/

#include <stdio.h>

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

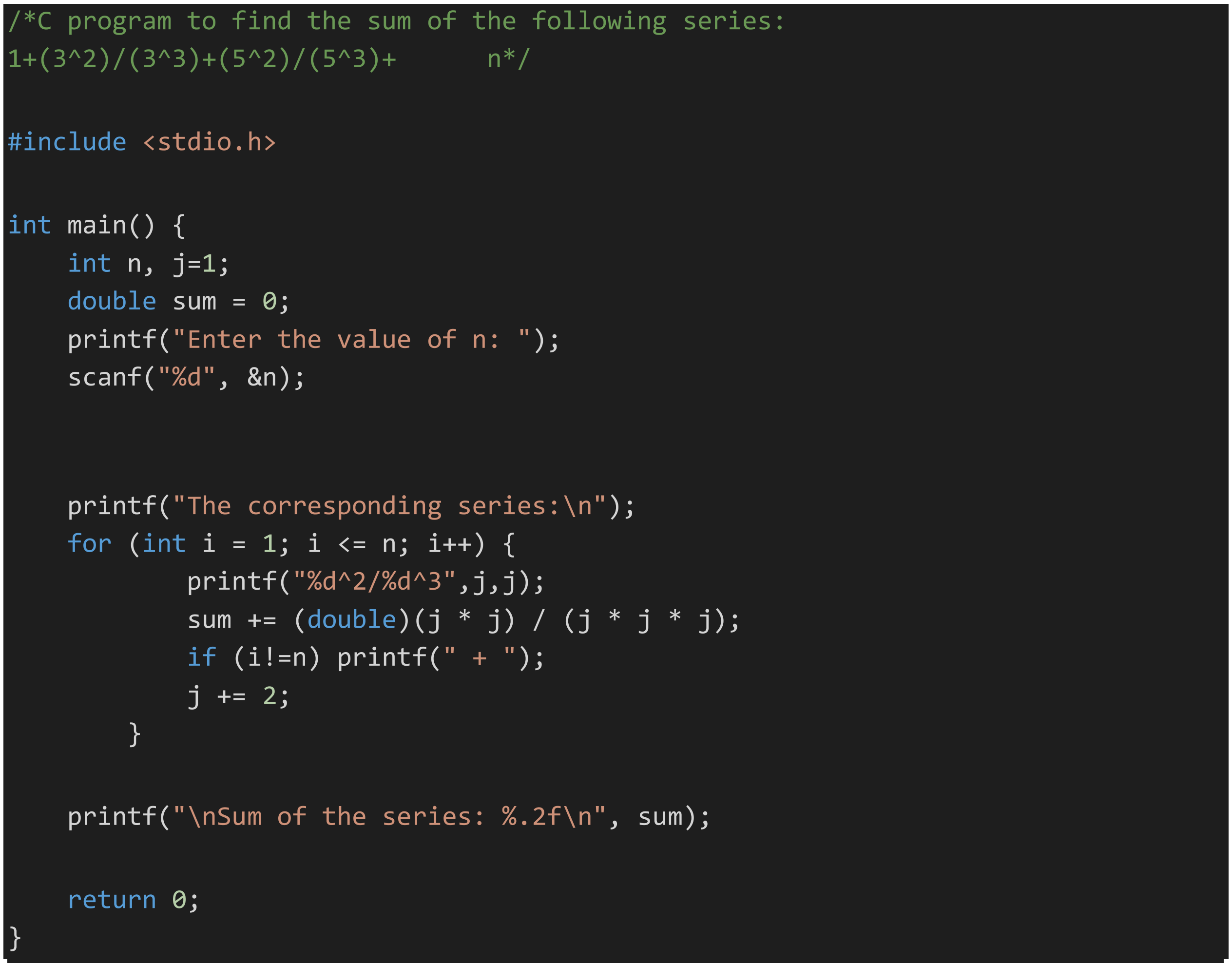


```
    int arr[n];
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Original array: {");
    for (int i = 0; i < n; i++) {
        printf("%d", arr[i]);
        if (i!=n-1) printf(", ");
    }
    printf("}\n");

    for (int i = 0; i < n; i++) {
        if (arr[i] % 2 == 0) {
            arr[i] = 0;
        } else {
            arr[i] = 1;
        }
    }

    printf("Modified array: {");
    for (int i = 0; i < n; i++) {
        printf("%d", arr[i]);
        if (i!=n-1) printf(", ");
    }
    printf("}\n");

    return 0;
}
```

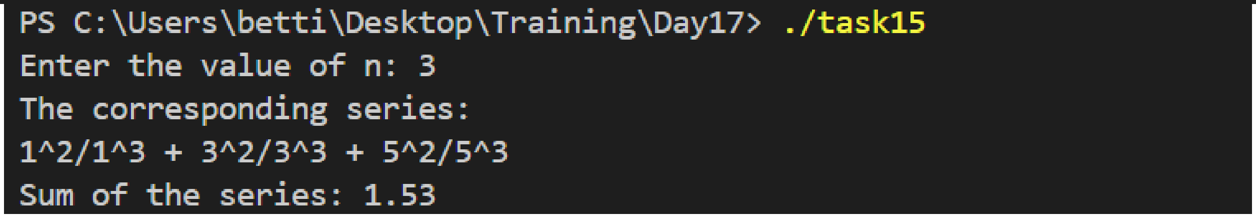


```
PS C:\Users\betti\Desktop\Training\Day17> ./task16
Enter the size of the array: 5
Enter the elements of the array:
1
2
3
4
5
Original array: {1, 2, 3, 4, 5}
Modified array: {1, 0, 1, 0, 1}
```

15.

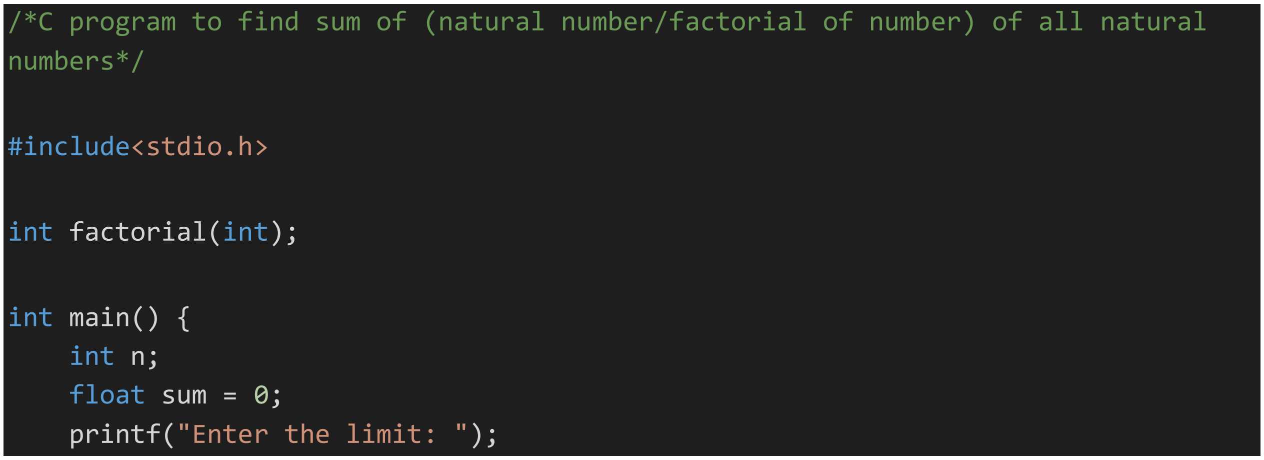


```c
/*C program to find the sum of the following series:
1+(3^2)/(3^3)+(5^2)/(5^3)+      n*/

#include <stdio.h>

int main() {
    int n, j=1;
    double sum = 0;
    printf("Enter the value of n: ");
    scanf("%d", &n);


    printf("The corresponding series:\n");
    for (int i = 1; i <= n; i++) {
            printf("%d^2/%d^3",j,j);
            sum += (double)(j * j) / (j * j * j);
            if (i!=n) printf(" + ");
            j += 2;
        }

    printf("\nSum of the series: %.2f\n", sum);

    return 0;
}
```

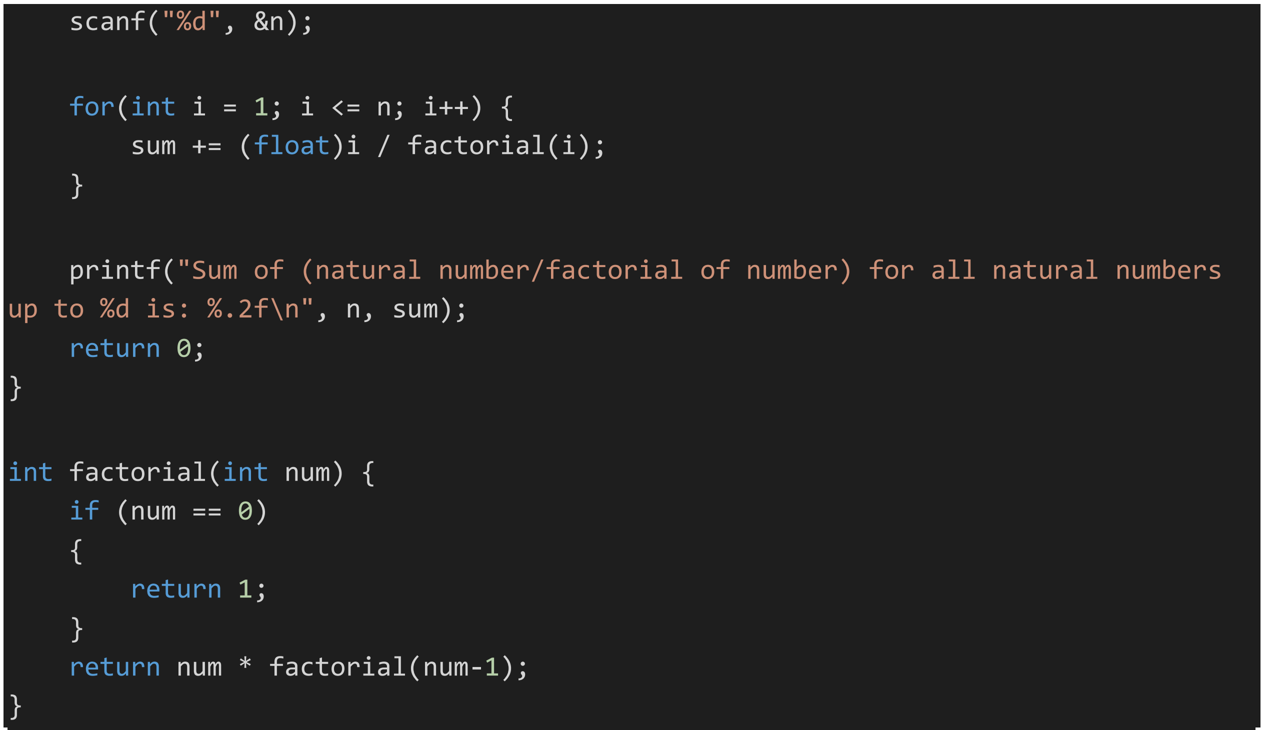


```
PS C:\Users\betti\Desktop\Training\Day17> ./task15
Enter the value of n: 3
The corresponding series:
1^2/1^3 + 3^2/3^3 + 5^2/5^3
Sum of the series: 1.53
```

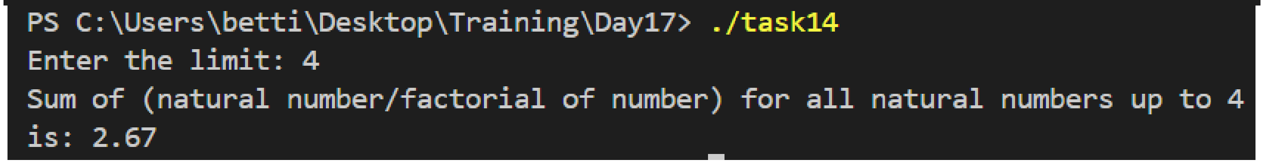
14.



```c
/*C program to find sum of (natural number/factorial of number) of all natural
numbers*/

#include<stdio.h>

int factorial(int);

int main() {
    int n;
    float sum = 0;
    printf("Enter the limit: ");
```

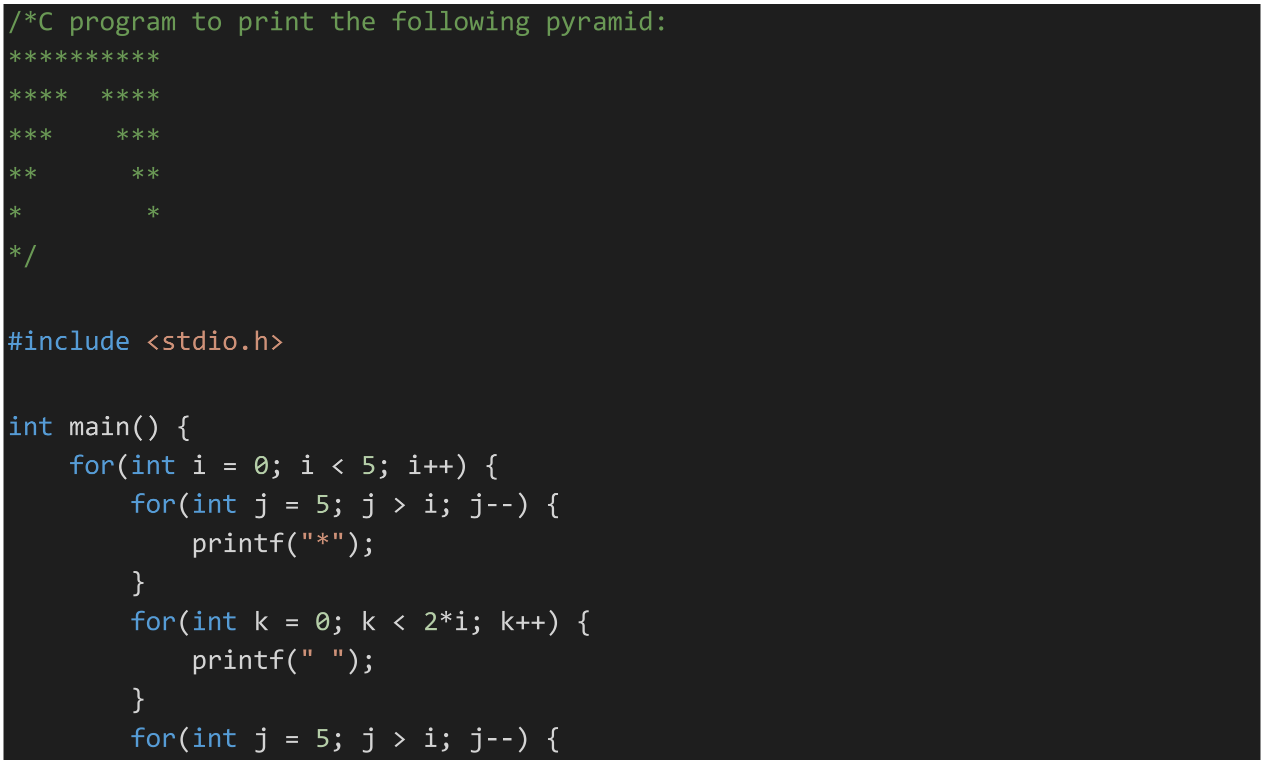


```c
    scanf("%d", &n);

    for(int i = 1; i <= n; i++) {
        sum += (float)i / factorial(i);
    }

    printf("Sum of (natural number/factorial of number) for all natural numbers
up to %d is: %.2f\n", n, sum);
    return 0;
}

int factorial(int num) {
    if (num == 0)
    {
        return 1;
    }
    return num * factorial(num-1);
}
```

