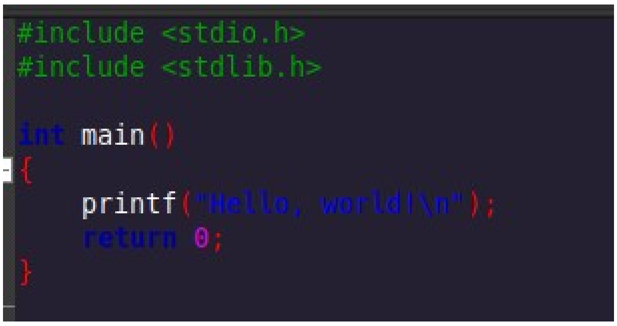
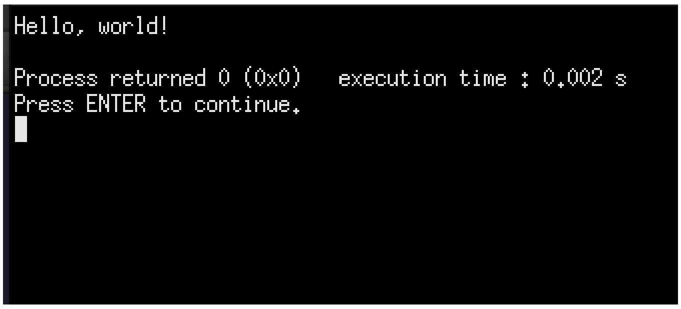
1.Introduction to Programming & C

Write a C program to print "Hello, World!" on the screen.

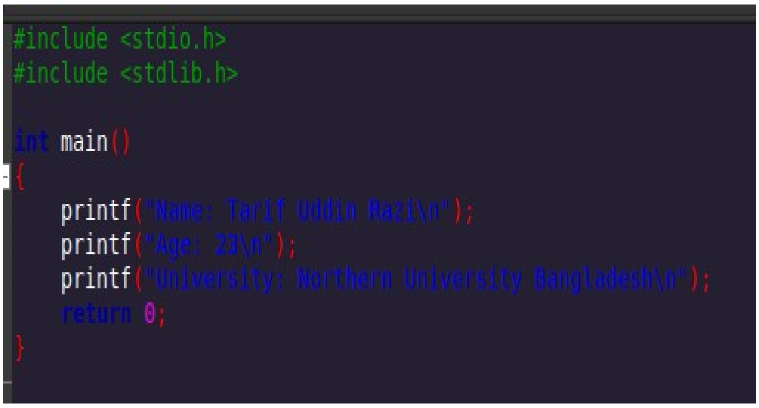


```c
#include <stdio.h>
#include <stdlib.h>

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

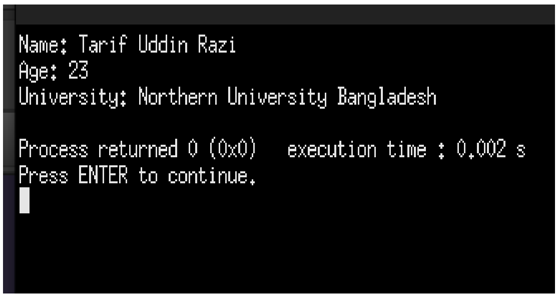


```
Hello, world!

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

Write a program to display your name, age, and university using printf.

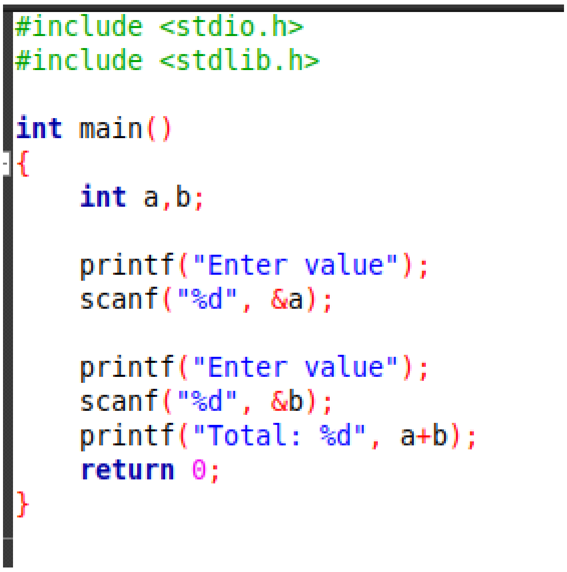


```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Name: Tarif Uddin Razi\n");
    printf("Age: 23\n");
    printf("University: Northern University Bangladesh\n");
    return 0;
}
```

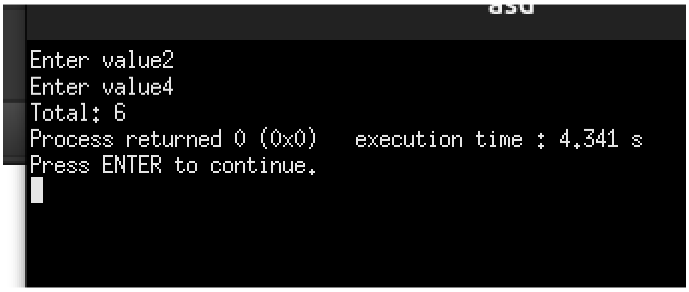


```
Name: Tarif Uddin Razi
Age: 23
University: Northern University Bangladesh

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

Create a program that prints the sum of two numbers entered by the user.

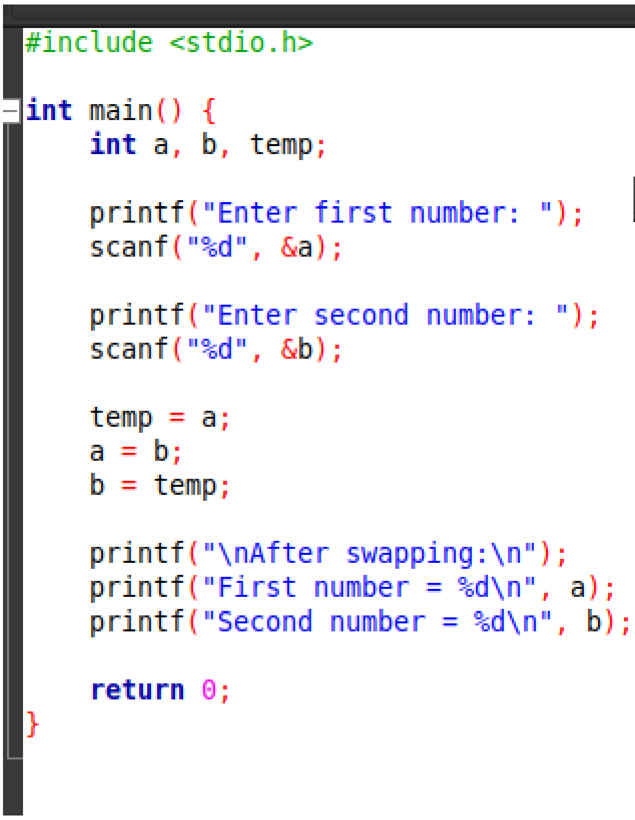


```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a,b;

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

    printf("Enter value");
    scanf("%d", &b);
    printf("Total: %d", a+b);
    return 0;
}
```

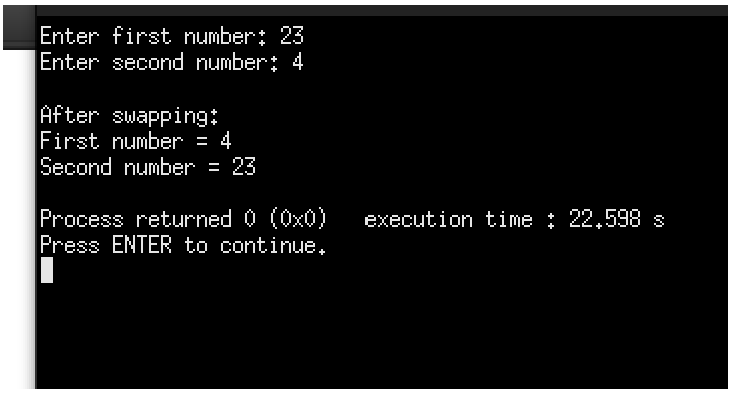


```
Enter value2
Enter value4
Total: 6
Process returned 0 (0x0)   execution time : 4.341 s
Press ENTER to continue.
```

Write a C program to swap two numbers using a temporary variable.

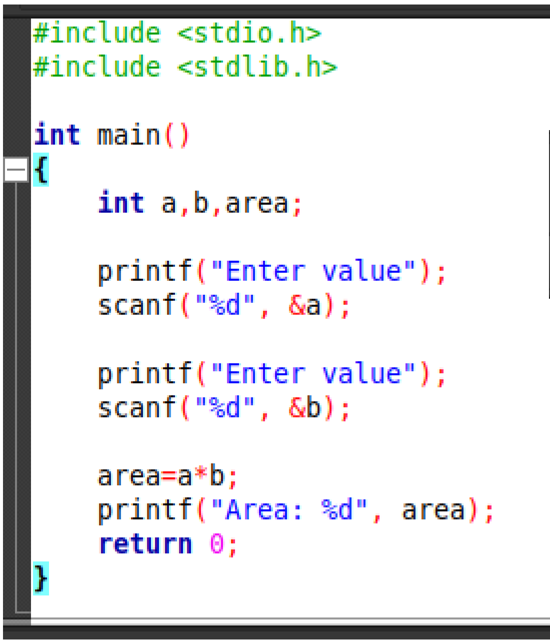


```c
#include <stdio.h>

int main() {
    int a, b, temp;

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

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

    temp = a;
    a = b;
    b = temp;

    printf("\nAfter swapping:\n");
    printf("First number = %d\n", a);
    printf("Second number = %d\n", b);

    return 0;
}
```

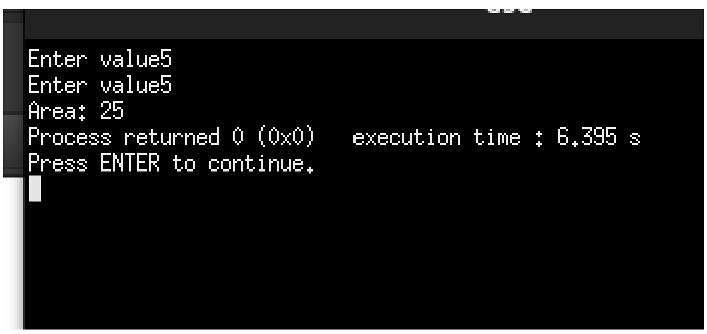


```
Enter first number: 23
Enter second number: 4

After swapping:
First number = 4
Second number = 23

Process returned 0 (0x0)   execution time : 22.598 s
Press ENTER to continue.
```

Write a program to calculate the area of a rectangle.

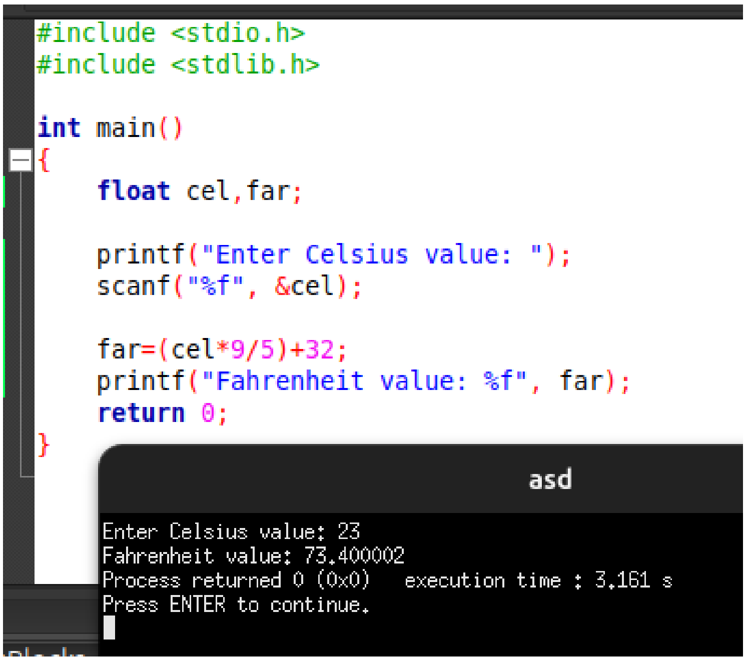


```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a,b,area;