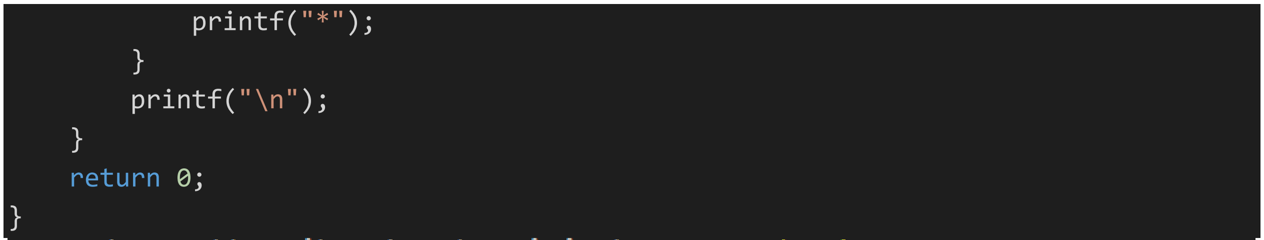


```
PS C:\Users\betti\Desktop\Training\Day17> ./task14
Enter the limit: 4
Sum of (natural number/factorial of number) for all natural numbers up to 4
is: 2.67
```

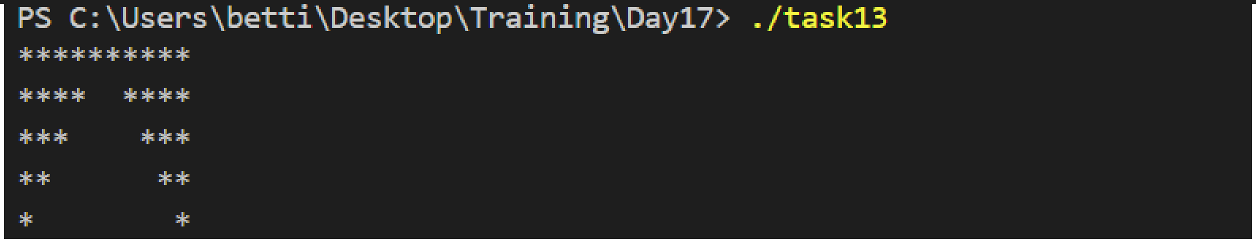
13.



```c
/*C program to print the following pyramid:
**********
****  ****
***    ***
**      **
*        *
*/

#include <stdio.h>

int main() {
    for(int i = 0; i < 5; i++) {
        for(int j = 5; j > i; j--) {
            printf("*");
        }
        for(int k = 0; k < 2*i; k++) {
            printf(" ");
        }
        for(int j = 5; j > i; j--) {
```

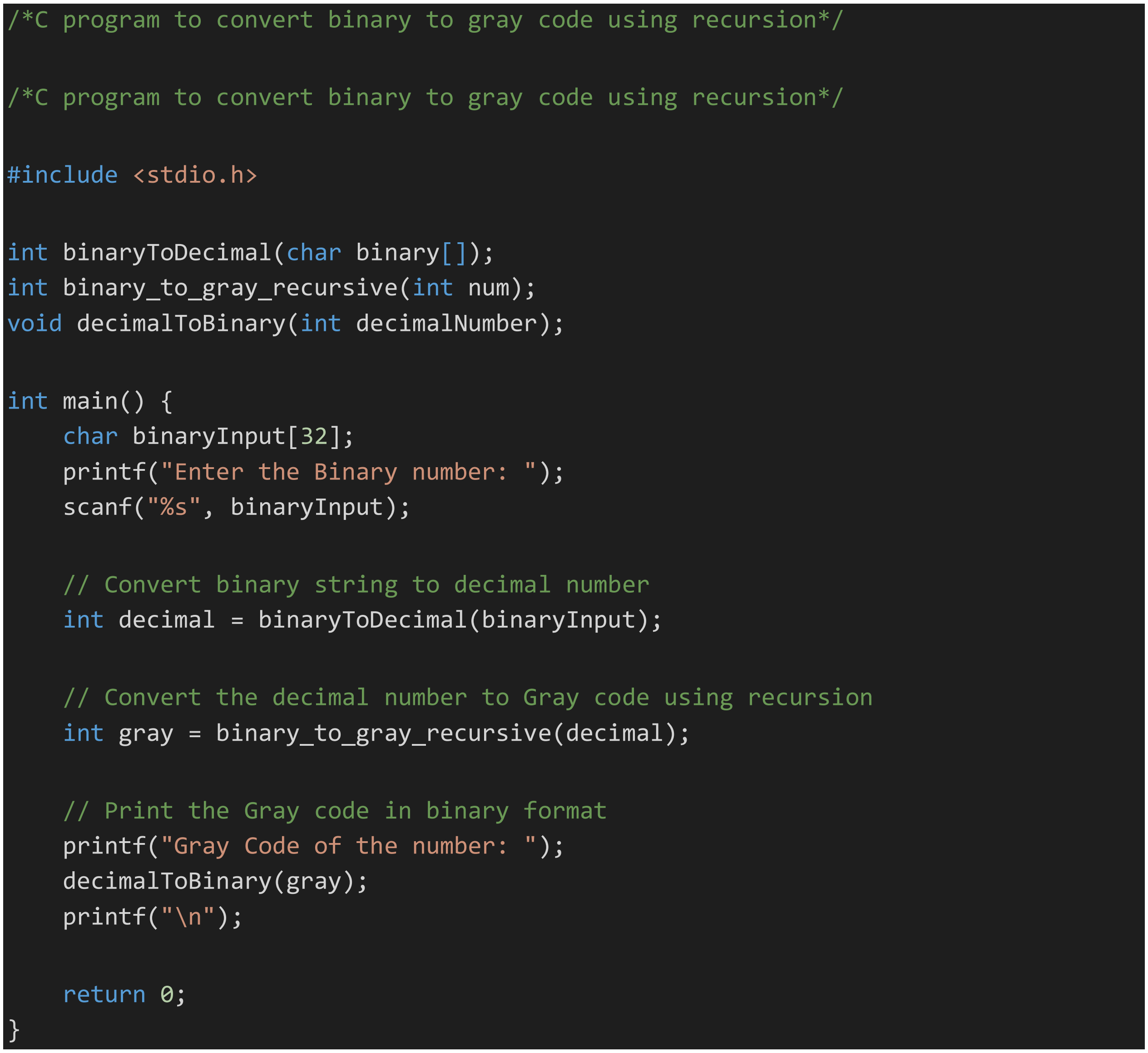


```
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

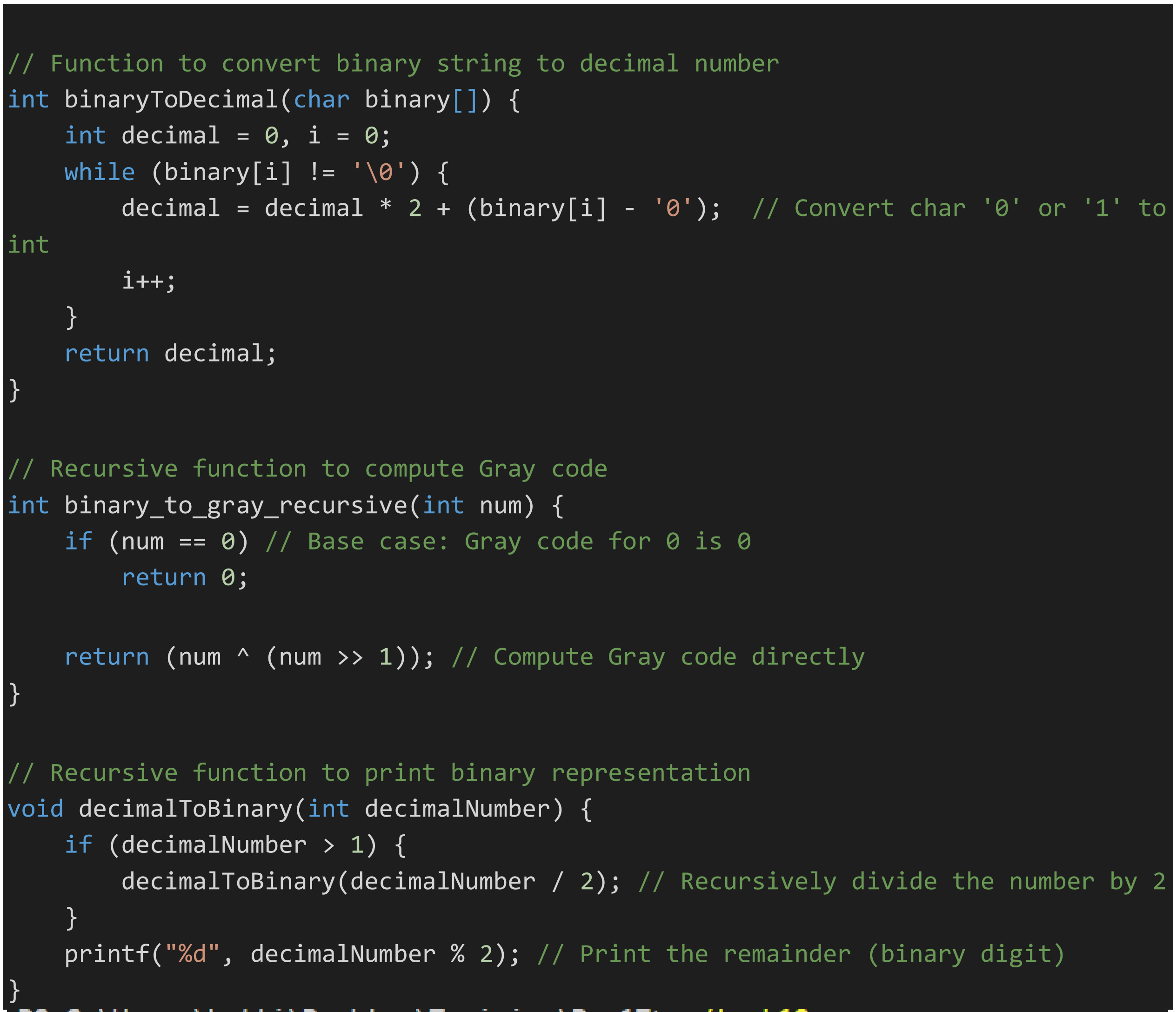


```
PS C:\Users\betti\Desktop\Training\Day17> ./task13
**********
****  ****
***    ***
**      **
*        *
```

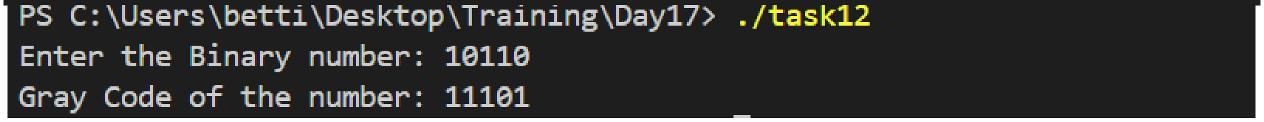
12.



```
/*C program to convert binary to gray code using recursion*/

/*C program to convert binary to gray code using recursion*/

#include <stdio.h>

int binaryToDecimal(char binary[]);
int binary_to_gray_recursive(int num);
void decimalToBinary(int decimalNumber);

int main() {
    char binaryInput[32];
    printf("Enter the Binary number: ");
    scanf("%s", binaryInput);

    // Convert binary string to decimal number
    int decimal = binaryToDecimal(binaryInput);

    // Convert the decimal number to Gray code using recursion
    int gray = binary_to_gray_recursive(decimal);

    // Print the Gray code in binary format
    printf("Gray Code of the number: ");
    decimalToBinary(gray);
    printf("\n");

    return 0;
}
```

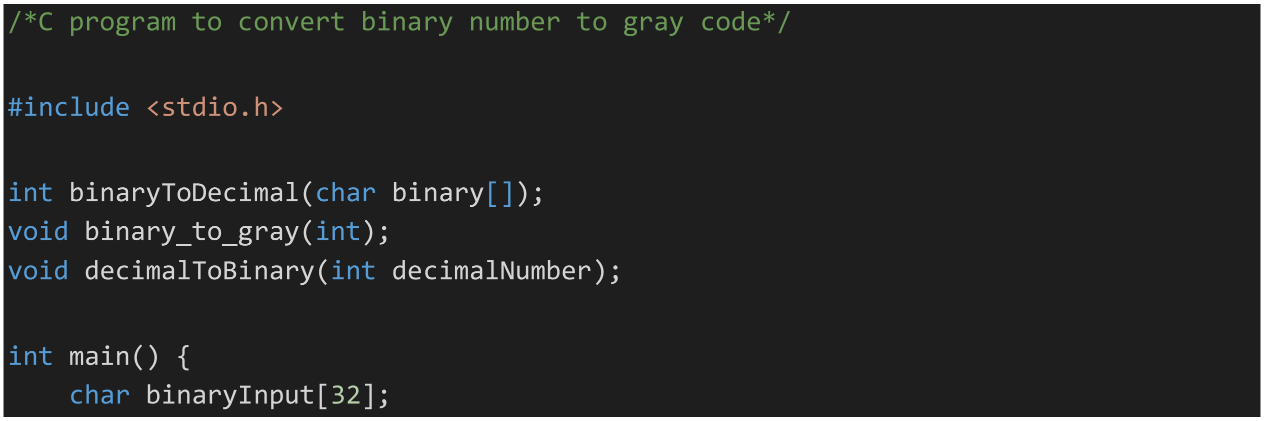


```c
// Function to convert binary string to decimal number
int binaryToDecimal(char binary[]) {
    int decimal = 0, i = 0;
    while (binary[i] != '\0') {
        decimal = decimal * 2 + (binary[i] - '0');  // Convert char '0' or '1' to
int
        i++;
    }
    return decimal;
}