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

    printf("Enter value");
    scanf("%d", &b);

    area=a*b;
    printf("Area: %d", area);
    return 0;
}
```



```
Enter value5
Enter value5
Area: 25
Process returned 0 (0x0)   execution time : 6.395 s
Press ENTER to continue.
```

Write a program to convert temperature from Celsius to Fahrenheit.

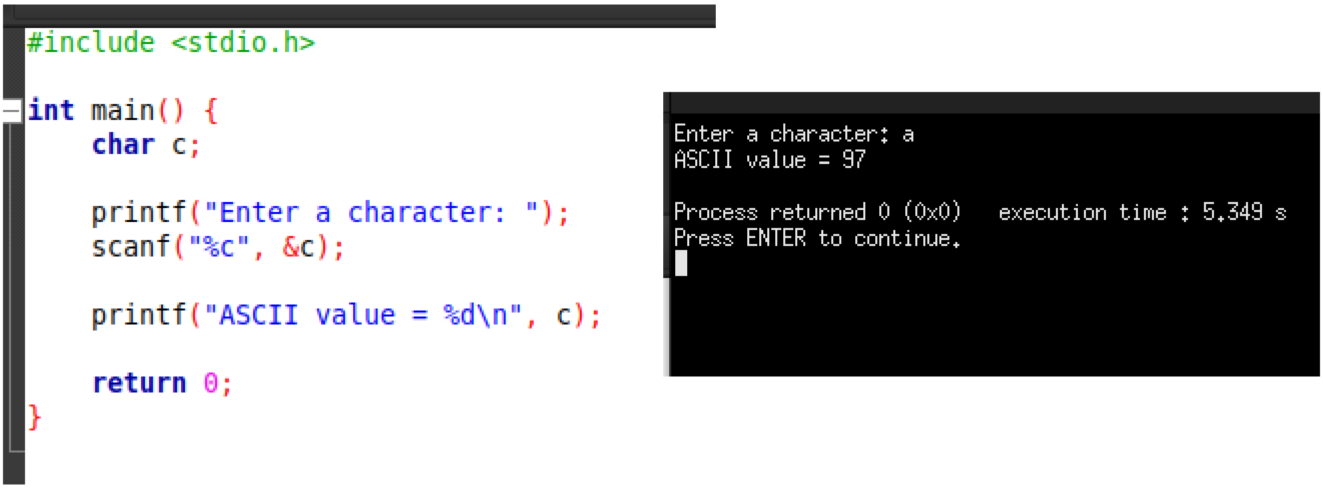


```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float cel,far;

    printf("Enter Celsius value: ");
    scanf("%f", &cel);

    far=(cel*9/5)+32;
    printf("Fahrenheit value: %f", far);
    return 0;
}
```

asd

```
Enter Celsius value: 23
Fahrenheit value: 73.400002
Process returned 0 (0x0)   execution time : 3.161 s
Press ENTER to continue.
```

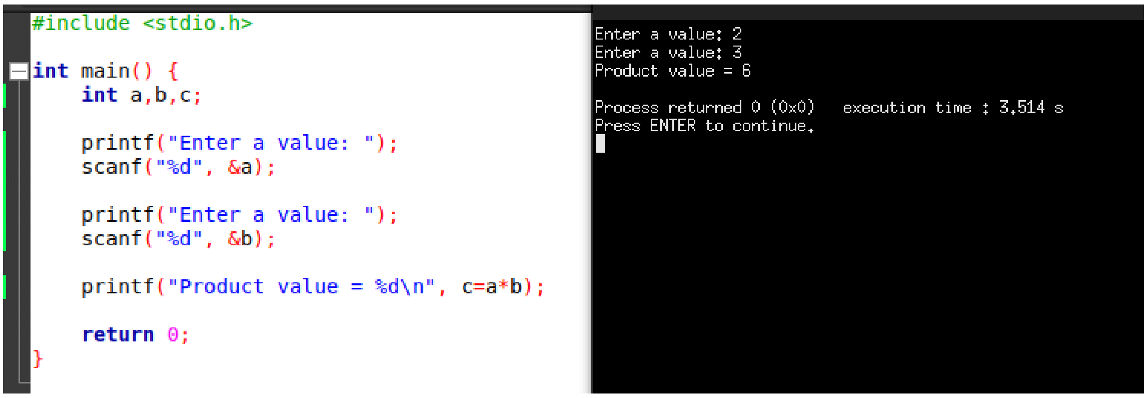
Write a program to display the ASCII value of a character entered by the user.



```c
#include <stdio.h>

int main() {
    char c;

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

    printf("ASCII value = %d\n", c);

    return 0;
}
```

```
Enter a character: a
ASCII value = 97

Process returned 0 (0x0)   execution time : 5.349 s
Press ENTER to continue.
```

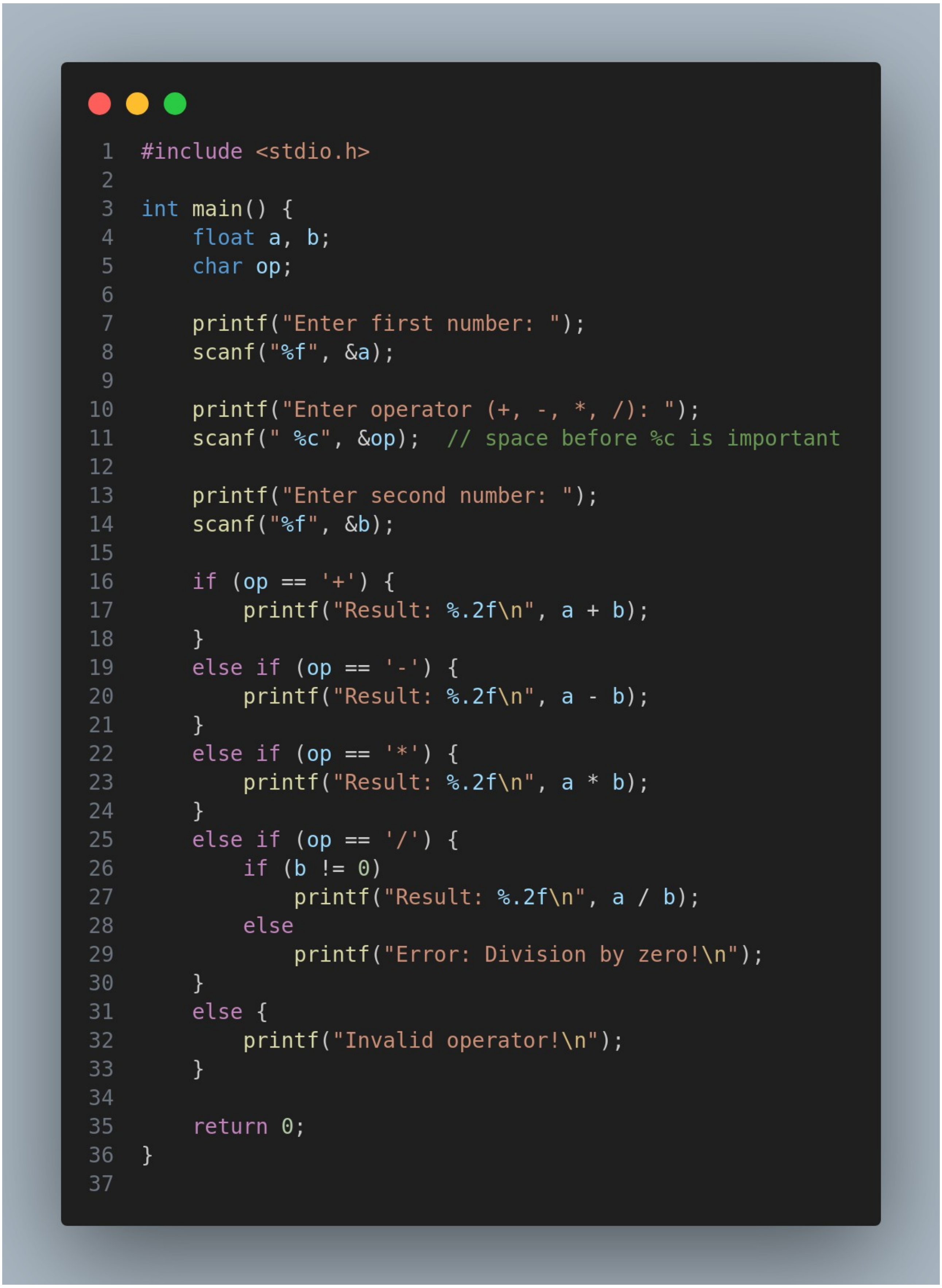
Write a program to take user input for two integers and display their product.



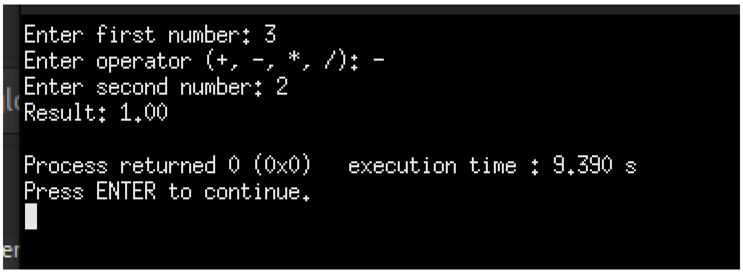
```c
#include <stdio.h>

int main() {
    int a,b,c;

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

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

    printf("Product value = %d\n", c=a*b);

    return 0;
}
```

```
Enter a value: 2
Enter a value: 3
Product value = 6

Process returned 0 (0x0)   execution time : 3.514 s
Press ENTER to continue.
```

Create a simple calculator that performs addition, subtraction, multiplication, and division.

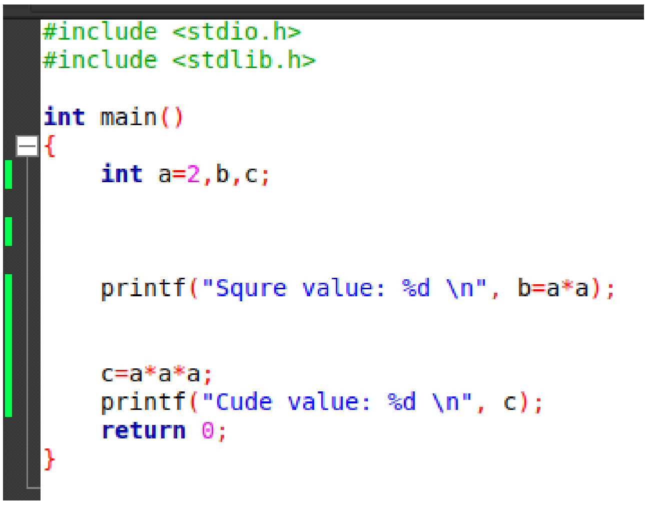


```c
#include <stdio.h>

int main() {
    float a, b;
    char op;

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

    printf("Enter operator (+, -, *, /): ");
    scanf(" %c", &op);  // space before %c is important

    printf("Enter second number: ");
    scanf("%f", &b);

    if (op == '+') {
        printf("Result: %.2f\n", a + b);
    }
    else if (op == '-') {
        printf("Result: %.2f\n", a - b);
    }
    else if (op == '*') {
        printf("Result: %.2f\n", a * b);
    }
    else if (op == '/') {
        if (b != 0)
            printf("Result: %.2f\n", a / b);
        else
            printf("Error: Division by zero!\n");
    }
    else {
        printf("Invalid operator!\n");
    }

    return 0;
}

```

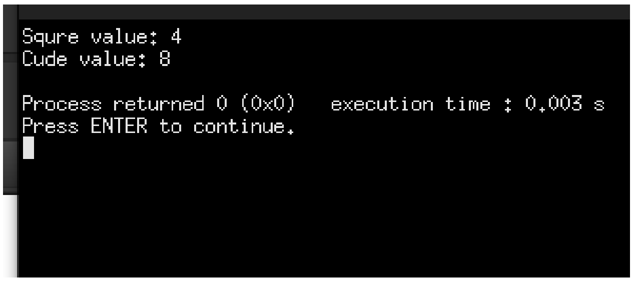


```
Enter first number: 3
Enter operator (+, -, *, /): -
Enter second number: 2
Result: 1.00

Process returned 0 (0x0)   execution time : 9.390 s
Press ENTER to continue.
```

Write a C program to find the square and cube of a given number.



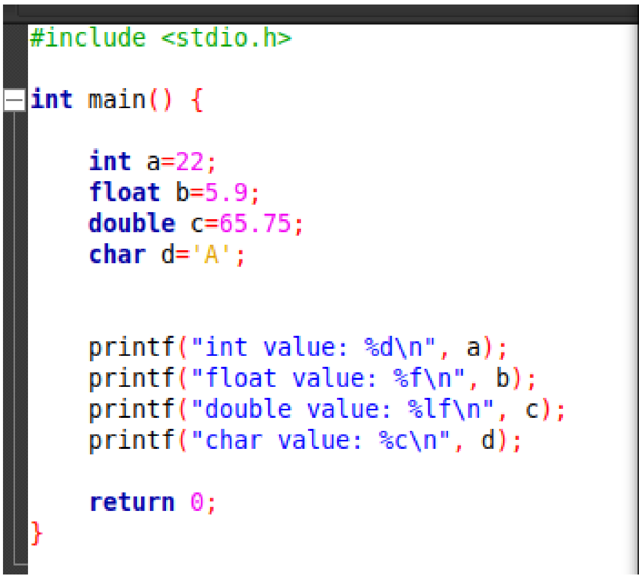
```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a=2,b,c;


    printf("Squre value: %d \n", b=a*a);


    c=a*a*a;
    printf("Cude value: %d \n", c);
    return 0;
}
```



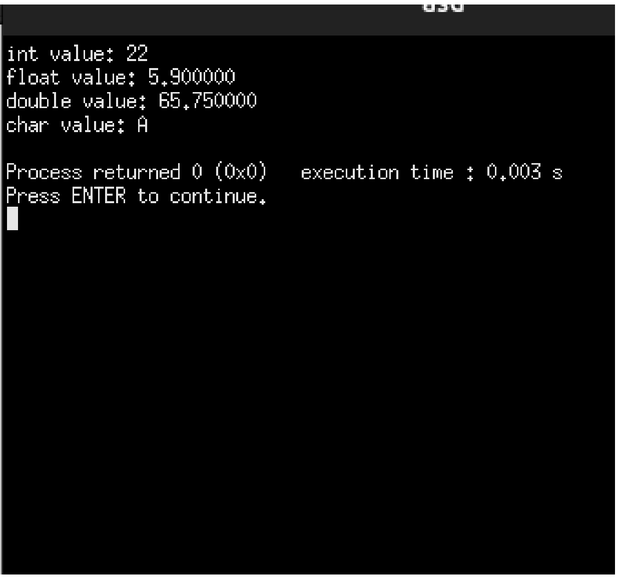
```
Squre value: 4
Cude value: 8