// Recursive function to compute Gray code
int binary_to_gray_recursive(int num) {
    if (num == 0) // Base case: Gray code for 0 is 0
        return 0;

    return (num ^ (num >> 1)); // Compute Gray code directly
}

// Recursive function to print binary representation
void decimalToBinary(int decimalNumber) {
    if (decimalNumber > 1) {
        decimalToBinary(decimalNumber / 2); // Recursively divide the number by 2
    }
    printf("%d", decimalNumber % 2); // Print the remainder (binary digit)
}
```

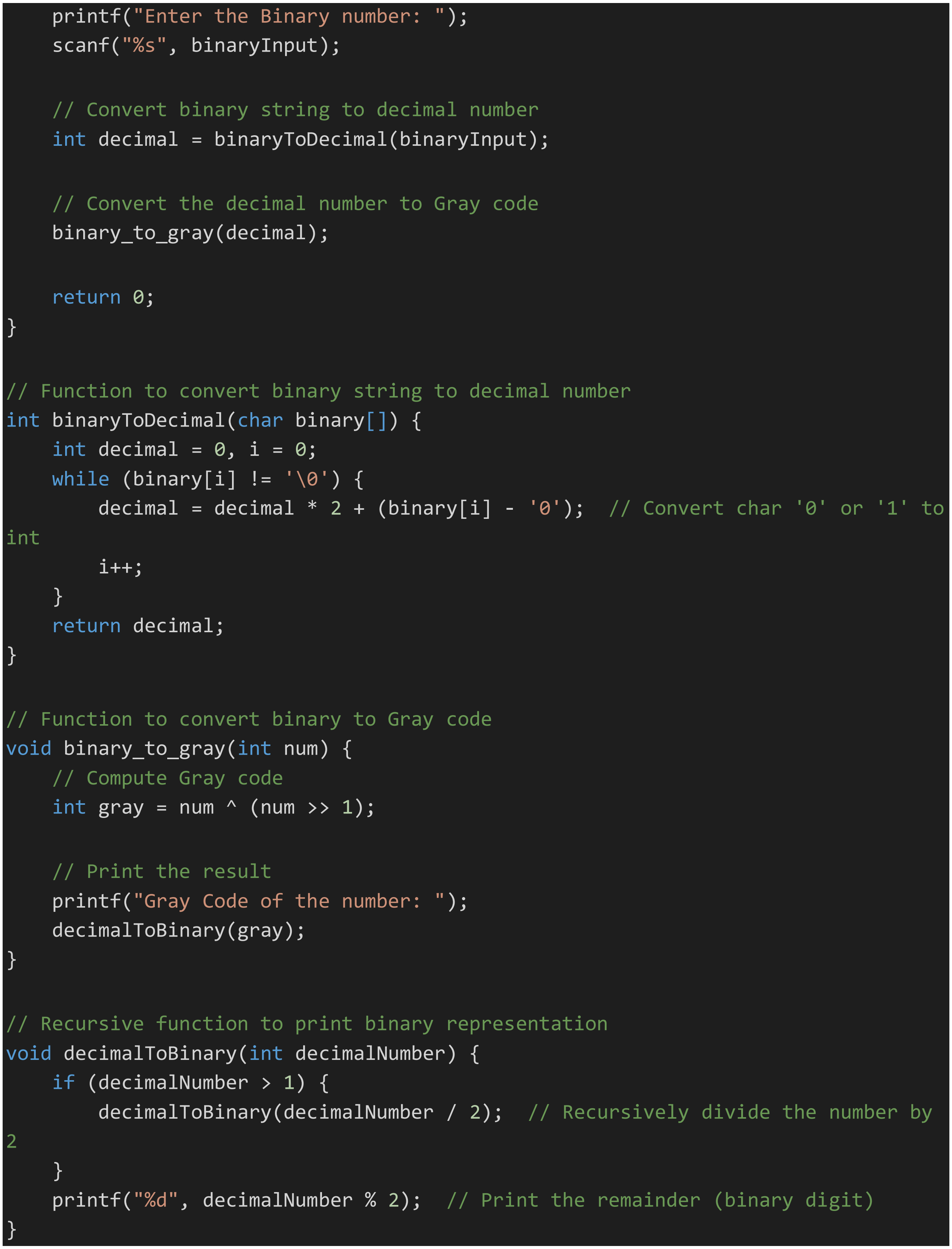


```
PS C:\Users\betti\Desktop\Training\Day17> ./task12
Enter the Binary number: 10110
Gray Code of the number: 11101
```

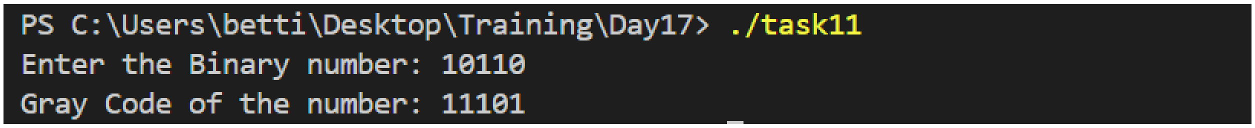
11.



```c
/*C program to convert binary number to gray code*/

#include <stdio.h>

int binaryToDecimal(char binary[]);
void binary_to_gray(int);
void decimalToBinary(int decimalNumber);

int main() {
    char binaryInput[32];
```

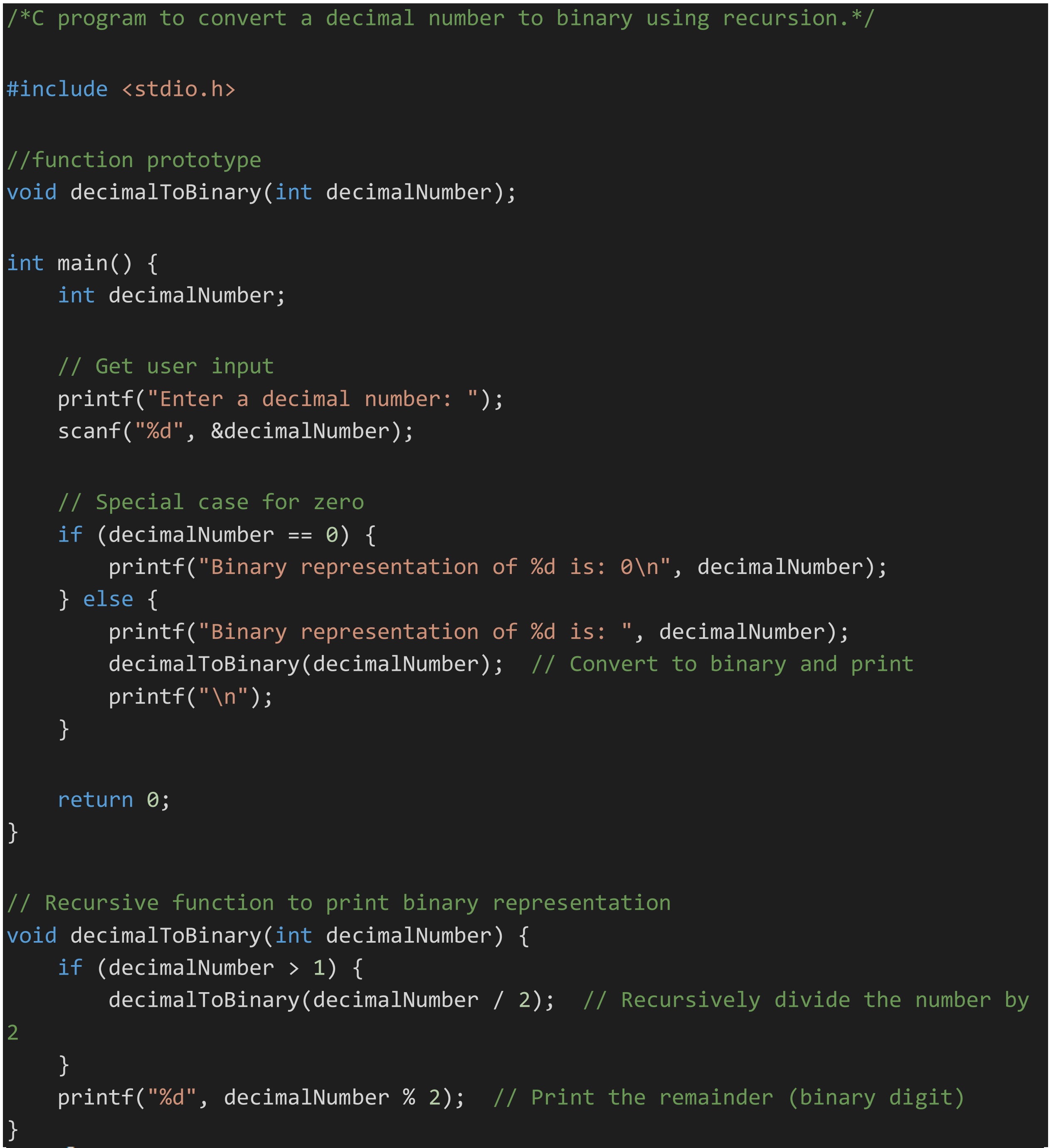


```
    printf("Enter the Binary number: ");
    scanf("%s", binaryInput);

    // Convert binary string to decimal number
    int decimal = binaryToDecimal(binaryInput);

    // Convert the decimal number to Gray code
    binary_to_gray(decimal);

    return 0;
}

// Function to convert binary string to decimal number
int binaryToDecimal(char binary[]) {
    int decimal = 0, i = 0;
    while (binary[i] != '\0') {
        decimal = decimal * 2 + (binary[i] - '0');  // Convert char '0' or '1' to
int
        i++;
    }
    return decimal;
}

// Function to convert binary to Gray code
void binary_to_gray(int num) {
    // Compute Gray code
    int gray = num ^ (num >> 1);

    // Print the result
    printf("Gray Code of the number: ");
    decimalToBinary(gray);
}

// Recursive function to print binary representation
void decimalToBinary(int decimalNumber) {
    if (decimalNumber > 1) {
        decimalToBinary(decimalNumber / 2);  // Recursively divide the number by
2
    }
    printf("%d", decimalNumber % 2);  // Print the remainder (binary digit)
}
```

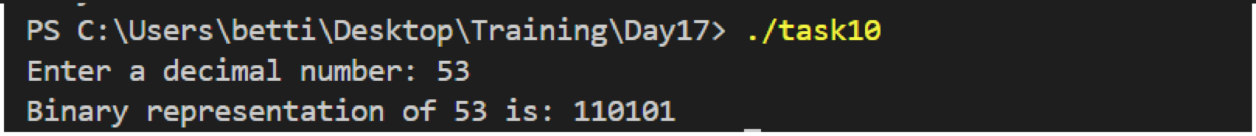


```
PS C:\Users\betti\Desktop\Training\Day17> ./task11
Enter the Binary number: 10110
Gray Code of the number: 11101
```

10.

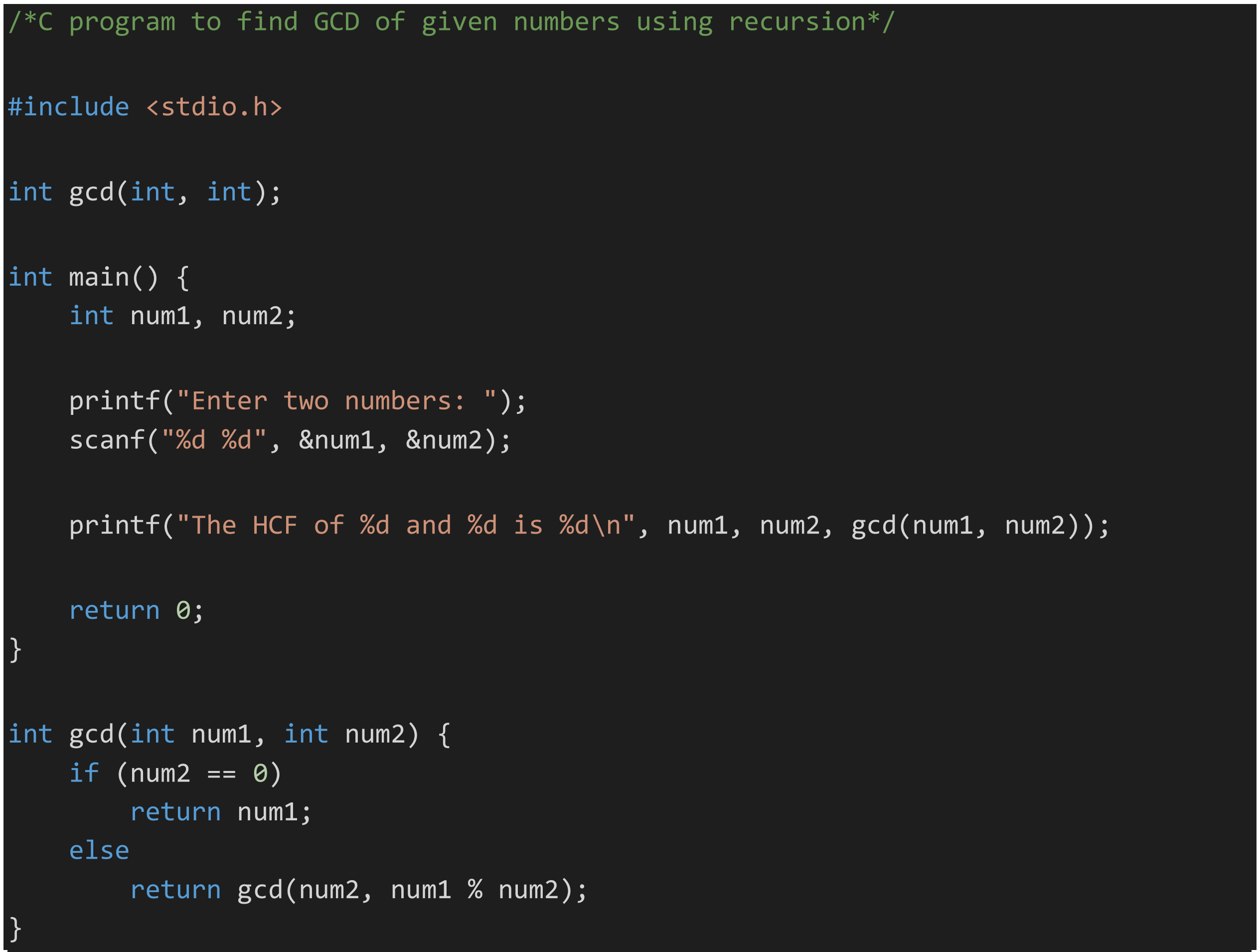