Process returned 0 (0x0)   execution time : 0.003 s
Press ENTER to continue.
```

2. Basic Syntax & Data Types

Write a program to declare and initialize variables of different data types and print their values.

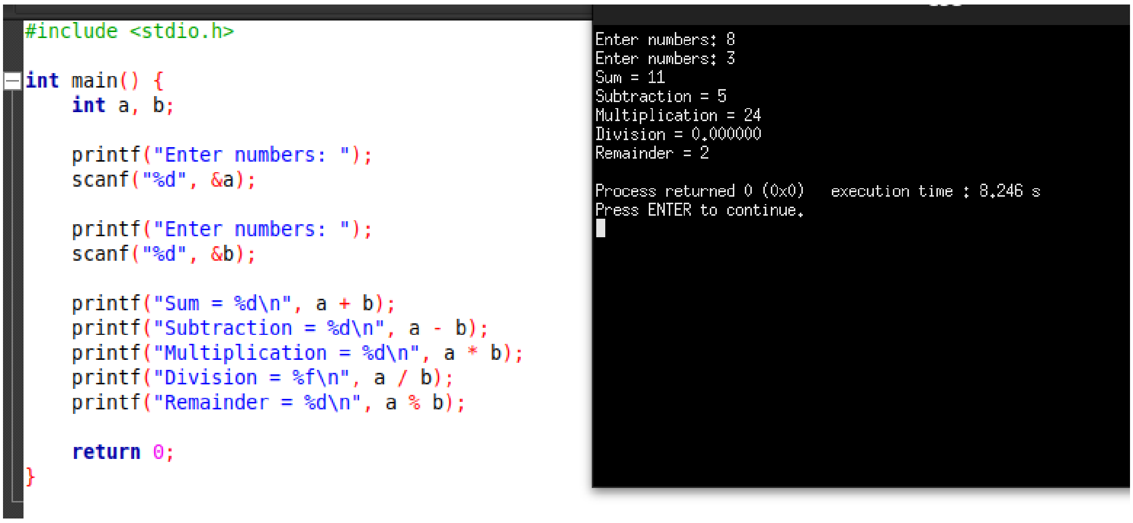


```c
#include <stdio.h>

int main() {

    int a=22;
    float b=5.9;
    double c=65.75;
    char d='A';


    printf("int value: %d\n", a);
    printf("float value: %f\n", b);
    printf("double value: %lf\n", c);
    printf("char value: %c\n", d);

    return 0;
}
```



```
int value: 22
float value: 5.900000
double value: 65.750000
char value: A

Process returned 0 (0x0)   execution time : 0.003 s
Press ENTER to continue.
```

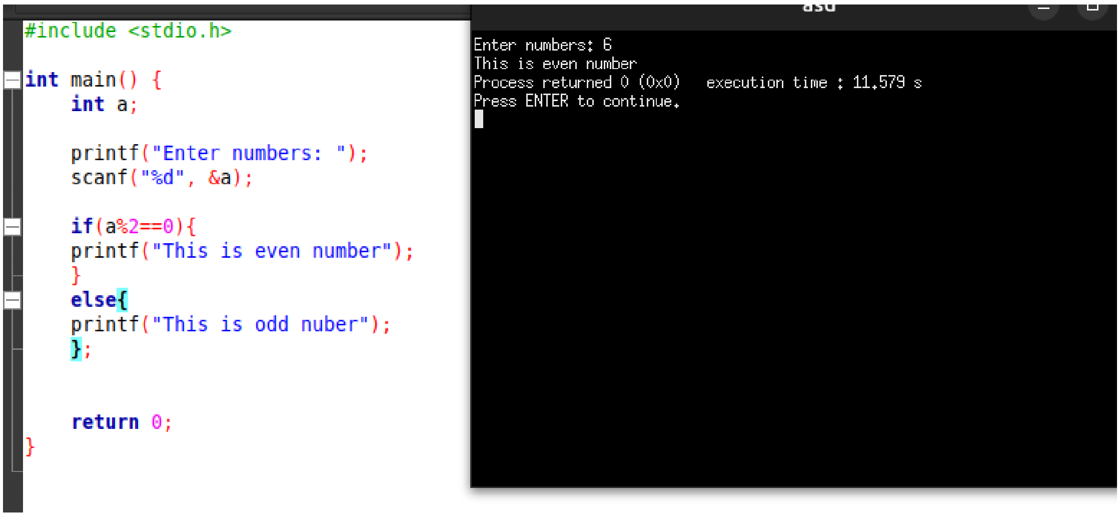
Write a program to take two integer inputs and perform all arithmetic operations on them.



```c
#include <stdio.h>

int main() {
    int a, b;

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

    printf("Enter numbers: ");
    scanf("%d", &b);

    printf("Sum = %d\n", a + b);
    printf("Subtraction = %d\n", a - b);
    printf("Multiplication = %d\n", a * b);
    printf("Division = %f\n", a / b);
    printf("Remainder = %d\n", a % b);

    return 0;
}
```

```
Enter numbers: 8
Enter numbers: 3
Sum = 11
Subtraction = 5
Multiplication = 24
Division = 0.000000
Remainder = 2

Process returned 0 (0x0)   execution time : 8.246 s
Press ENTER to continue.
```

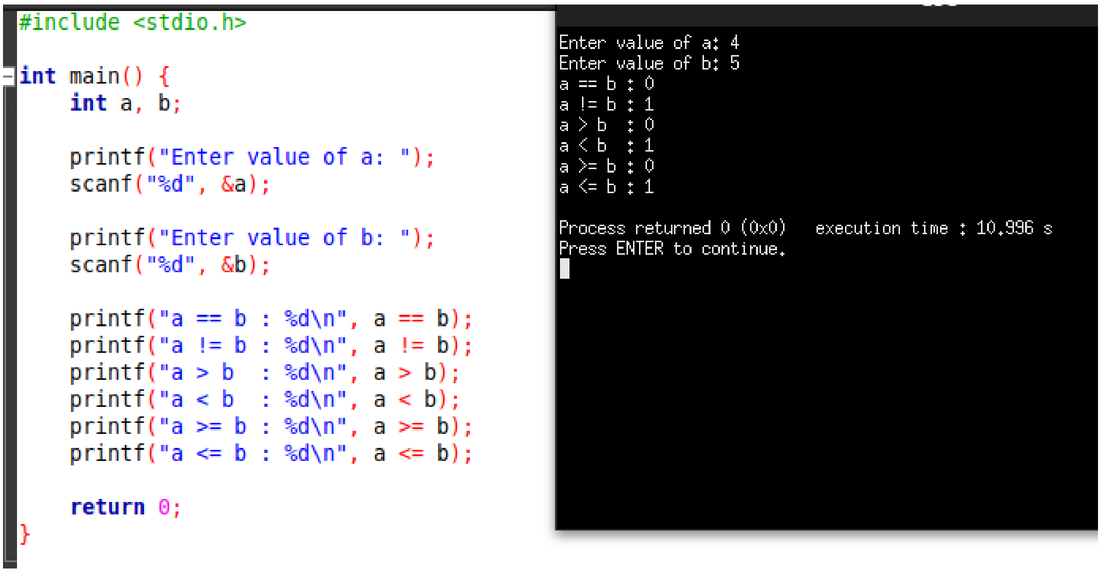
Create a program that checks whether a given number is even or odd.



```c
#include <stdio.h>

int main() {
    int a;

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

    if(a%2==0){
    printf("This is even number");
    }
    else{
    printf("This is odd nuber");
    };


    return 0;
}
```

```
Enter numbers: 6
This is even number
Process returned 0 (0x0)   execution time : 11.579 s
Press ENTER to continue.
```

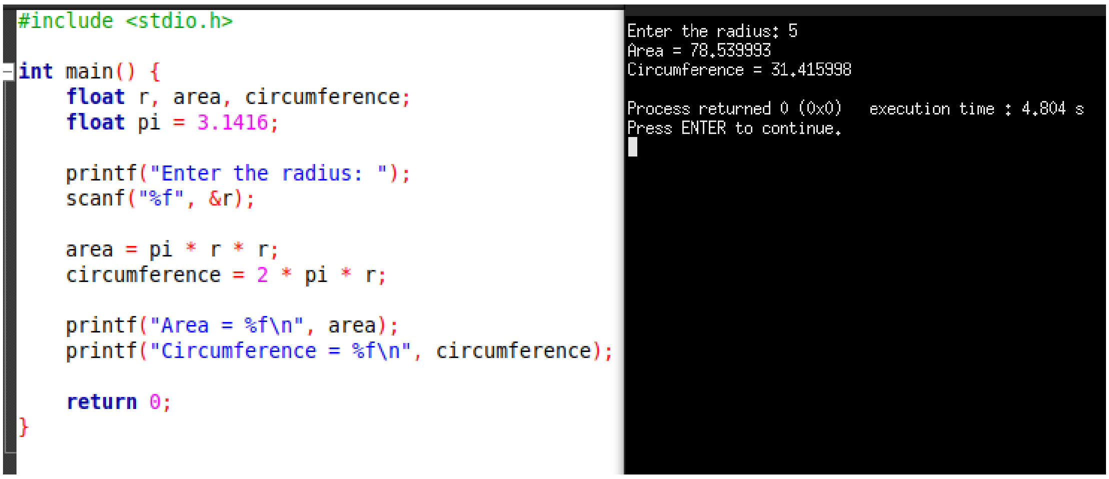
Write a program to demonstrate the use of relational operators.



```c
#include <stdio.h>

int main() {
    int a, b;

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

    printf("Enter value of b: ");
    scanf("%d", &b);

    printf("a == b : %d\n", a == b);
    printf("a != b : %d\n", a != b);
    printf("a > b  : %d\n", a > b);
    printf("a < b  : %d\n", a < b);
    printf("a >= b : %d\n", a >= b);
    printf("a <= b : %d\n", a <= b);

    return 0;
}
```

```
Enter value of a: 4
Enter value of b: 5
a == b : 0
a != b : 1
a > b  : 0
a < b  : 1
a >= b : 0
a <= b : 1

Process returned 0 (0x0)   execution time : 10.996 s
Press ENTER to continue.
```

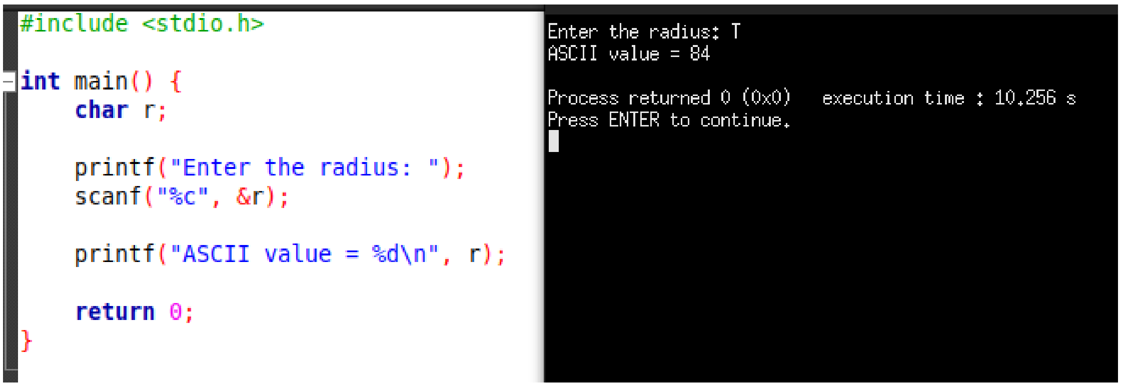
Write a program to compute the area and circumference of a circle using float data type.



```c
#include <stdio.h>

int main() {
    float r, area, circumference;
    float pi = 3.1416;

    printf("Enter the radius: ");
    scanf("%f", &r);

    area = pi * r * r;
    circumference = 2 * pi * r;

    printf("Area = %f\n", area);
    printf("Circumference = %f\n", circumference);

    return 0;
}
```

```
Enter the radius: 5
Area = 78.539993
Circumference = 31.415998

Process returned 0 (0x0)   execution time : 4.804 s
Press ENTER to continue.
```

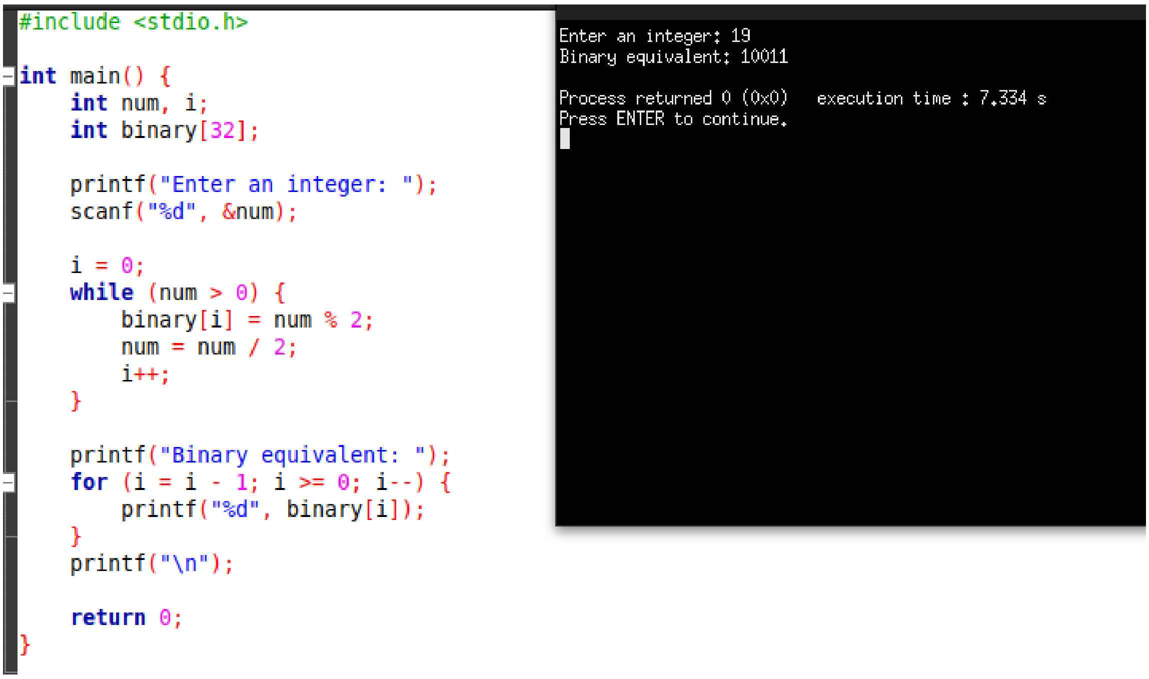
Write a program that takes a character as input and prints its ASCII value.



```c
#include <stdio.h>

int main() {
    char r;

    printf("Enter the radius: ");
    scanf("%c", &r);

    printf("ASCII value = %d\n", r);

    return 0;
}
```

```
Enter the radius: T
ASCII value = 84

Process returned 0 (0x0)   execution time : 10.256 s
Press ENTER to continue.
```

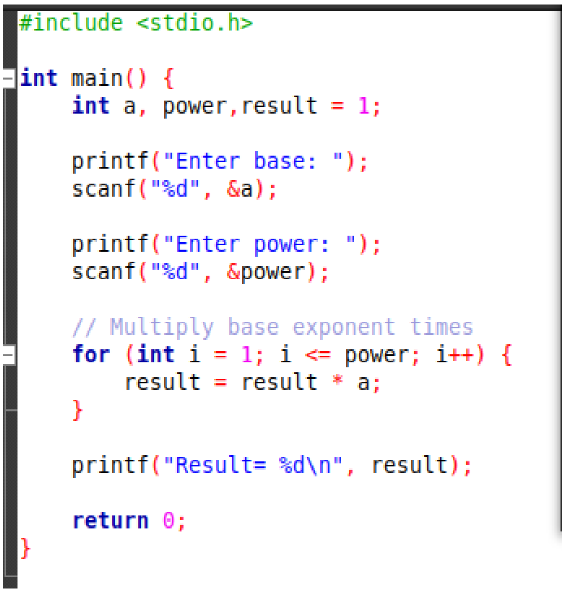
Write a program to take an integer input and display its binary equivalent.



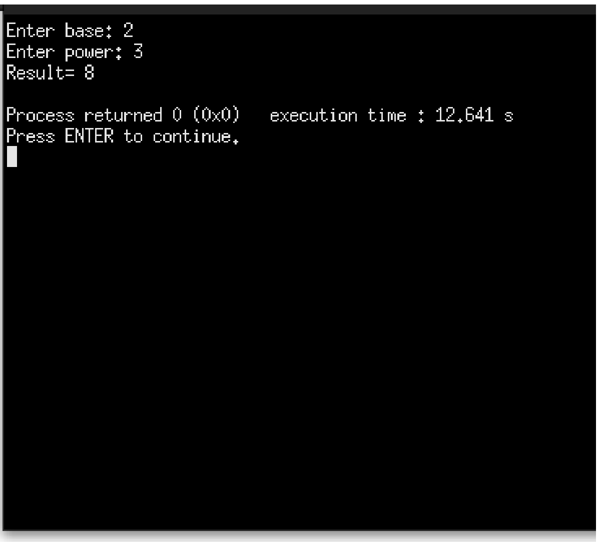
```c
#include <stdio.h>

int main() {
    int num, i;
    int binary[32];

    printf("Enter an integer: ");
    scanf("%d", &num);

    i = 0;
    while (num > 0) {
        binary[i] = num % 2;
        num = num / 2;
        i++;
    }

    printf("Binary equivalent: ");
    for (i = i - 1; i >= 0; i--) {
        printf("%d", binary[i]);
    }
    printf("\n");

    return 0;
}
```

```
Enter an integer: 19
Binary equivalent: 10011

Process returned 0 (0x0)   execution time : 7.334 s
Press ENTER to continue.
```

Create a program that calculates the power of a number without using the pow() function.

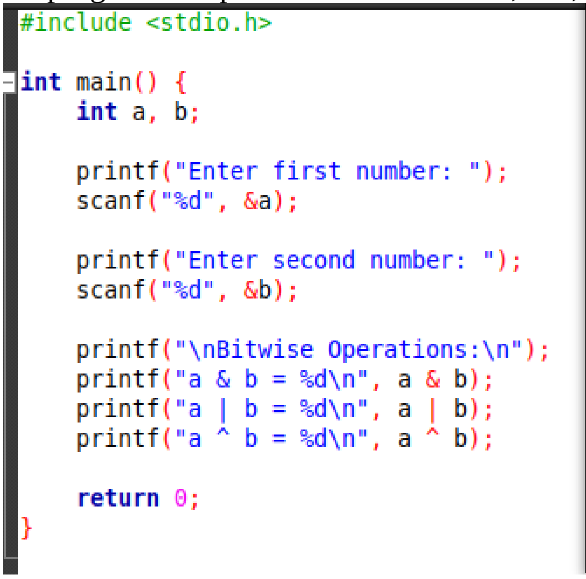


```c
#include <stdio.h>

int main() {
    int a, power,result = 1;

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

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

    // Multiply base exponent times
    for (int i = 1; i <= power; i++) {
        result = result * a;
    }

    printf("Result= %d\n", result);

    return 0;
}
```

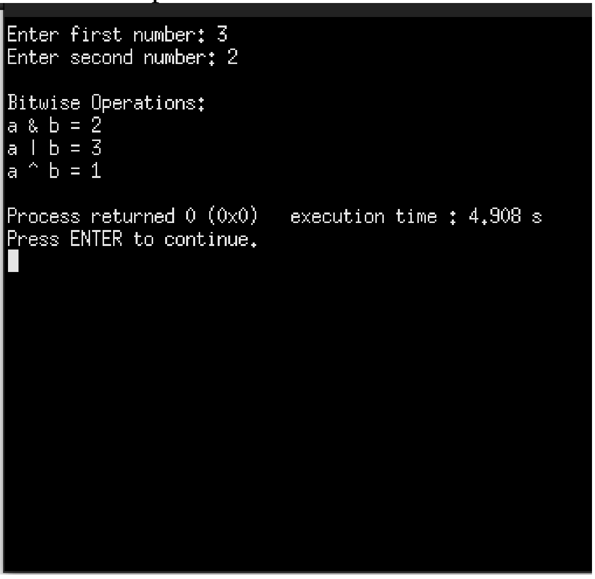


```
Enter base: 2
Enter power: 3
Result= 8

Process returned 0 (0x0)   execution time : 12.641 s
Press ENTER to continue.
```

Write a program that performs bitwise AND, OR, and XOR operations on two numbers.

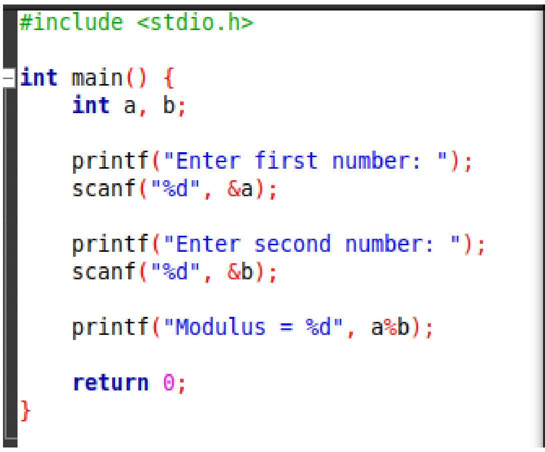


```c
#include <stdio.h>

int main() {
    int a, b;

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

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

    printf("\nBitwise Operations:\n");
    printf("a & b = %d\n", a & b);
    printf("a | b = %d\n", a | b);
    printf("a ^ b = %d\n", a ^ b);

    return 0;
}
```

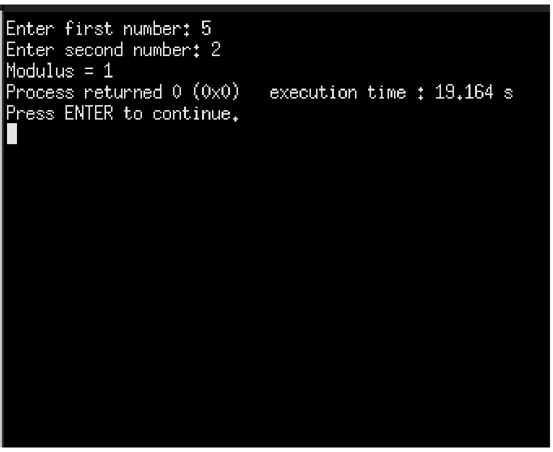


```
Enter first number: 3
Enter second number: 2

Bitwise Operations:
a & b = 2
a | b = 3
a ^ b = 1

Process returned 0 (0x0)   execution time : 4.908 s
Press ENTER to continue.
```

Write a program to demonstrate the use of the modulus operator.



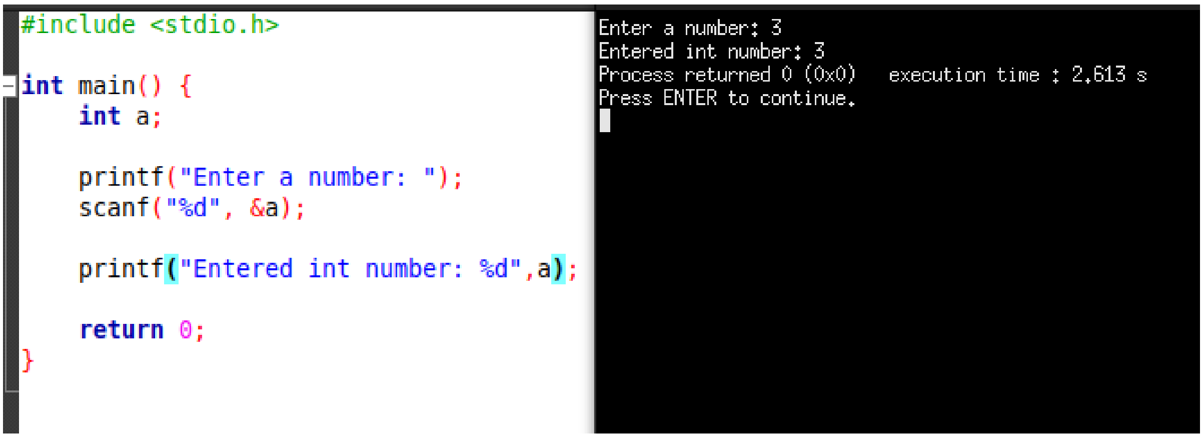
```c
#include <stdio.h>

int main() {
    int a, b;

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

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

    printf("Modulus = %d", a%b);

    return 0;
}
```



```
Enter first number: 5
Enter second number: 2
Modulus = 1
Process returned 0 (0x0)   execution time : 19.164 s
Press ENTER to continue.
```

2. Input and Output

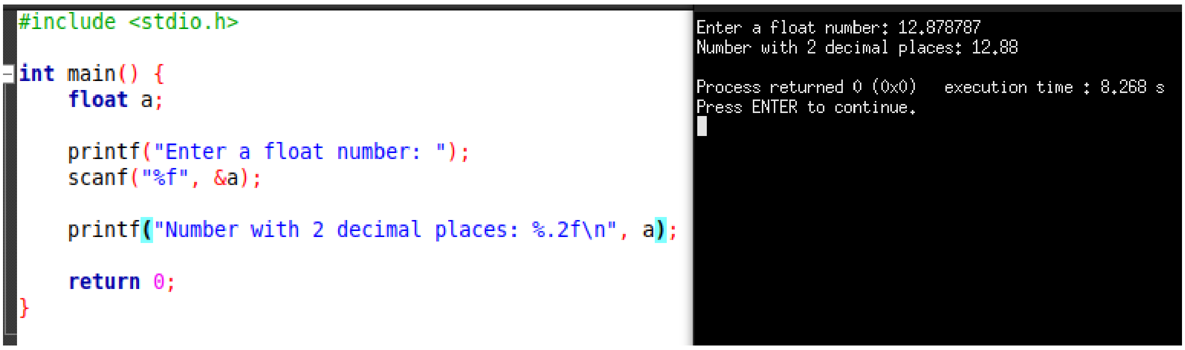
Write a program to take an integer as input and print it.