```c
/*C program to convert a decimal number to binary using recursion.*/

#include <stdio.h>

//function prototype
void decimalToBinary(int decimalNumber);

int main() {
    int decimalNumber;

    // Get user input
    printf("Enter a decimal number: ");
    scanf("%d", &decimalNumber);

    // Special case for zero
    if (decimalNumber == 0) {
        printf("Binary representation of %d is: 0\n", decimalNumber);
    } else {
        printf("Binary representation of %d is: ", decimalNumber);
        decimalToBinary(decimalNumber);  // Convert to binary and print
        printf("\n");
    }

    return 0;
}

// Recursive function to print binary representation
void decimalToBinary(int decimalNumber) {
    if (decimalNumber > 1) {
        decimalToBinary(decimalNumber / 2);  // Recursively divide the number by
2
    }
    printf("%d", decimalNumber % 2);  // Print the remainder (binary digit)
}
```

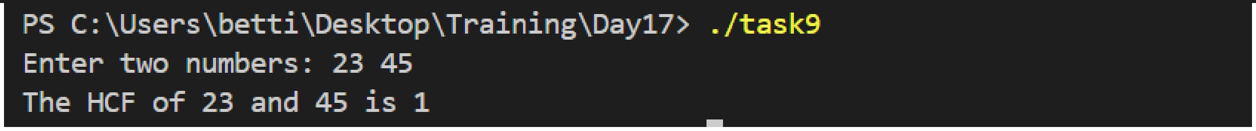


```
PS C:\Users\betti\Desktop\Training\Day17> ./task10
Enter a decimal number: 53
Binary representation of 53 is: 110101
```

9.

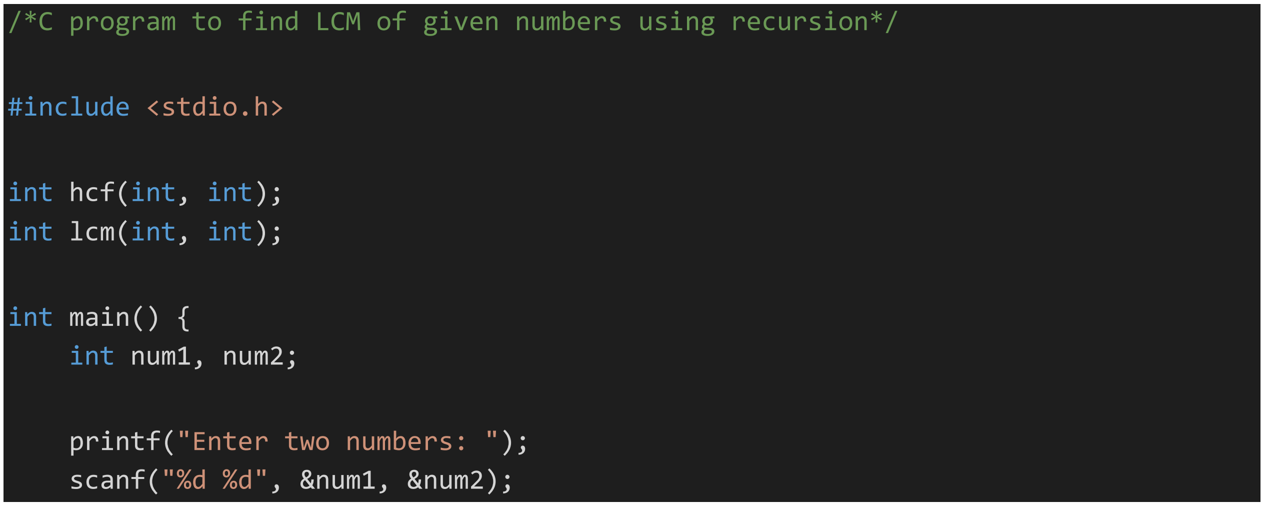


```c
/*C program to find GCD of given numbers using recursion*/

#include <stdio.h>

int gcd(int, int);

int main() {
    int num1, num2;

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

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

    return 0;
}

int gcd(int num1, int num2) {
    if (num2 == 0)
        return num1;
    else
        return gcd(num2, num1 % num2);
}
```

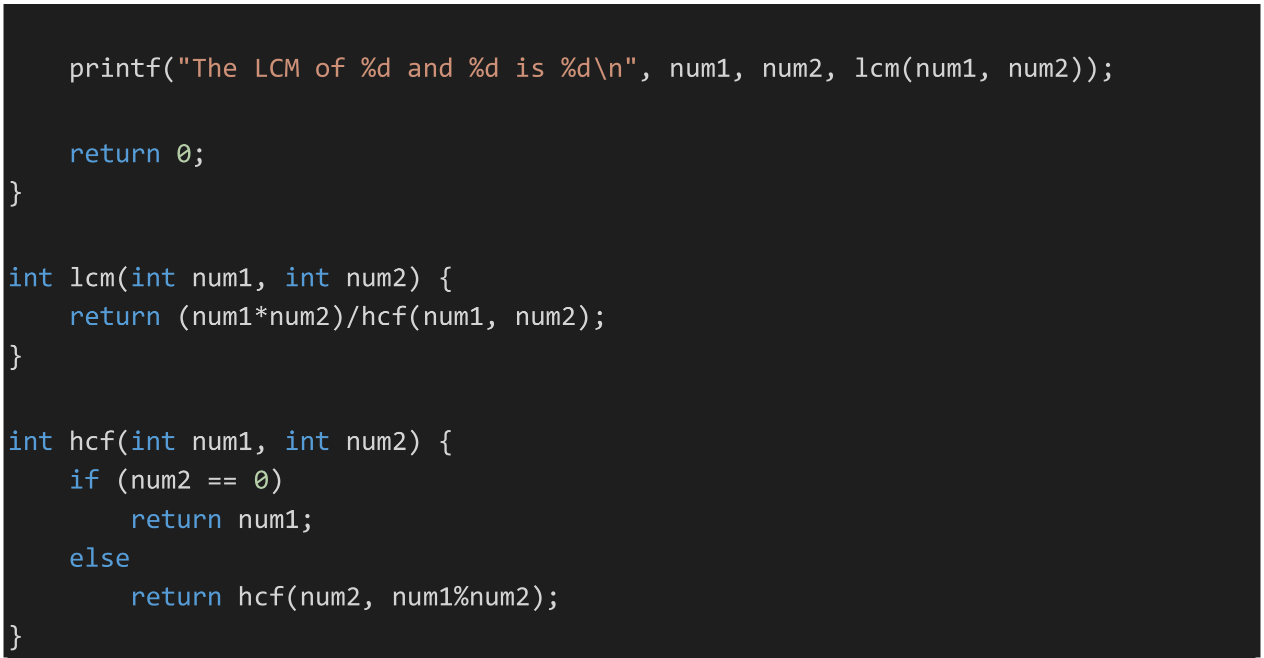


```
PS C:\Users\betti\Desktop\Training\Day17> ./task9
Enter two numbers: 23 45
The HCF of 23 and 45 is 1
```

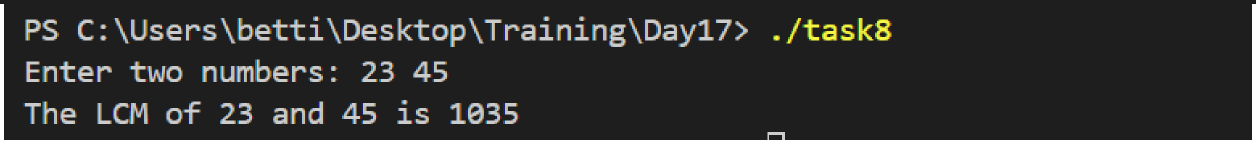
8.



```c
/*C program to find LCM of given numbers using recursion*/

#include <stdio.h>

int hcf(int, int);
int lcm(int, int);

int main() {
    int num1, num2;

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

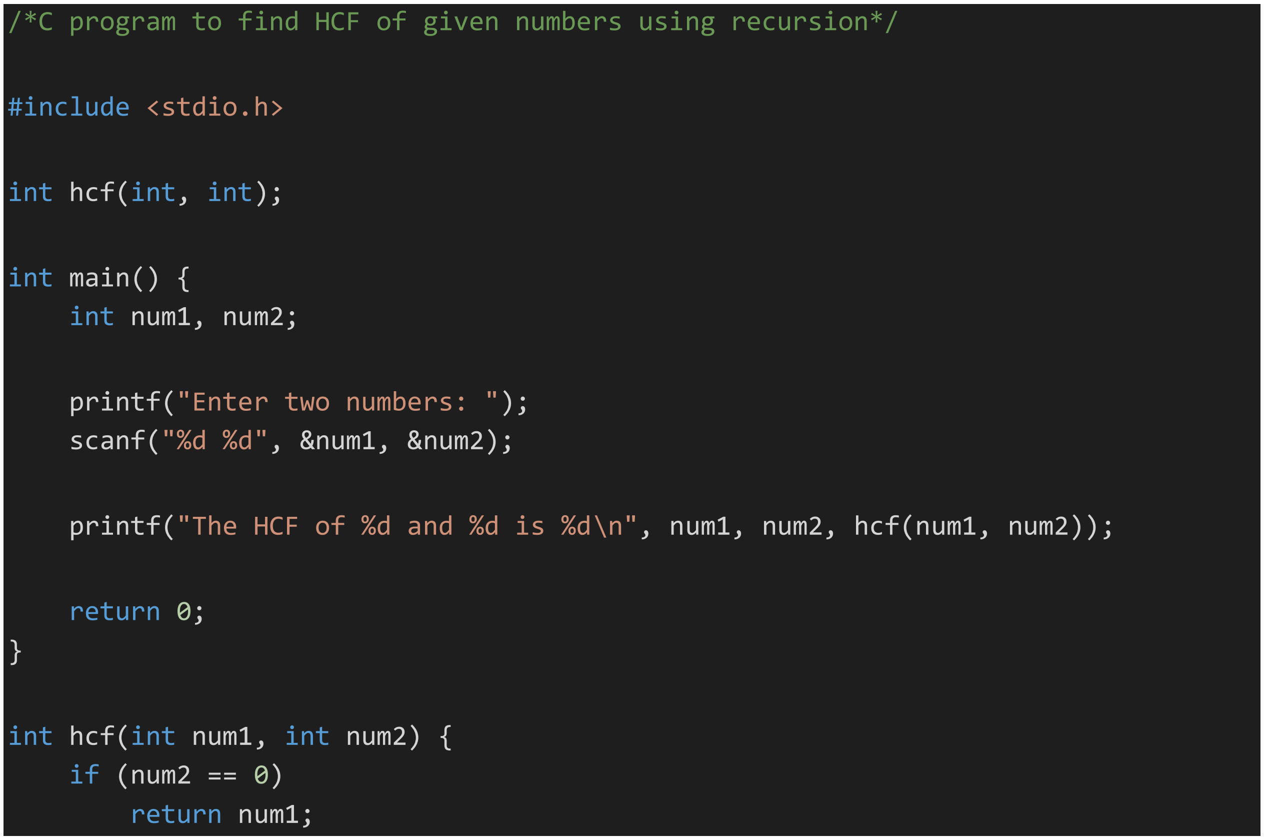


```c
    printf("The LCM of %d and %d is %d\n", num1, num2, lcm(num1, num2));

    return 0;
}

int lcm(int num1, int num2) {
    return (num1*num2)/hcf(num1, num2);
}

int hcf(int num1, int num2) {
    if (num2 == 0)
        return num1;
    else
        return hcf(num2, num1%num2);
}
```

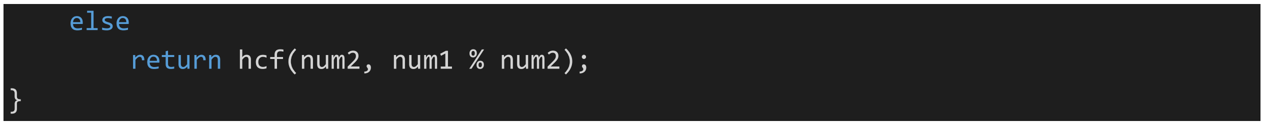


```
PS C:\Users\betti\Desktop\Training\Day17> ./task8
Enter two numbers: 23 45
The LCM of 23 and 45 is 1035
```

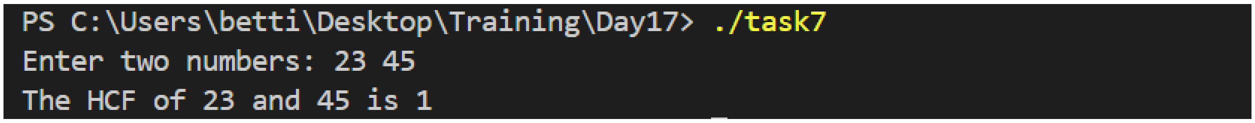
7.



```c
/*C program to find HCF of given numbers using recursion*/

#include <stdio.h>

int hcf(int, int);

int main() {
    int num1, num2;

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

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

    return 0;
}

int hcf(int num1, int num2) {
    if (num2 == 0)
        return num1;
```