```
#include <stdio.h>

int main() {
    int a;

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

    printf("Entered int number: %d",a);

    return 0;
}
```

```
Enter a number: 3
Entered int number: 3
Process returned 0 (0x0)   execution time : 2.613 s
Press ENTER to continue.
```

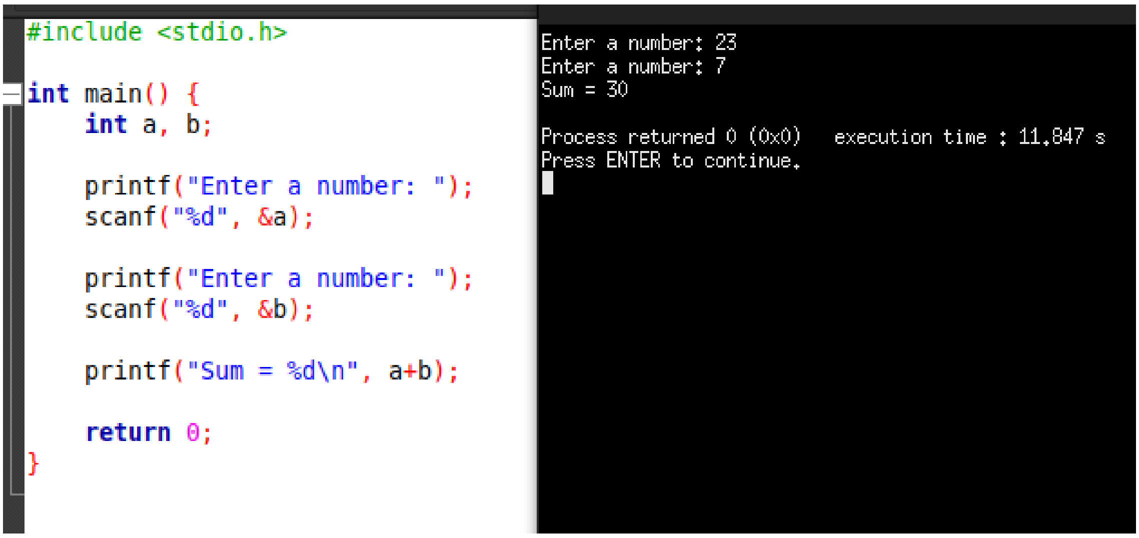
Write a program to take a floating-point number as input and display it with two decimal places.



```
#include <stdio.h>

int main() {
    float a;

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

    printf("Number with 2 decimal places: %.2f\n", a);

    return 0;
}
```

```
Enter a float number: 12.878787
Number with 2 decimal places: 12.88

Process returned 0 (0x0)   execution time : 8.268 s
Press ENTER to continue.
```

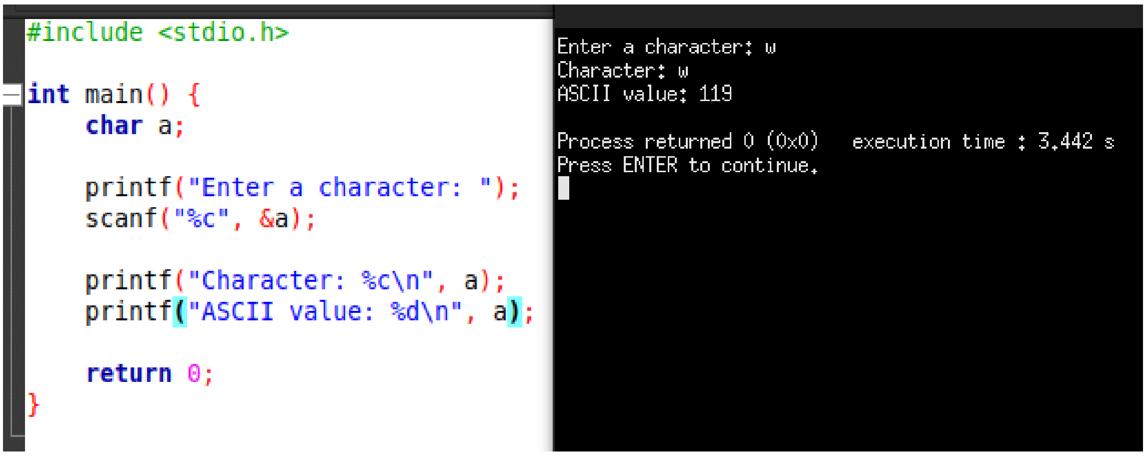
Create a program that takes two integer inputs and prints their sum.



```
#include <stdio.h>

int main() {
    int a, b;

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

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

    printf("Sum = %d\n", a+b);

    return 0;
}
```

```
Enter a number: 23
Enter a number: 7
Sum = 30

Process returned 0 (0x0)   execution time : 11.847 s
Press ENTER to continue.
```

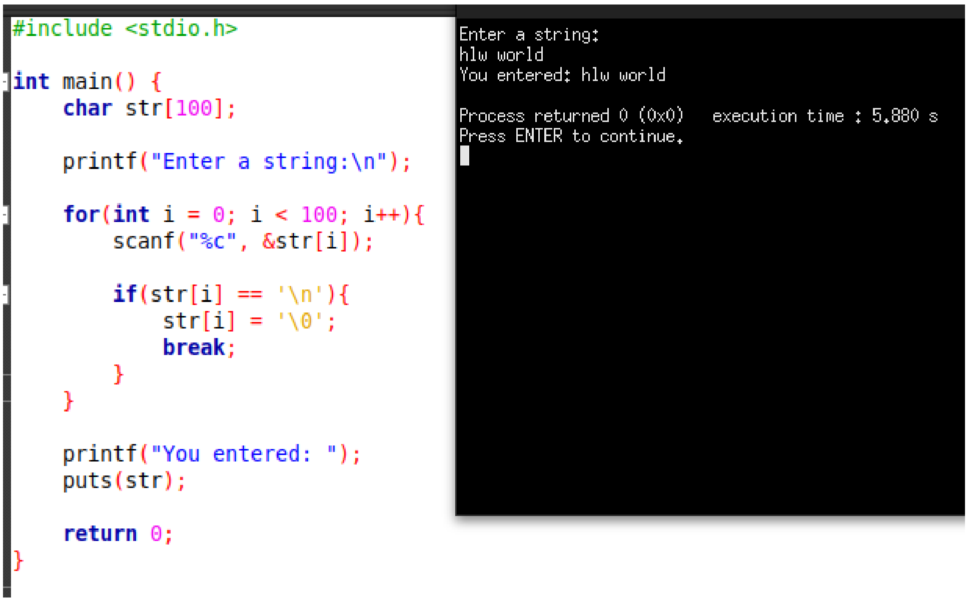
Write a program that reads a character and prints it along with its ASCII value.



```c
#include <stdio.h>

int main() {
    char a;

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

    printf("Character: %c\n", a);
    printf("ASCII value: %d\n", a);

    return 0;
}
```

```
Enter a character: w
Character: w
ASCII value: 119

Process returned 0 (0x0)   execution time : 3.442 s
Press ENTER to continue.
```

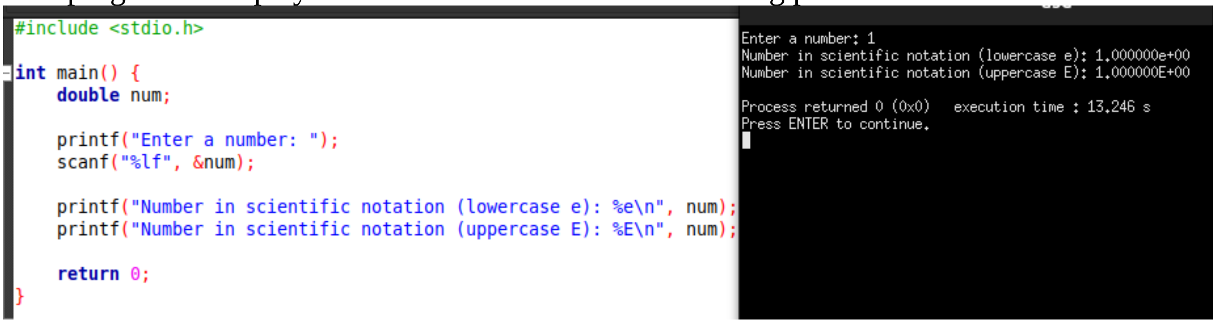
Write a program to take a sentence as input and display it using puts().



```c
#include <stdio.h>

int main() {
    char str[100];

    printf("Enter a string:\n");

    for(int i = 0; i < 100; i++){
        scanf("%c", &str[i]);

        if(str[i] == '\n'){
            str[i] = '\0';
            break;
        }
    }

    printf("You entered: ");
    puts(str);

    return 0;
}
```

```
Enter a string:
hlw world
You entered: hlw world

Process returned 0 (0x0)   execution time : 5.880 s
Press ENTER to continue.
```

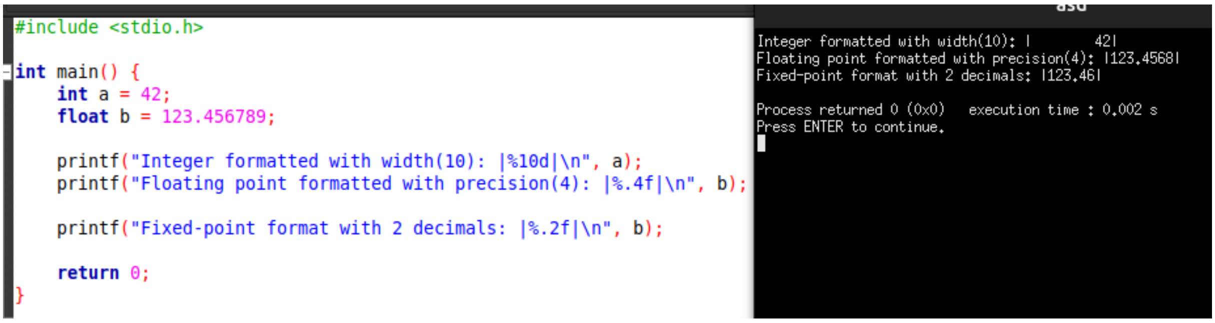
Write a program to display a number in scientific notation using printf.



```c
#include <stdio.h>

int main() {
    double num;

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

    printf("Number in scientific notation (lowercase e): %e\n", num);
    printf("Number in scientific notation (uppercase E): %E\n", num);

    return 0;
}
```

```
Enter a number: 1
Number in scientific notation (lowercase e): 1.000000e+00
Number in scientific notation (uppercase E): 1.000000E+00

Process returned 0 (0x0)   execution time : 13.246 s
Press ENTER to continue.
```

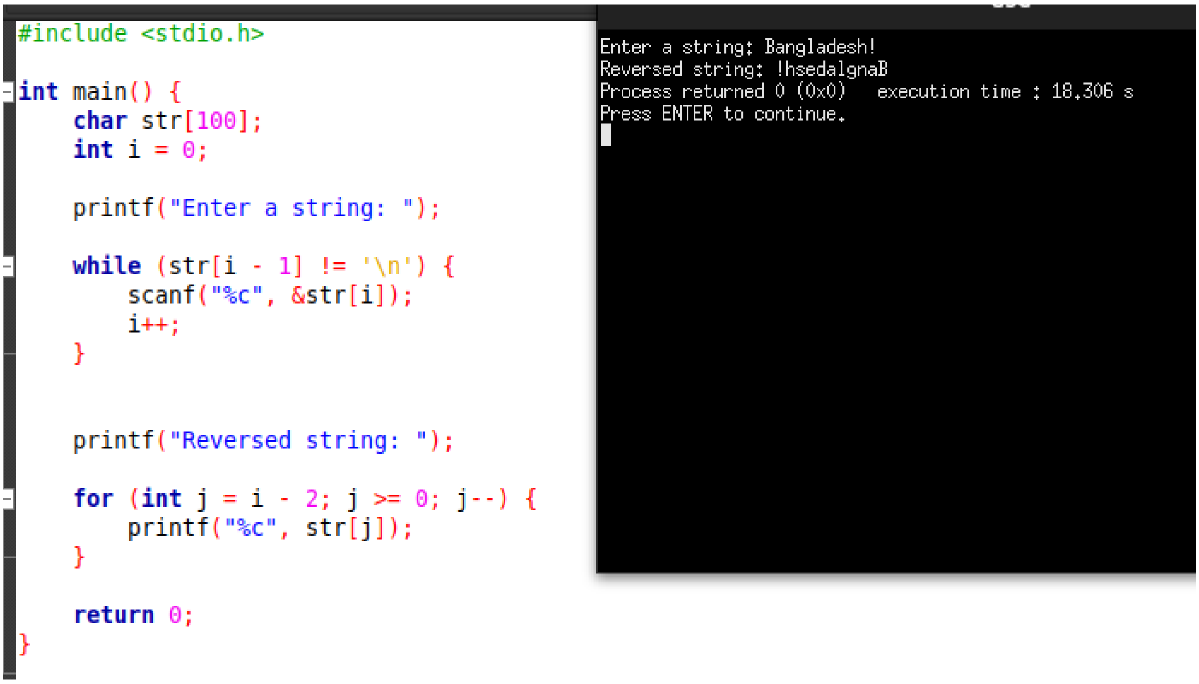
Create a program that formats the output using setw(), setprecision(), and fixed. (Used C here)



```c
#include <stdio.h>

int main() {
    int a = 42;
    float b = 123.456789;

    printf("Integer formatted with width(10): |%10d|\n", a);
    printf("Floating point formatted with precision(4): |%.4f|\n", b);

    printf("Fixed-point format with 2 decimals: |%.2f|\n", b);

    return 0;
}
```

```
Integer formatted with width(10): |        42|
Floating point formatted with precision(4): |123.4568|
Fixed-point format with 2 decimals: |123.46|

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

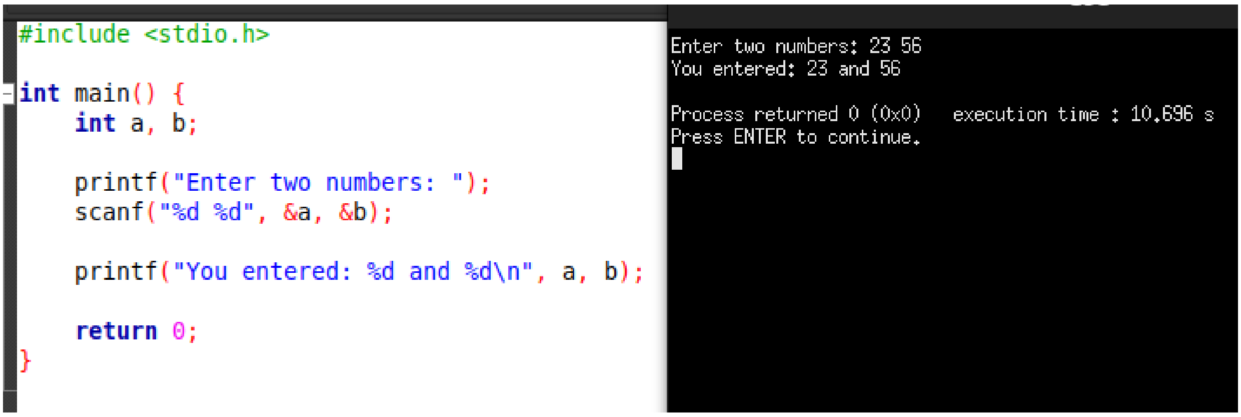
Write a program that takes user input and displays it in reverse order.



```c
#include <stdio.h>

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

    printf("Enter a string: ");

    while (str[i - 1] != '\n') {
        scanf("%c", &str[i]);
        i++;
    }


    printf("Reversed string: ");

    for (int j = i - 2; j >= 0; j--) {
        printf("%c", str[j]);
    }

    return 0;
}
```

```
Enter a string: Bangladesh!
Reversed string: !hsedalgnaB
Process returned 0 (0x0)   execution time : 18.306 s
Press ENTER to continue.
```

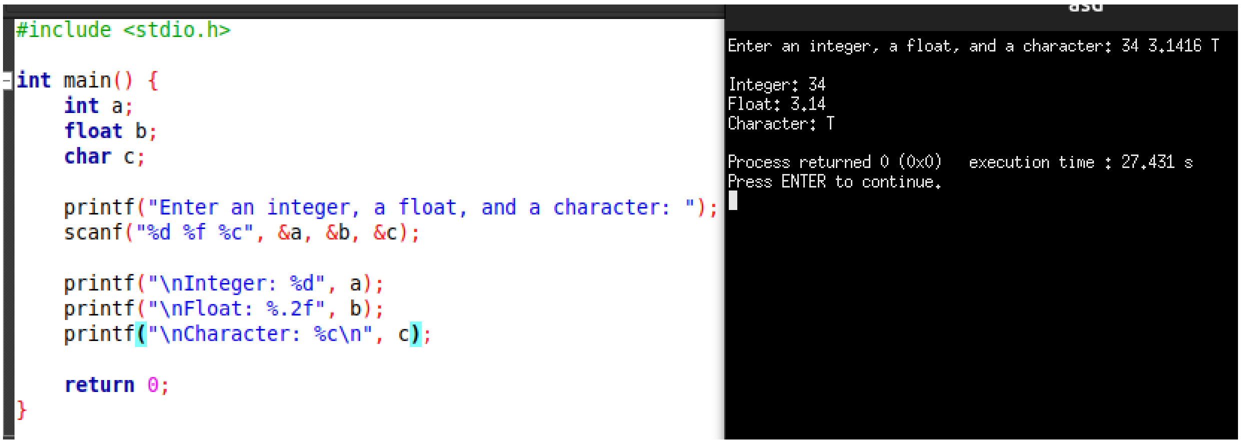
Write a program to read two space-separated integers in a single input statement and print them.



```c
#include <stdio.h>

int main() {
    int a, b;

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

    printf("You entered: %d and %d\n", a, b);

    return 0;
}
```

```
Enter two numbers: 23 56
You entered: 23 and 56

Process returned 0 (0x0)   execution time : 10.696 s
Press ENTER to continue.
```

Write a program that takes three inputs and displays them in different data types.

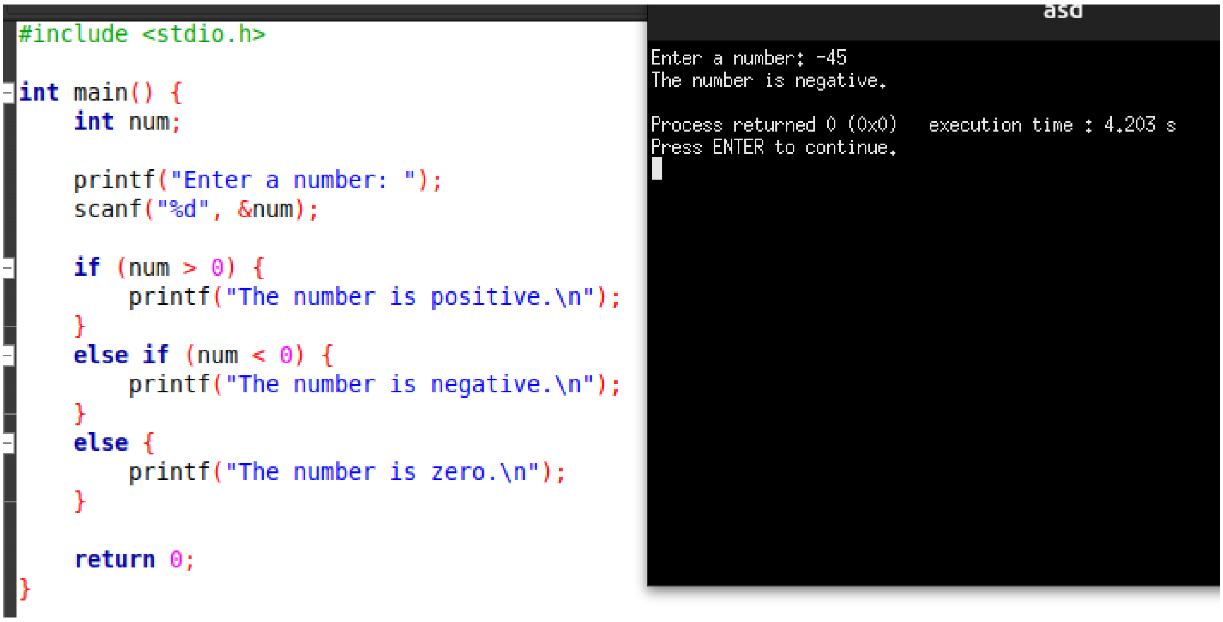


```c
#include <stdio.h>

int main() {
    int a;
    float b;
    char c;

    printf("Enter an integer, a float, and a character: ");
    scanf("%d %f %c", &a, &b, &c);

    printf("\nInteger: %d", a);
    printf("\nFloat: %.2f", b);
    printf("\nCharacter: %c\n", c);

    return 0;
}
```

```
Enter an integer, a float, and a character: 34 3.1416 T

Integer: 34
Float: 3.14
Character: T

Process returned 0 (0x0)   execution time : 27.431 s
Press ENTER to continue.
```

4. Control Flow Statements

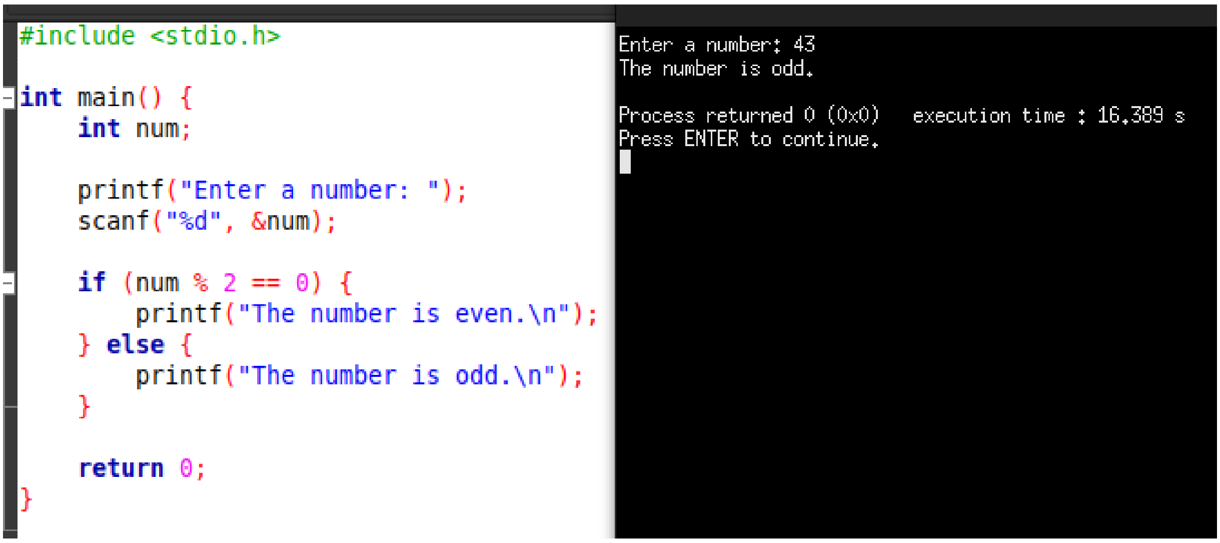
Write a program to check if a number is positive, negative, or zero.



```c
#include <stdio.h>

int main() {
    int num;

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

    if (num > 0) {
        printf("The number is positive.\n");
    }
    else if (num < 0) {
        printf("The number is negative.\n");
    }
    else {
        printf("The number is zero.\n");
    }

    return 0;
}
```

```
Enter a number: -45
The number is negative.

Process returned 0 (0x0)   execution time : 4.203 s
Press ENTER to continue.
```