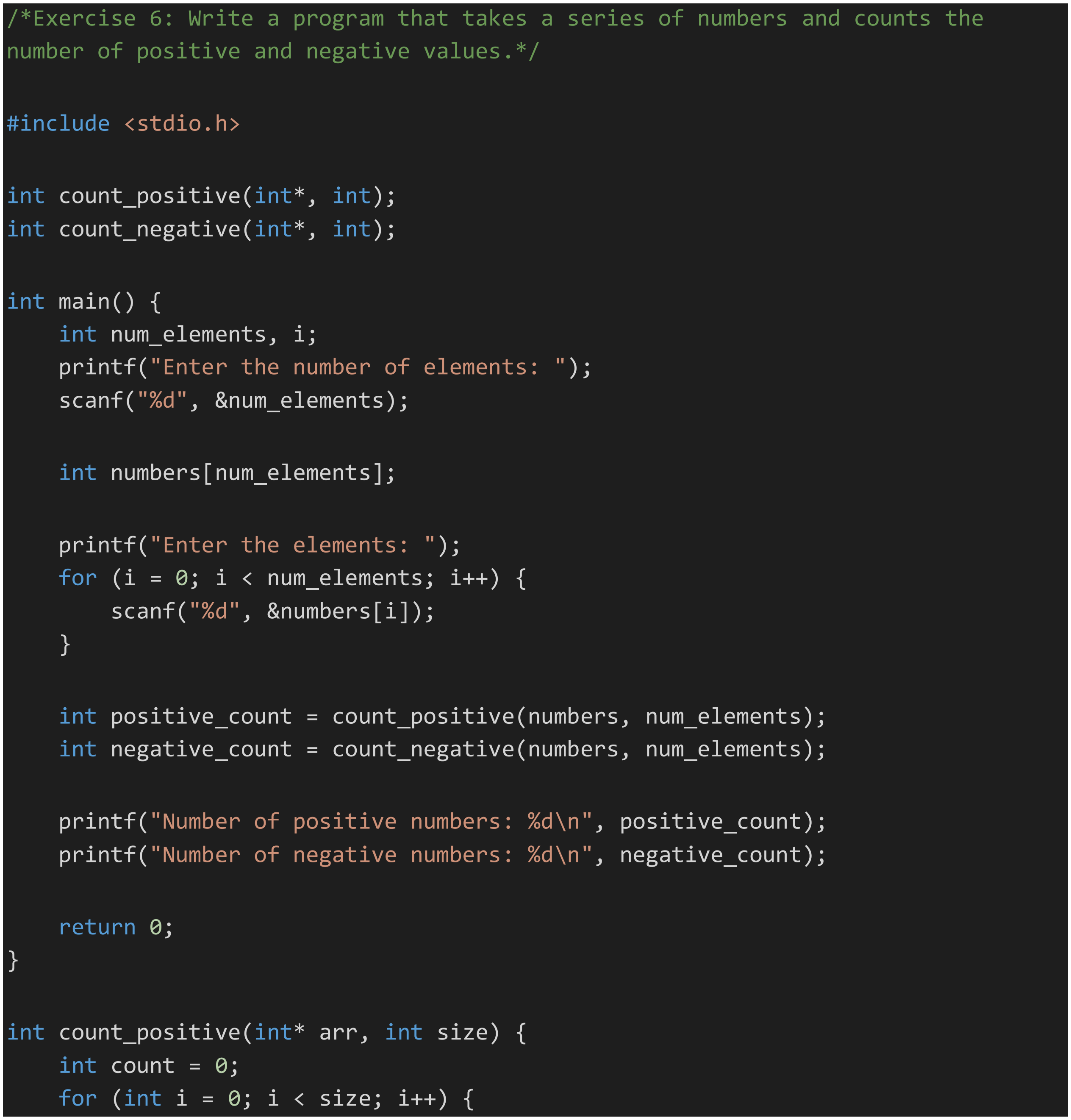


```
    else
        return hcf(num2, num1 % num2);
}
```

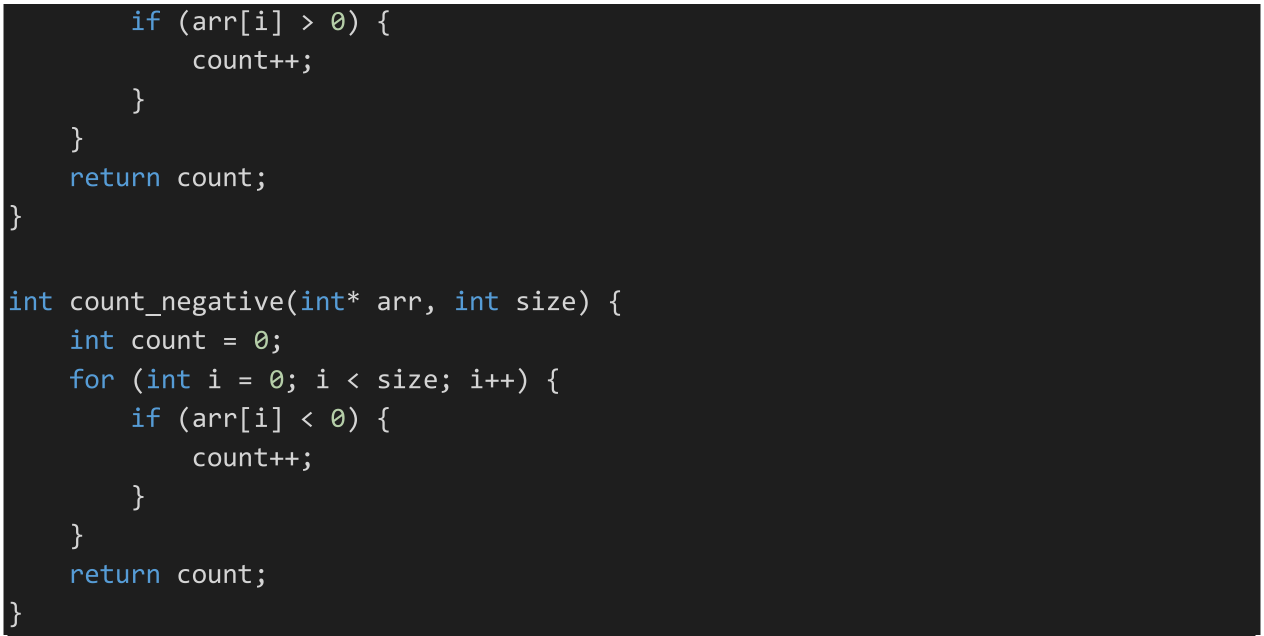


```
PS C:\Users\betti\Desktop\Training\Day17> ./task7
Enter two numbers: 23 45
The HCF of 23 and 45 is 1
```

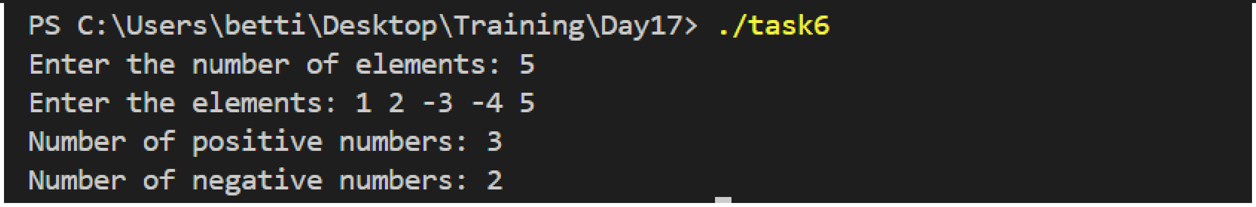
6.



```
/*Exercise 6: Write a program that takes a series of numbers and counts the
number of positive and negative values.*/

#include <stdio.h>

int count_positive(int*, int);
int count_negative(int*, int);

int main() {
    int num_elements, i;
    printf("Enter the number of elements: ");
    scanf("%d", &num_elements);

    int numbers[num_elements];

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

    int positive_count = count_positive(numbers, num_elements);
    int negative_count = count_negative(numbers, num_elements);

    printf("Number of positive numbers: %d\n", positive_count);
    printf("Number of negative numbers: %d\n", negative_count);

    return 0;
}

int count_positive(int* arr, int size) {
    int count = 0;
    for (int i = 0; i < size; i++) {
```

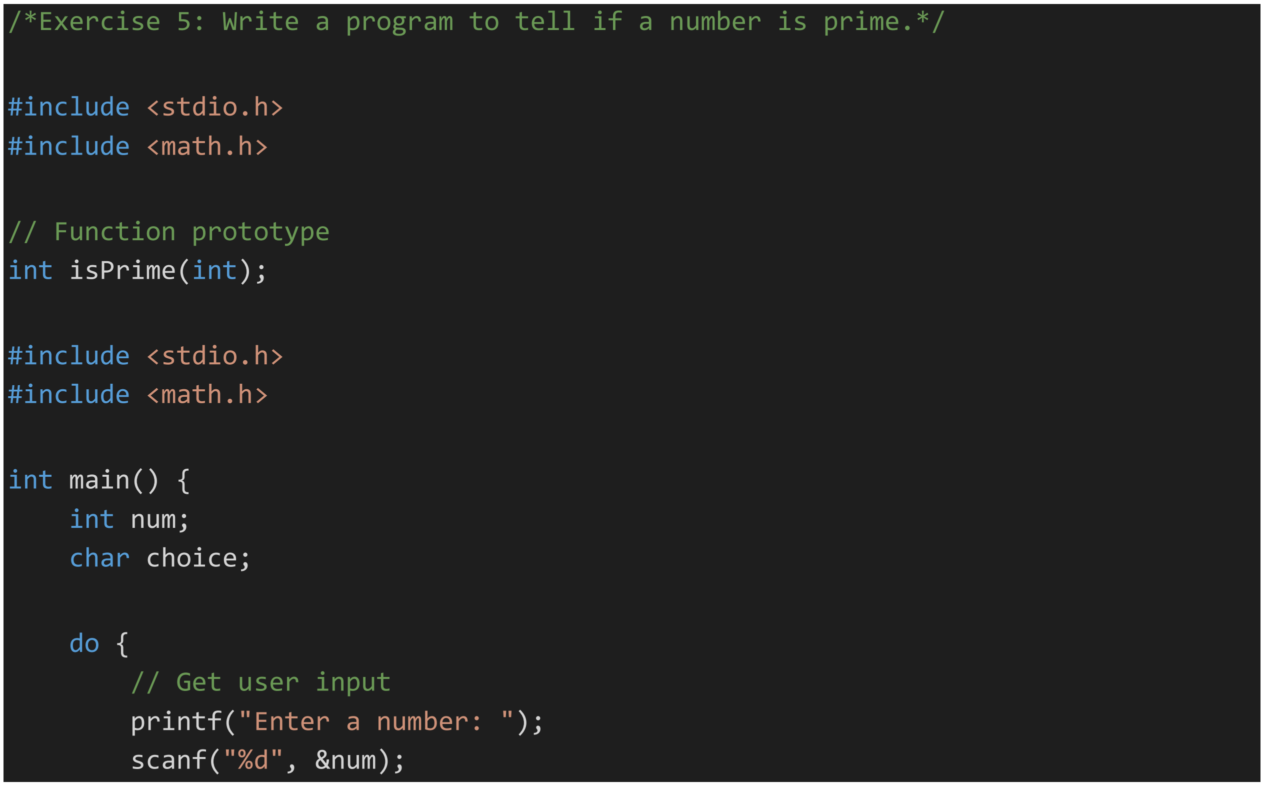


```c
        if (arr[i] > 0) {
            count++;
        }
    }
    return count;
}

int count_negative(int* arr, int size) {
    int count = 0;
    for (int i = 0; i < size; i++) {
        if (arr[i] < 0) {
            count++;
        }
    }
    return count;
}
```

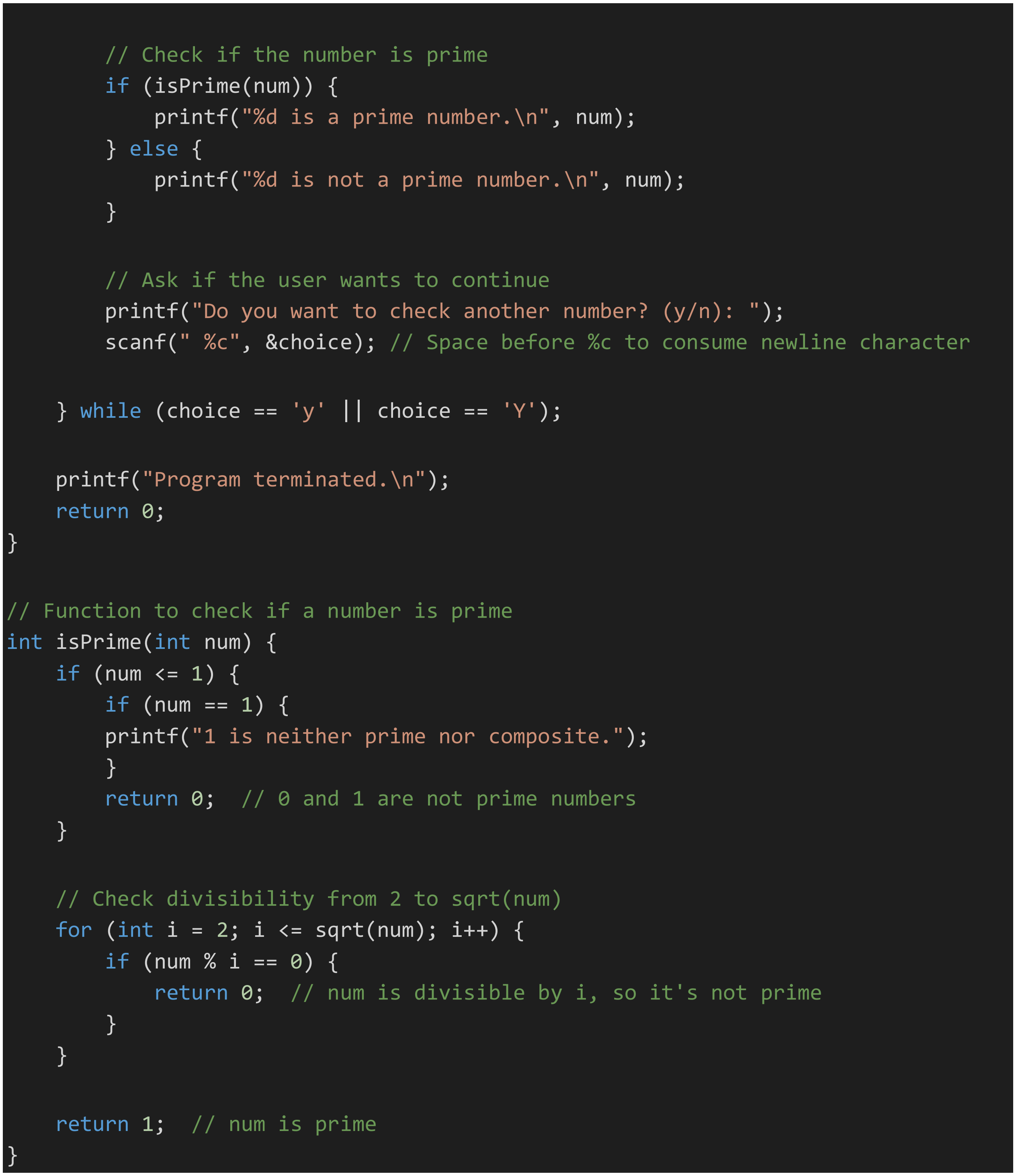


```
PS C:\Users\betti\Desktop\Training\Day17> ./task6
Enter the number of elements: 5
Enter the elements: 1 2 -3 -4 5
Number of positive numbers: 3
Number of negative numbers: 2
```