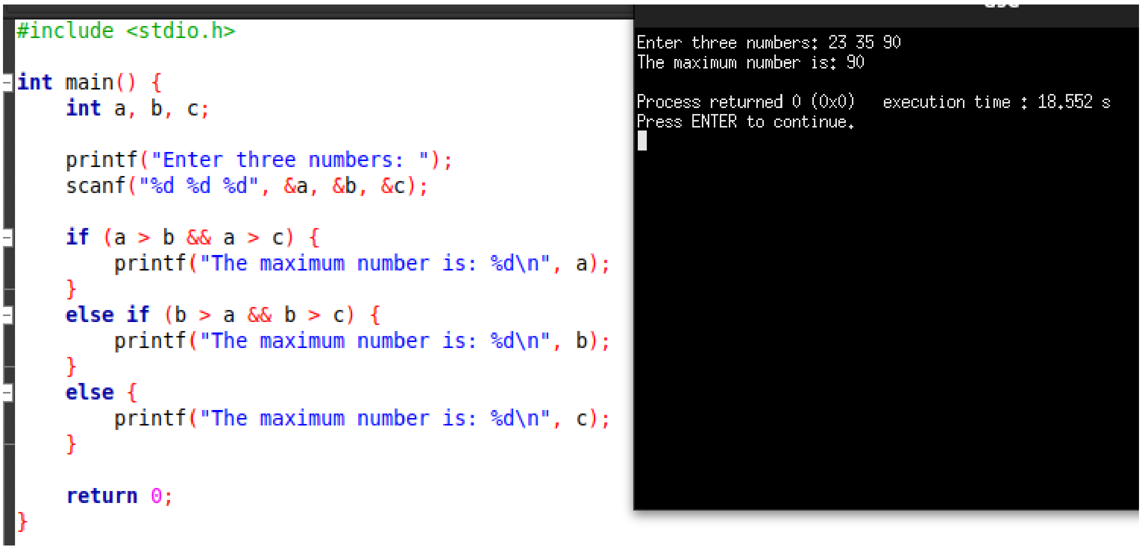
Write a program that checks if a number is even or odd using the if-else statement.



```c
#include <stdio.h>

int main() {
    int num;

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

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

    return 0;
}
```

```
Enter a number: 43
The number is odd.

Process returned 0 (0x0)   execution time : 16.389 s
Press ENTER to continue.
```

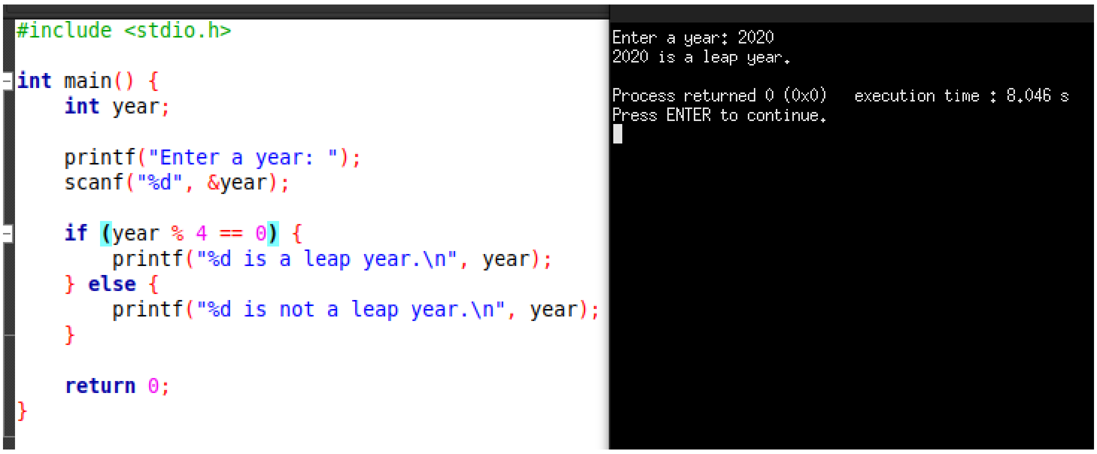
Create a program that finds the maximum of three numbers using if-else.



```c
#include <stdio.h>

int main() {
    int a, b, c;

    printf("Enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    if (a > b && a > c) {
        printf("The maximum number is: %d\n", a);
    }
    else if (b > a && b > c) {
        printf("The maximum number is: %d\n", b);
    }
    else {
        printf("The maximum number is: %d\n", c);
    }

    return 0;
}
```

```
Enter three numbers: 23 35 90
The maximum number is: 90

Process returned 0 (0x0)   execution time : 18.552 s
Press ENTER to continue.
```

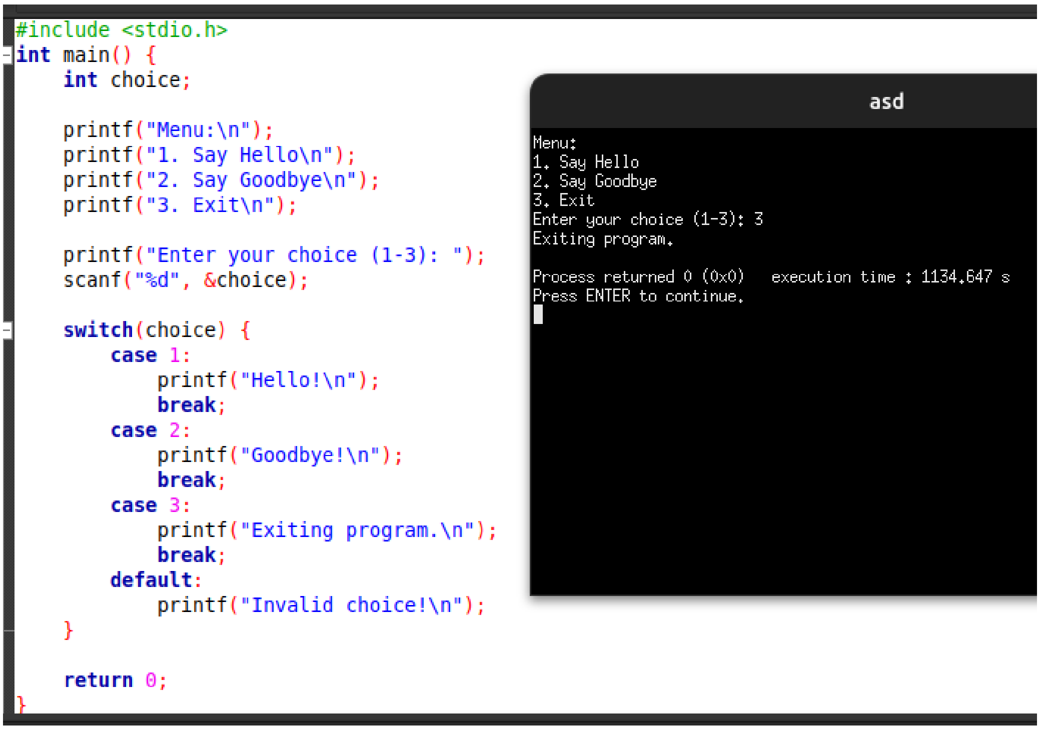
Write a program to check if a given year is a leap year.



```c
#include <stdio.h>

int main() {
    int year;

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

    if (year % 4 == 0) {
        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.

Process returned 0 (0x0)   execution time : 8.046 s
Press ENTER to continue.
```

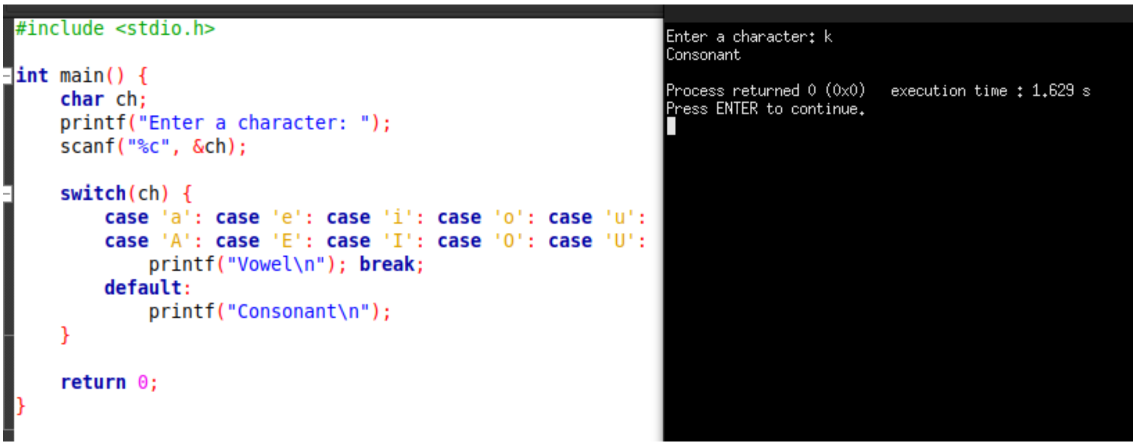
Create a program to display a menu and take user input using a switch statement.



```c
#include <stdio.h>
int main() {
    int choice;

    printf("Menu:\n");
    printf("1. Say Hello\n");
    printf("2. Say Goodbye\n");
    printf("3. Exit\n");

    printf("Enter your choice (1-3): ");
    scanf("%d", &choice);

    switch(choice) {
        case 1:
            printf("Hello!\n");
            break;
        case 2:
            printf("Goodbye!\n");
            break;
        case 3:
            printf("Exiting program.\n");
            break;
        default:
            printf("Invalid choice!\n");
    }

    return 0;
}
```

```
Menu:
1. Say Hello
2. Say Goodbye
3. Exit
Enter your choice (1-3): 3
Exiting program.

Process returned 0 (0x0)   execution time : 1134.647 s
Press ENTER to continue.
```

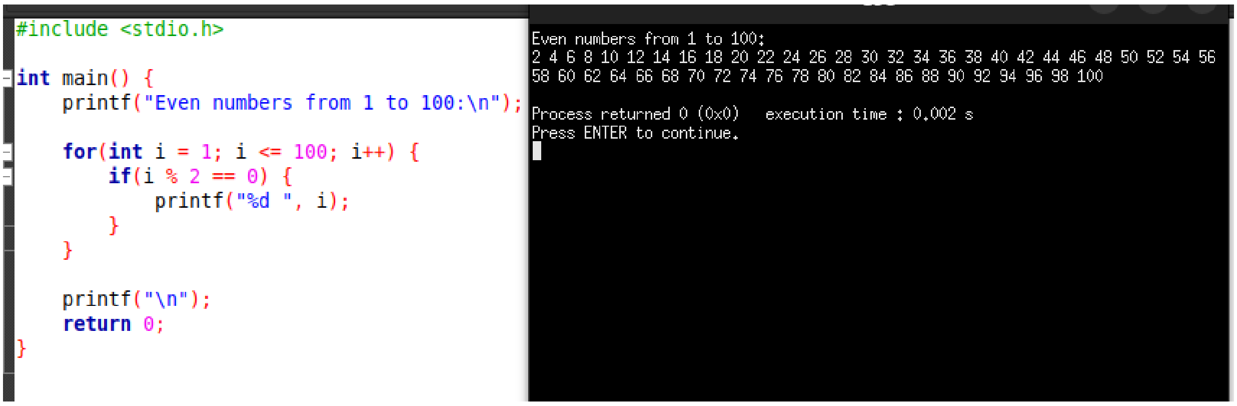
Write a program to check if a character is a vowel or consonant using switch.



```
#include <stdio.h>

int main() {
    char ch;
    printf("Enter a character: ");
    scanf("%c", &ch);

    switch(ch) {
        case 'a': case 'e': case 'i': case 'o': case 'u':
        case 'A': case 'E': case 'I': case 'O': case 'U':
            printf("Vowel\n"); break;
        default:
            printf("Consonant\n");
    }

    return 0;
}
```

```
Enter a character: k
Consonant

Process returned 0 (0x0)   execution time : 1.629 s
Press ENTER to continue.
```

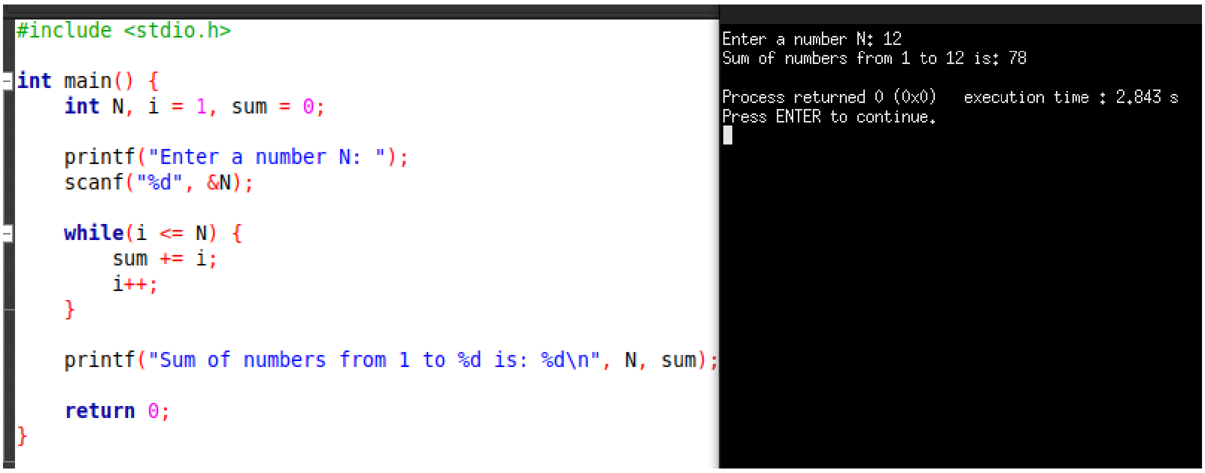
Write a program that prints all even numbers from 1 to 100 using a for loop.



```
#include <stdio.h>

int main() {
    printf("Even numbers from 1 to 100:\n");

    for(int i = 1; i <= 100; i++) {
        if(i % 2 == 0) {
            printf("%d ", i);
        }
    }

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