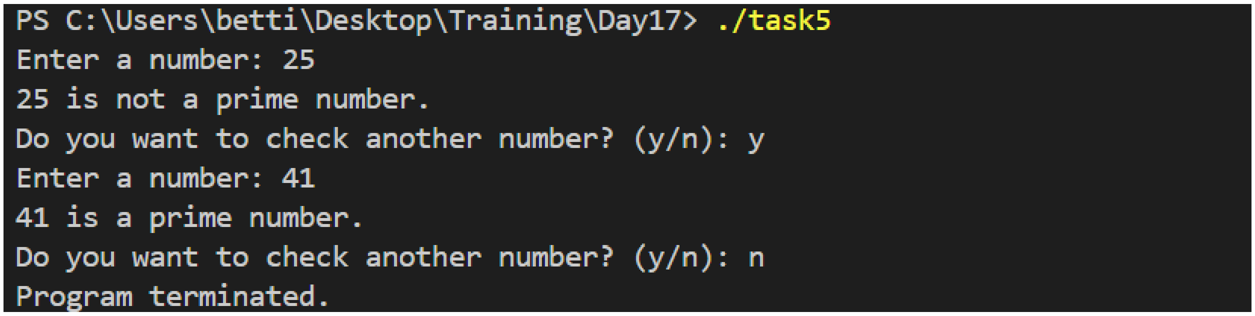
5.



```c
/*Exercise 5: Write a program to tell if a number is prime.*/

#include <stdio.h>
#include <math.h>

// Function prototype
int isPrime(int);

#include <stdio.h>
#include <math.h>

int main() {
    int num;
    char choice;

    do {
        // Get user input
        printf("Enter a number: ");
        scanf("%d", &num);
```

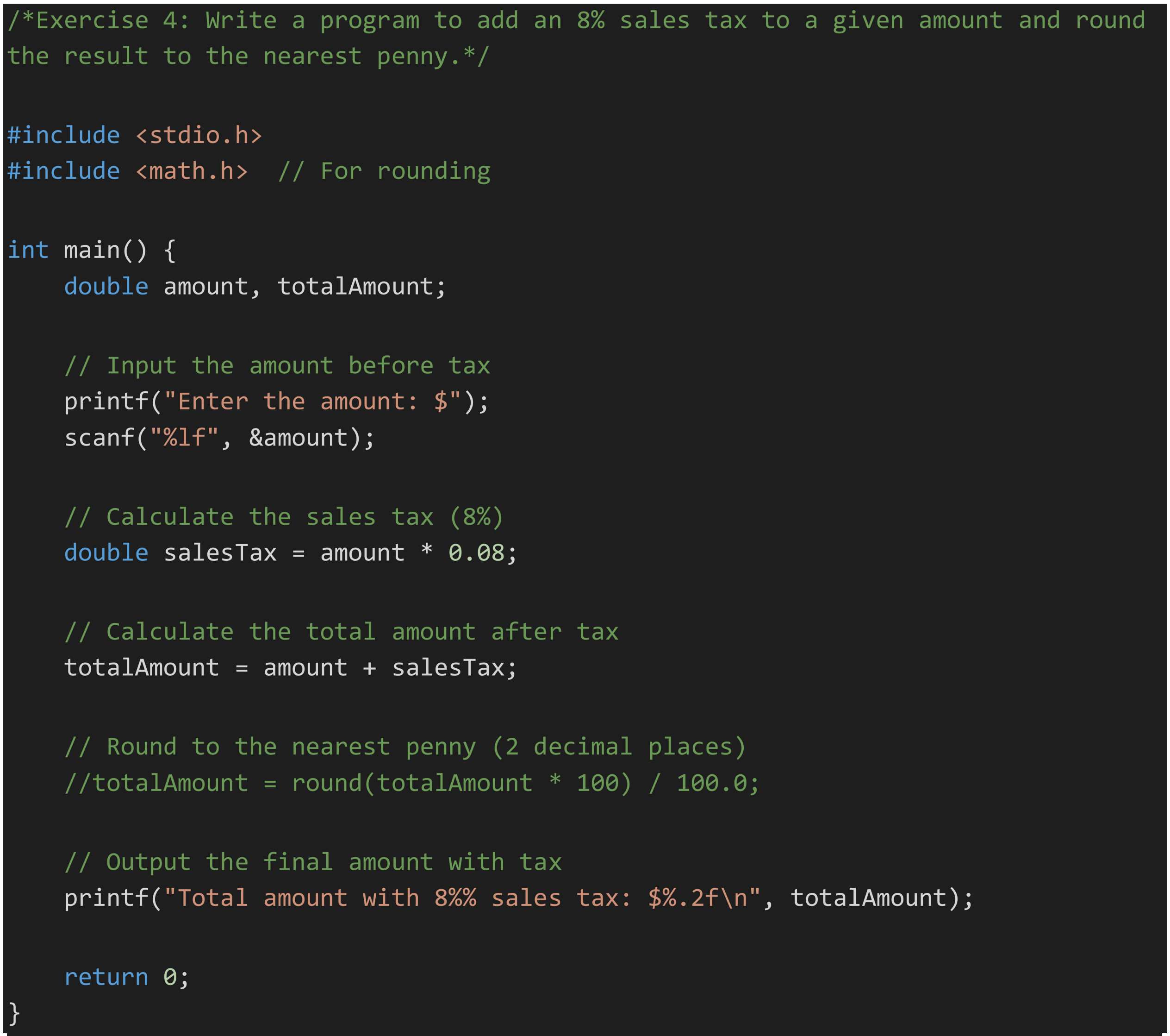


```c
        // Check if the number is prime
        if (isPrime(num)) {
            printf("%d is a prime number.\n", num);
        } else {
            printf("%d is not a prime number.\n", num);
        }

        // Ask if the user wants to continue
        printf("Do you want to check another number? (y/n): ");
        scanf(" %c", &choice); // Space before %c to consume newline character

    } while (choice == 'y' || choice == 'Y');

    printf("Program terminated.\n");
    return 0;
}

// Function to check if a number is prime
int isPrime(int num) {
    if (num <= 1) {
        if (num == 1) {
        printf("1 is neither prime nor composite.");
        }
        return 0;  // 0 and 1 are not prime numbers
    }

    // Check divisibility from 2 to sqrt(num)
    for (int i = 2; i <= sqrt(num); i++) {
        if (num % i == 0) {
            return 0;  // num is divisible by i, so it's not prime
        }
    }

    return 1;  // num is prime
}
```

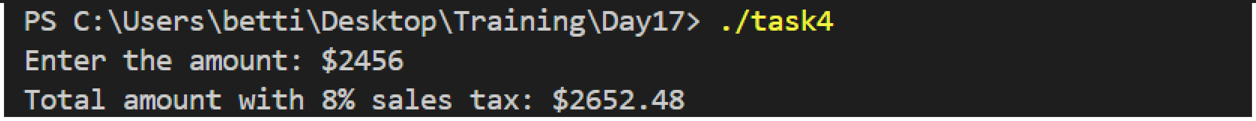


```
PS C:\Users\betti\Desktop\Training\Day17> ./task5
Enter a number: 25
25 is not a prime number.
Do you want to check another number? (y/n): y
Enter a number: 41
41 is a prime number.
Do you want to check another number? (y/n): n
Program terminated.
```

4.

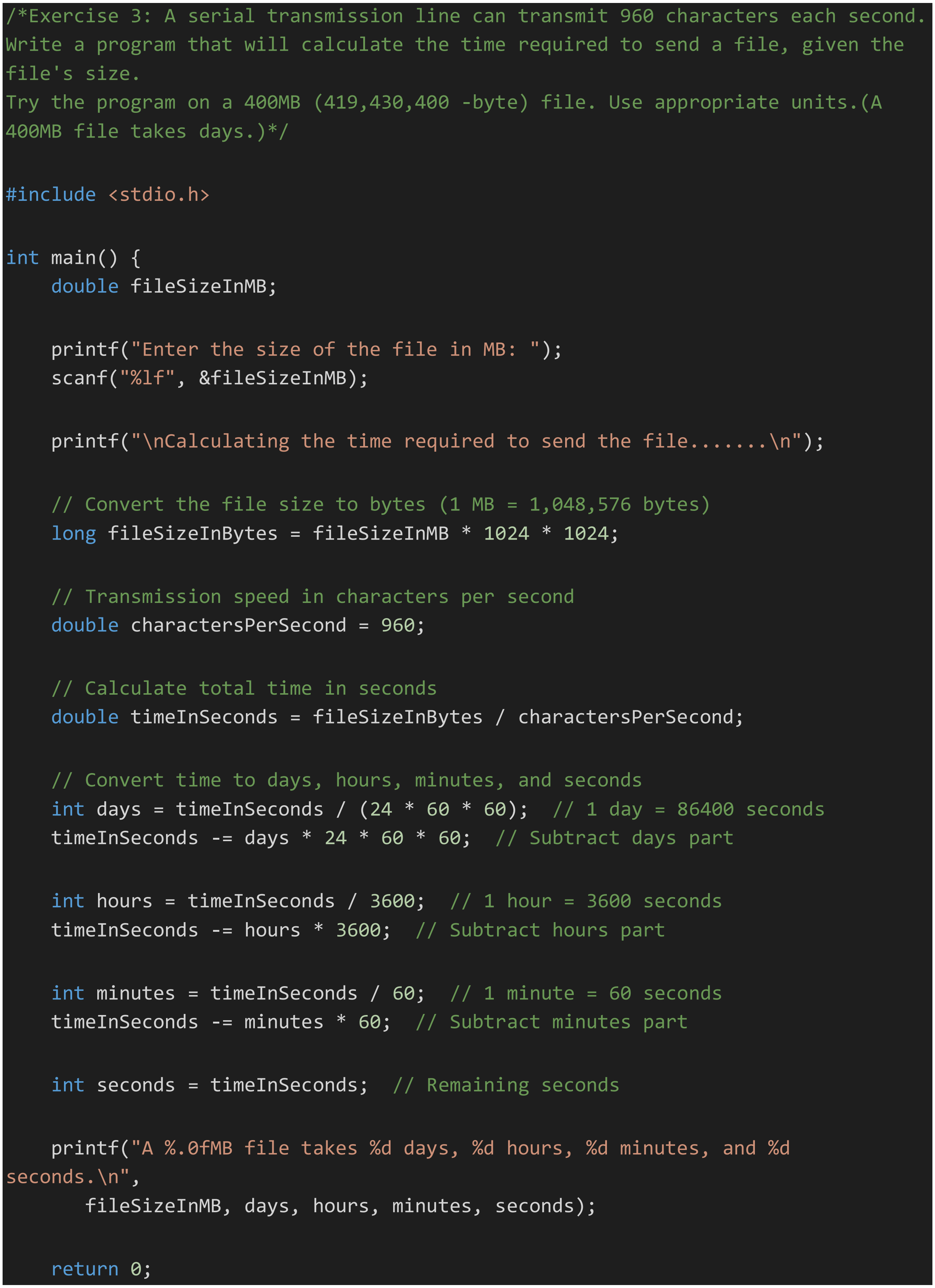


```c
/*Exercise 4: Write a program to add an 8% sales tax to a given amount and round
the result to the nearest penny.*/

#include <stdio.h>
#include <math.h>  // For rounding

int main() {
    double amount, totalAmount;

    // Input the amount before tax
    printf("Enter the amount: $");
    scanf("%lf", &amount);

    // Calculate the sales tax (8%)
    double salesTax = amount * 0.08;

    // Calculate the total amount after tax
    totalAmount = amount + salesTax;

    // Round to the nearest penny (2 decimal places)
    //totalAmount = round(totalAmount * 100) / 100.0;

    // Output the final amount with tax
    printf("Total amount with 8%% sales tax: $%.2f\n", totalAmount);

    return 0;
}
```

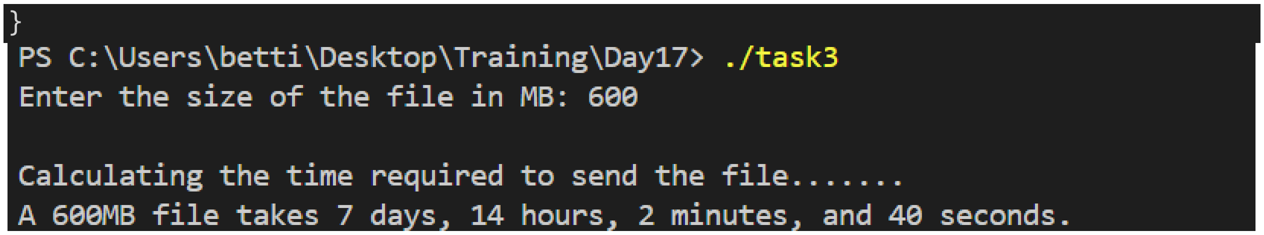


```
PS C:\Users\betti\Desktop\Training\Day17> ./task4
Enter the amount: $2456
Total amount with 8% sales tax: $2652.48
```

3.



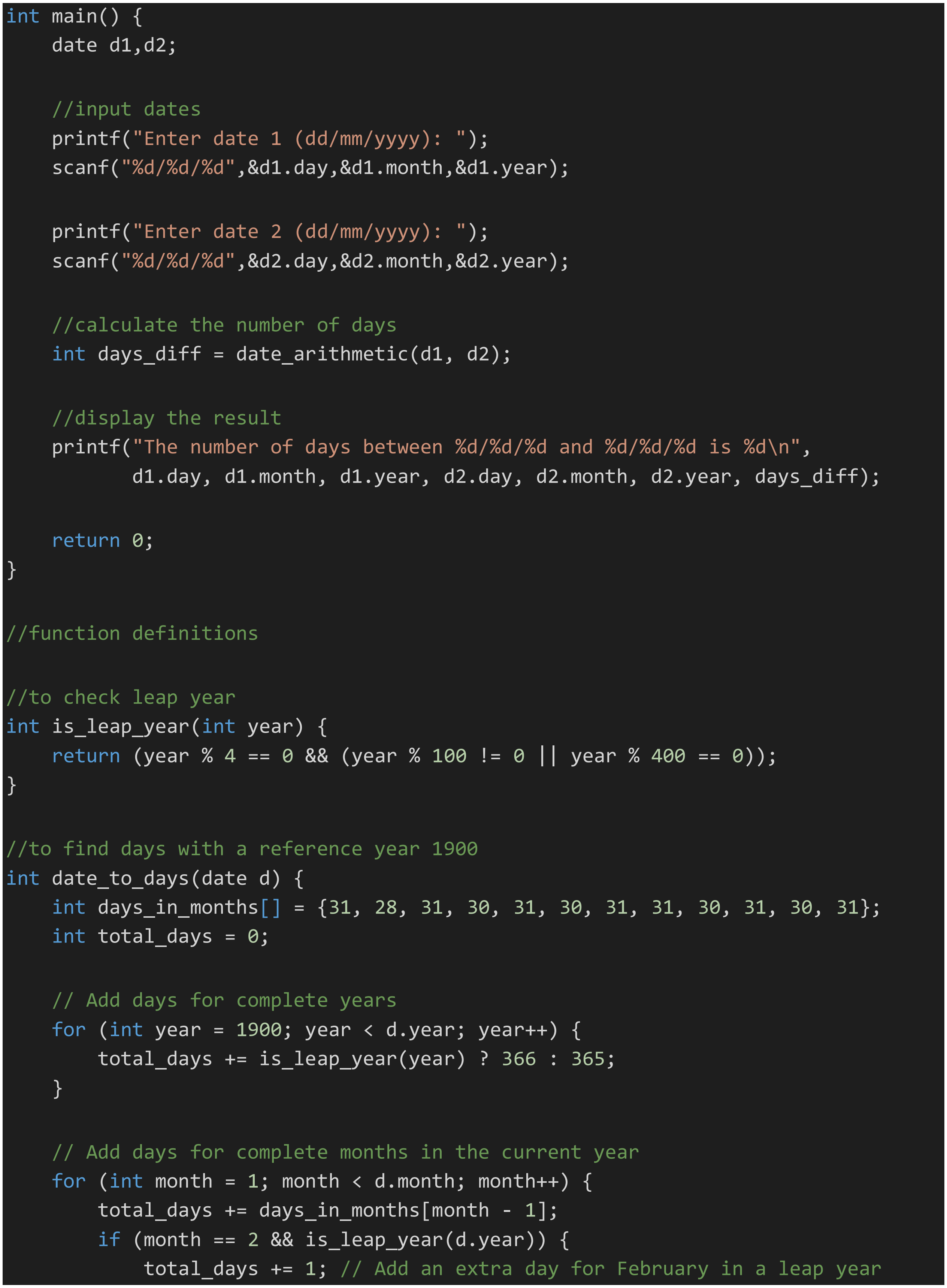
```
/*Exercise 3: A serial transmission line can transmit 960 characters each second.
Write a program that will calculate the time required to send a file, given the
file's size.
Try the program on a 400MB (419,430,400 -byte) file. Use appropriate units.(A
400MB file takes days.)*/

#include <stdio.h>

int main() {
    double fileSizeInMB;

    printf("Enter the size of the file in MB: ");
    scanf("%lf", &fileSizeInMB);

    printf("\nCalculating the time required to send the file.......\n");

    // Convert the file size to bytes (1 MB = 1,048,576 bytes)
    long fileSizeInBytes = fileSizeInMB * 1024 * 1024;

    // Transmission speed in characters per second
    double charactersPerSecond = 960;

    // Calculate total time in seconds
    double timeInSeconds = fileSizeInBytes / charactersPerSecond;

    // Convert time to days, hours, minutes, and seconds
    int days = timeInSeconds / (24 * 60 * 60);  // 1 day = 86400 seconds
    timeInSeconds -= days * 24 * 60 * 60;  // Subtract days part

    int hours = timeInSeconds / 3600;  // 1 hour = 3600 seconds
    timeInSeconds -= hours * 3600;  // Subtract hours part

    int minutes = timeInSeconds / 60;  // 1 minute = 60 seconds
    timeInSeconds -= minutes * 60;  // Subtract minutes part

    int seconds = timeInSeconds;  // Remaining seconds

    printf("A %.0fMB file takes %d days, %d hours, %d minutes, and %d
seconds.\n",
       fileSizeInMB, days, hours, minutes, seconds);

    return 0;
}
```



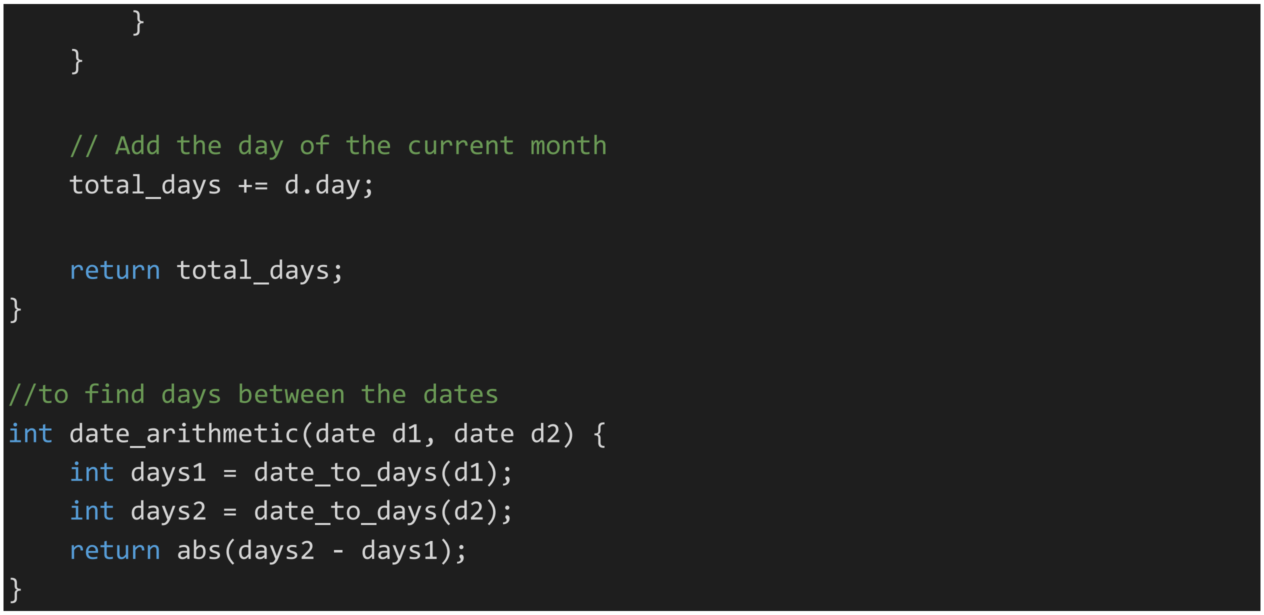
```
PS C:\Users\betti\Desktop\Training\Day17> ./task3
Enter the size of the file in MB: 600

Calculating the time required to send the file.......
A 600MB file takes 7 days, 14 hours, 2 minutes, and 40 seconds.
```

2.

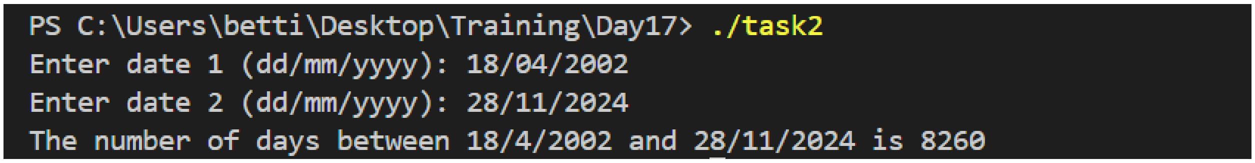
```c
/*Exercise 2: Write a program to perform date arithmetic such as how many days
there are between 6/6/90 and 4/3/92. Include a specification and a code design.*/

/*Program Specification:
Purpose:
The program performs the date arithmetic to find the number of days between two
given dates

Inputs:
2 dates to fiind days between them

Function:
1. Takes two dates as input.
2. Returns the number of days between them

Outputs:
Days between the given dates*/

#include <stdio.h>
#include <stdlib.h>

//structure to hold the date
typedef struct date {
    int day;
    int month;
    int year;
}date;

//function prototypes
int date_arithmetic(date, date);
int is_leap_year(int);
int days_to_date(date);
```



```c
int main() {
    date d1,d2;

    //input dates
    printf("Enter date 1 (dd/mm/yyyy): ");
    scanf("%d/%d/%d",&d1.day,&d1.month,&d1.year);

    printf("Enter date 2 (dd/mm/yyyy): ");
    scanf("%d/%d/%d",&d2.day,&d2.month,&d2.year);

    //calculate the number of days
    int days_diff = date_arithmetic(d1, d2);

    //display the result
    printf("The number of days between %d/%d/%d and %d/%d/%d is %d\n",
           d1.day, d1.month, d1.year, d2.day, d2.month, d2.year, days_diff);

    return 0;
}

//function definitions

//to check leap year
int is_leap_year(int year) {
    return (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0));
}

//to find days with a reference year 1900
int date_to_days(date d) {
    int days_in_months[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    int total_days = 0;

    // Add days for complete years
    for (int year = 1900; year < d.year; year++) {
        total_days += is_leap_year(year) ? 366 : 365;
    }

    // Add days for complete months in the current year
    for (int month = 1; month < d.month; month++) {
        total_days += days_in_months[month - 1];
        if (month == 2 && is_leap_year(d.year)) {
            total_days += 1; // Add an extra day for February in a leap year
```

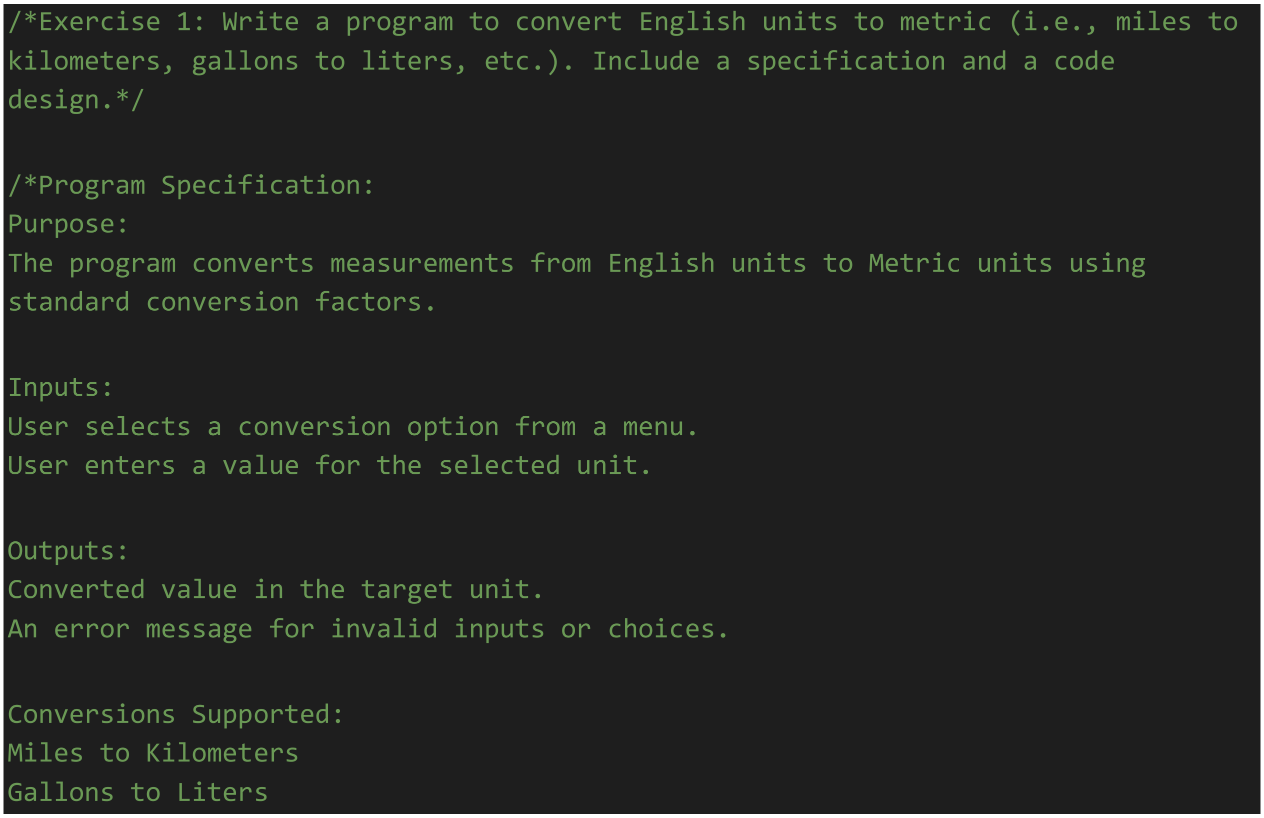


```
        }
    }

    // Add the day of the current month
    total_days += d.day;

    return total_days;
}

//to find days between the dates
int date_arithmetic(date d1, date d2) {
    int days1 = date_to_days(d1);
    int days2 = date_to_days(d2);
    return abs(days2 - days1);
}
```

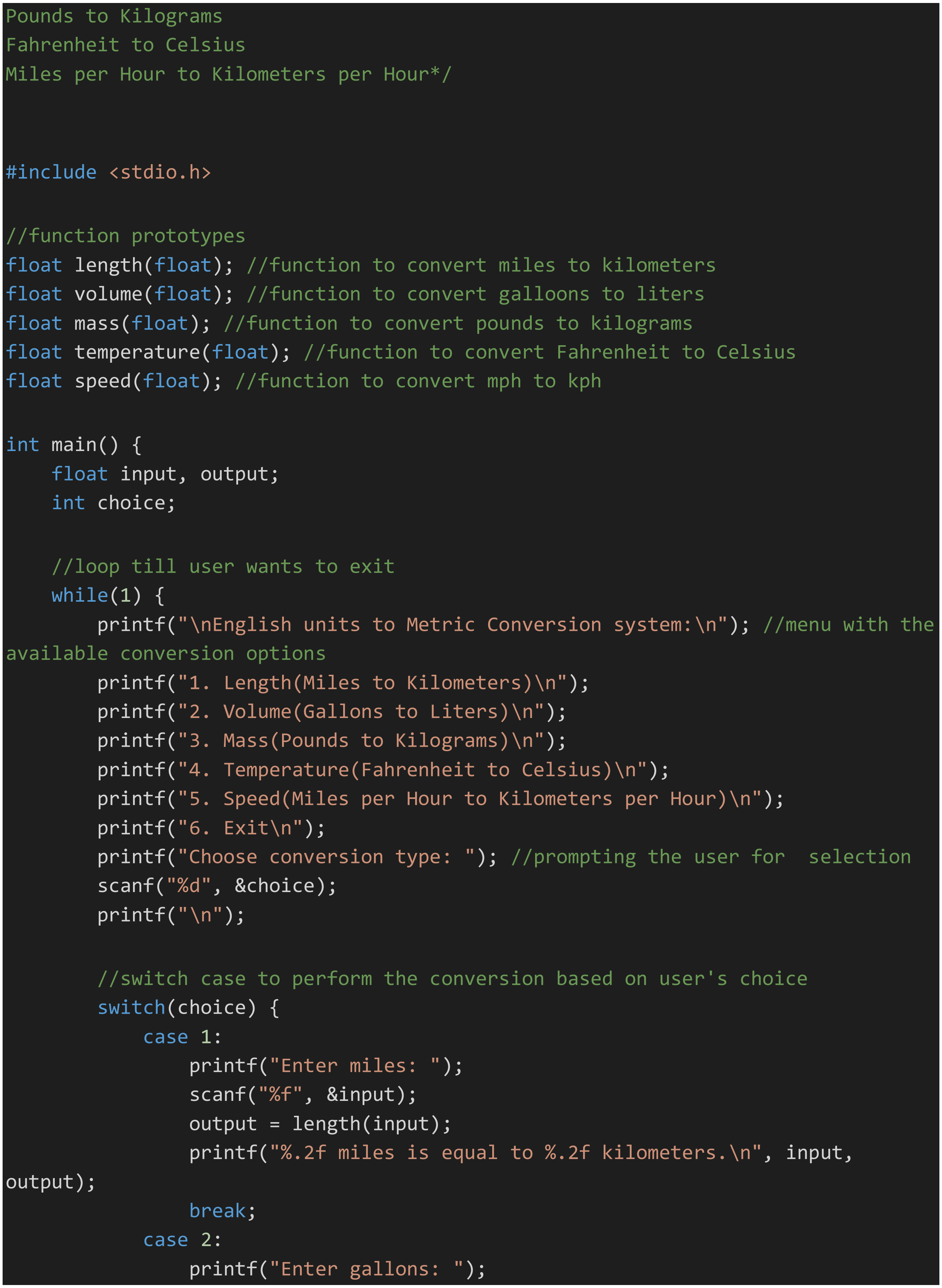


```
PS C:\Users\betti\Desktop\Training\Day17> ./task2
Enter date 1 (dd/mm/yyyy): 18/04/2002
Enter date 2 (dd/mm/yyyy): 28/11/2024
The number of days between 18/4/2002 and 28/11/2024 is 8260
```

1.



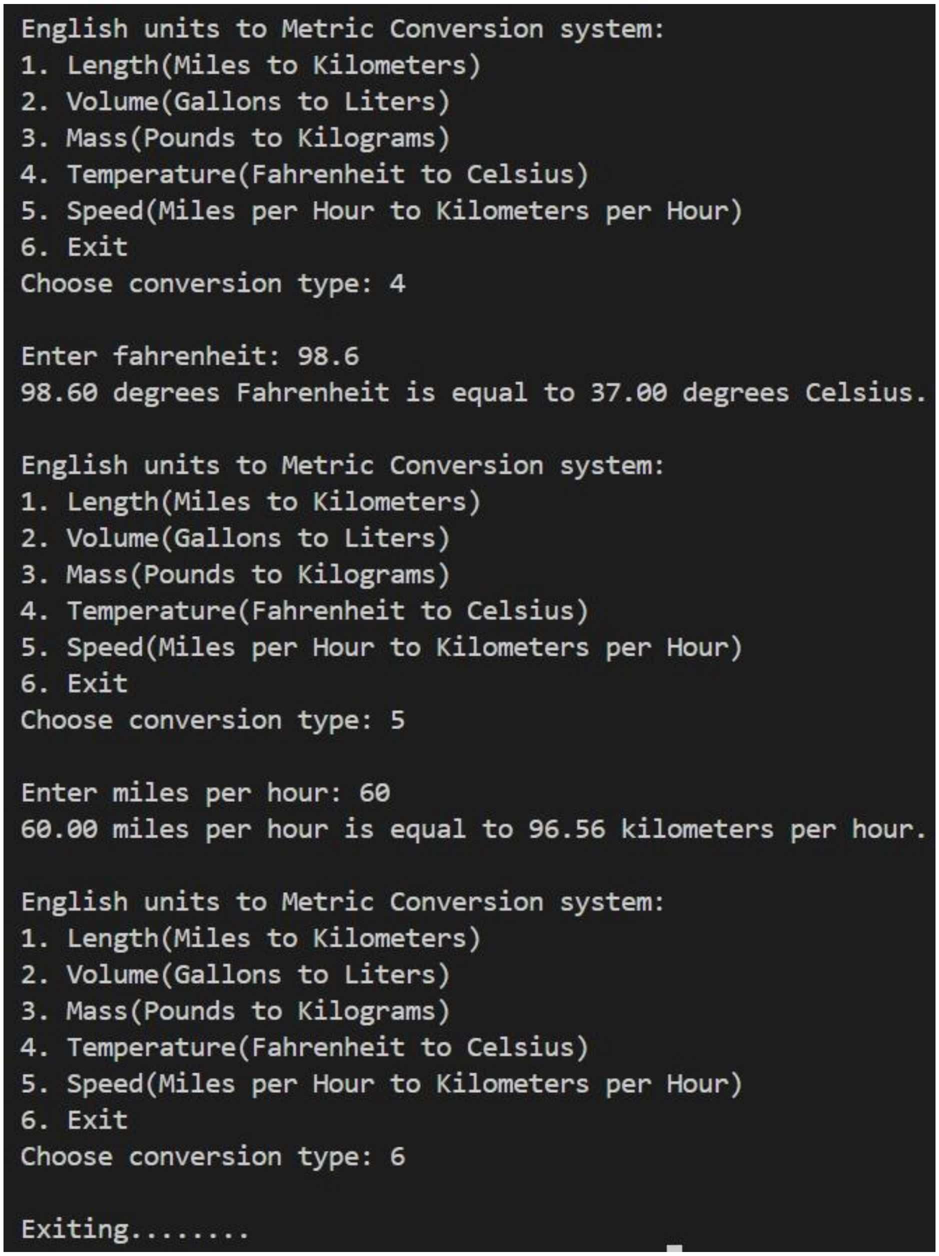
```
/*Exercise 1: Write a program to convert English units to metric (i.e., miles to
kilometers, gallons to liters, etc.). Include a specification and a code
design.*/

/*Program Specification:
Purpose:
The program converts measurements from English units to Metric units using
standard conversion factors.

Inputs:
User selects a conversion option from a menu.
User enters a value for the selected unit.

Outputs:
Converted value in the target unit.
An error message for invalid inputs or choices.

Conversions Supported:
Miles to Kilometers
Gallons to Liters
```



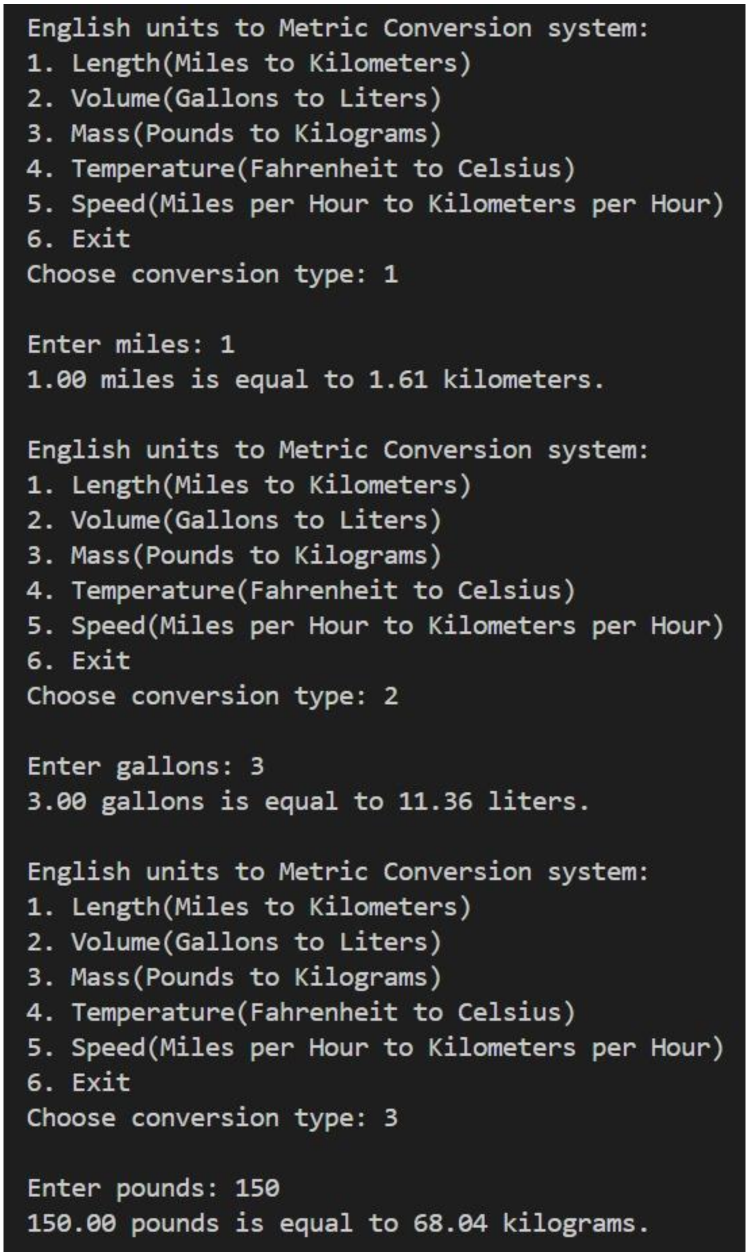
```
Pounds to Kilograms
Fahrenheit to Celsius
Miles per Hour to Kilometers per Hour*/


#include <stdio.h>

//function prototypes
float length(float); //function to convert miles to kilometers
float volume(float); //function to convert galloons to liters
float mass(float); //function to convert pounds to kilograms
float temperature(float); //function to convert Fahrenheit to Celsius
float speed(float); //function to convert mph to kph

int main() {
    float input, output;
    int choice;

    //loop till user wants to exit
    while(1) {
        printf("\nEnglish units to Metric Conversion system:\n"); //menu with the
available conversion options
        printf("1. Length(Miles to Kilometers)\n");
        printf("2. Volume(Gallons to Liters)\n");
        printf("3. Mass(Pounds to Kilograms)\n");
        printf("4. Temperature(Fahrenheit to Celsius)\n");
        printf("5. Speed(Miles per Hour to Kilometers per Hour)\n");
        printf("6. Exit\n");
        printf("Choose conversion type: "); //prompting the user for  selection
        scanf("%d", &choice);
        printf("\n");

        //switch case to perform the conversion based on user's choice
        switch(choice) {
            case 1:
                printf("Enter miles: ");
                scanf("%f", &input);
                output = length(input);
                printf("%.2f miles is equal to %.2f kilometers.\n", input,
output);
                break;
            case 2:
                printf("Enter gallons: ");
```

```c
                scanf("%f", &input);
                output = volume(input);
                printf("%.2f gallons is equal to %.2f liters.\n", input, output);
                break;
            case 3:
                printf("Enter pounds: ");
                scanf("%f", &input);
                output = mass(input);
                printf("%.2f pounds is equal to %.2f kilograms.\n", input,
output);
                break;
            case 4:
                printf("Enter fahrenheit: ");
                scanf("%f", &input);
                output = temperature(input);
                printf("%.2f degrees Fahrenheit is equal to %.2f degrees
Celsius.\n", input, output);
                break;
            case 5:
                printf("Enter miles per hour: ");
                scanf("%f", &input);
                output = speed(input);
                printf("%.2f miles per hour is equal to %.2f kilometers per
hour.\n", input, output);
                break;
            case 6:
                printf("Exiting........\n"); //exit code
                return 0;
            default:
                printf("Invalid choice. Please try again.\n"); //invalid option
check
                break;
        }
    }
    return 0;
}

//function definitions for each conversion
float length(float miles) {
    return miles * 1.60934;
}

float volume(float gallons) {
    return gallons * 3.78541;
```

```
}

float mass(float pounds) {
    return pounds * 0.453592;
}

float temperature(float fahrenheit) {
    return (fahrenheit - 32) * 5/9;
}

float speed(float milesPerHour) {
    return milesPerHour * 1.60934;
}
```



```
English units to Metric Conversion system:
1. Length(Miles to Kilometers)
2. Volume(Gallons to Liters)
3. Mass(Pounds to Kilograms)
4. Temperature(Fahrenheit to Celsius)
5. Speed(Miles per Hour to Kilometers per Hour)
6. Exit
Choose conversion type: 4

Enter fahrenheit: 98.6
98.60 degrees Fahrenheit is equal to 37.00 degrees Celsius.

English units to Metric Conversion system:
1. Length(Miles to Kilometers)
2. Volume(Gallons to Liters)
3. Mass(Pounds to Kilograms)
4. Temperature(Fahrenheit to Celsius)
5. Speed(Miles per Hour to Kilometers per Hour)
6. Exit
Choose conversion type: 5

Enter miles per hour: 60
60.00 miles per hour is equal to 96.56 kilometers per hour.

English units to Metric Conversion system:
1. Length(Miles to Kilometers)
2. Volume(Gallons to Liters)
3. Mass(Pounds to Kilograms)
4. Temperature(Fahrenheit to Celsius)
5. Speed(Miles per Hour to Kilometers per Hour)
6. Exit
Choose conversion type: 6

Exiting........
```



```
English units to Metric Conversion system:
1. Length(Miles to Kilometers)
2. Volume(Gallons to Liters)
3. Mass(Pounds to Kilograms)
4. Temperature(Fahrenheit to Celsius)
5. Speed(Miles per Hour to Kilometers per Hour)
6. Exit
Choose conversion type: 1

Enter miles: 1
1.00 miles is equal to 1.61 kilometers.

English units to Metric Conversion system:
1. Length(Miles to Kilometers)
2. Volume(Gallons to Liters)
3. Mass(Pounds to Kilograms)
4. Temperature(Fahrenheit to Celsius)
5. Speed(Miles per Hour to Kilometers per Hour)
6. Exit
Choose conversion type: 2

Enter gallons: 3
3.00 gallons is equal to 11.36 liters.

English units to Metric Conversion system:
1. Length(Miles to Kilometers)
2. Volume(Gallons to Liters)
3. Mass(Pounds to Kilograms)
4. Temperature(Fahrenheit to Celsius)
5. Speed(Miles per Hour to Kilometers per Hour)
6. Exit
Choose conversion type: 3

Enter pounds: 150
150.00 pounds is equal to 68.04 kilograms.
```