```
Even numbers from 1 to 100:
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56
58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

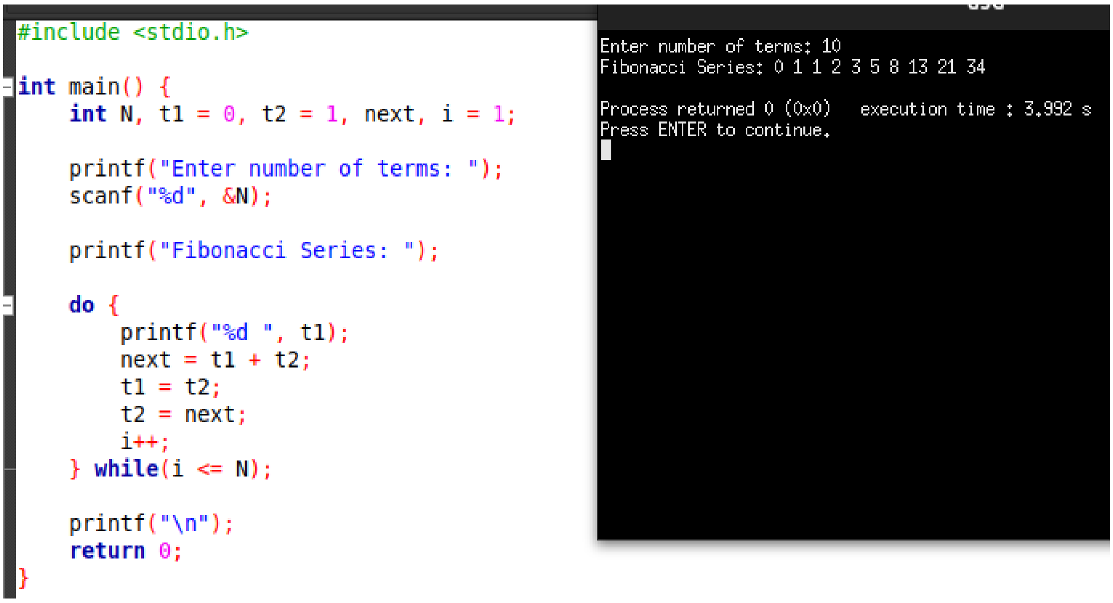
Write a program to print the sum of all numbers from 1 to N using a while loop.



```
#include <stdio.h>

int main() {
    int N, i = 1, sum = 0;

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

    while(i <= N) {
        sum += i;
        i++;
    }

    printf("Sum of numbers from 1 to %d is: %d\n", N, sum);

    return 0;
}
```

```
Enter a number N: 12
Sum of numbers from 1 to 12 is: 78

Process returned 0 (0x0)   execution time : 2.843 s
Press ENTER to continue.
```

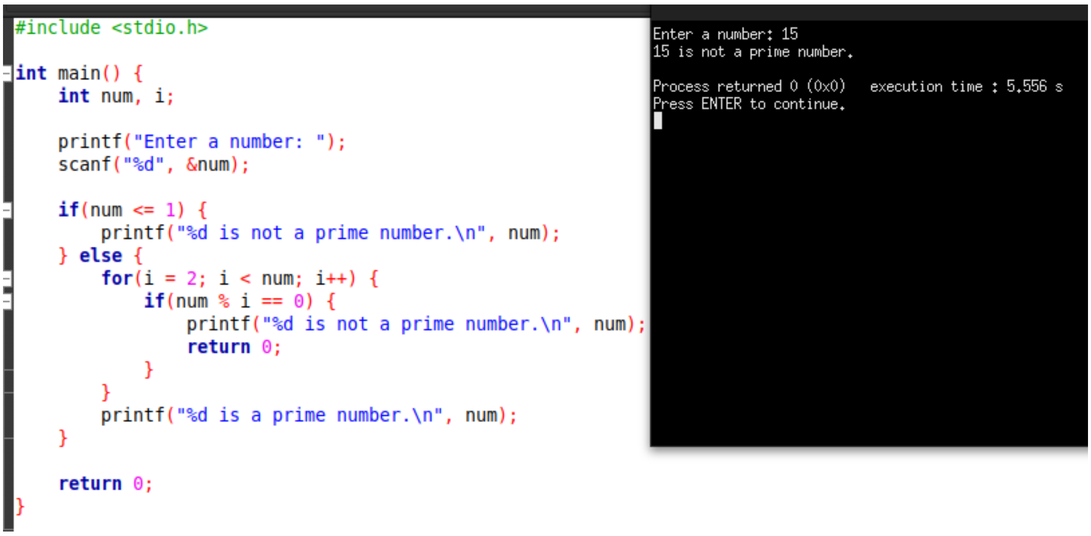
Create a program that generates the Fibonacci series up to N terms using a do-while loop.



```c
#include <stdio.h>

int main() {
    int N, t1 = 0, t2 = 1, next, i = 1;

    printf("Enter number of terms: ");
    scanf("%d", &N);

    printf("Fibonacci Series: ");

    do {
        printf("%d ", t1);
        next = t1 + t2;
        t1 = t2;
        t2 = next;
        i++;
    } while(i <= N);

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

```
Enter number of terms: 10
Fibonacci Series: 0 1 1 2 3 5 8 13 21 34

Process returned 0 (0x0)   execution time : 3.992 s
Press ENTER to continue.
```

Write a program to check if a number is prime.

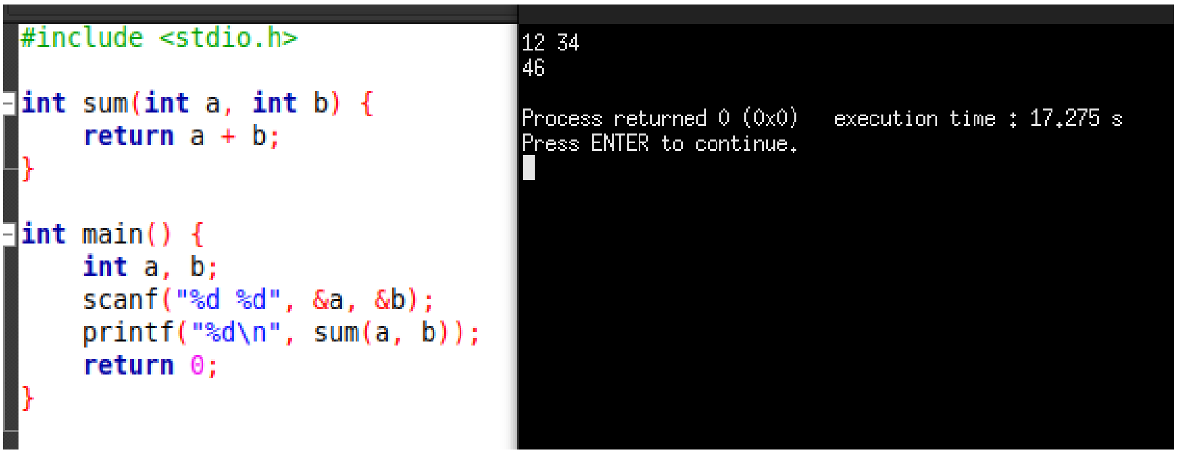


```c
#include <stdio.h>

int main() {
    int num, i;

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

    if(num <= 1) {
        printf("%d is not a prime number.\n", num);
    } else {
        for(i = 2; i < num; i++) {
            if(num % i == 0) {
                printf("%d is not a prime number.\n", num);
                return 0;
            }
        }
        printf("%d is a prime number.\n", num);
    }

    return 0;
}
```

```
Enter a number: 15
15 is not a prime number.

Process returned 0 (0x0)   execution time : 5.556 s
Press ENTER to continue.
```

5. Functions

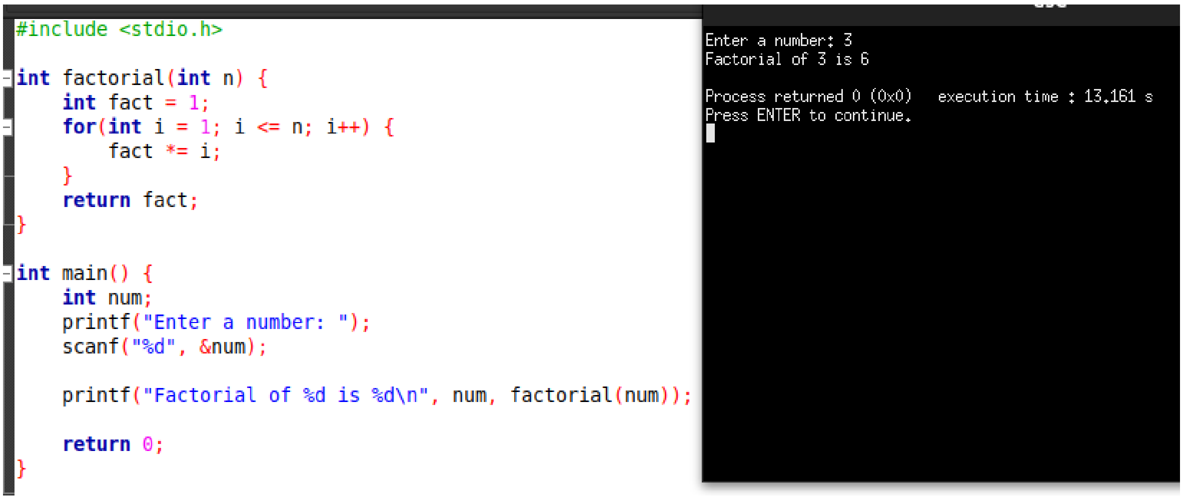
Write a function to return the sum of two numbers.



```c
#include <stdio.h>

int sum(int a, int b) {
    return a + b;
}

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", sum(a, b));
    return 0;
}
```

```
12 34
46

Process returned 0 (0x0)   execution time : 17.275 s
Press ENTER to continue.
```

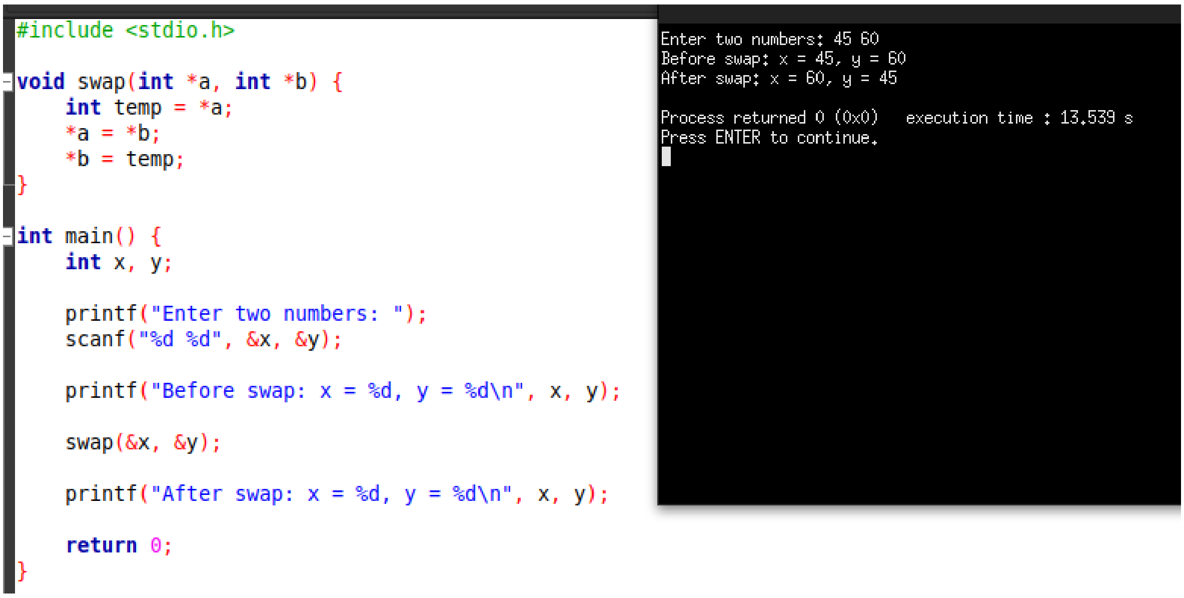
Write a function to find the factorial of a number.



```c
#include <stdio.h>

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

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

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

    return 0;
}
```

```
Enter a number: 3
Factorial of 3 is 6

Process returned 0 (0x0)   execution time : 13.161 s
Press ENTER to continue.
```

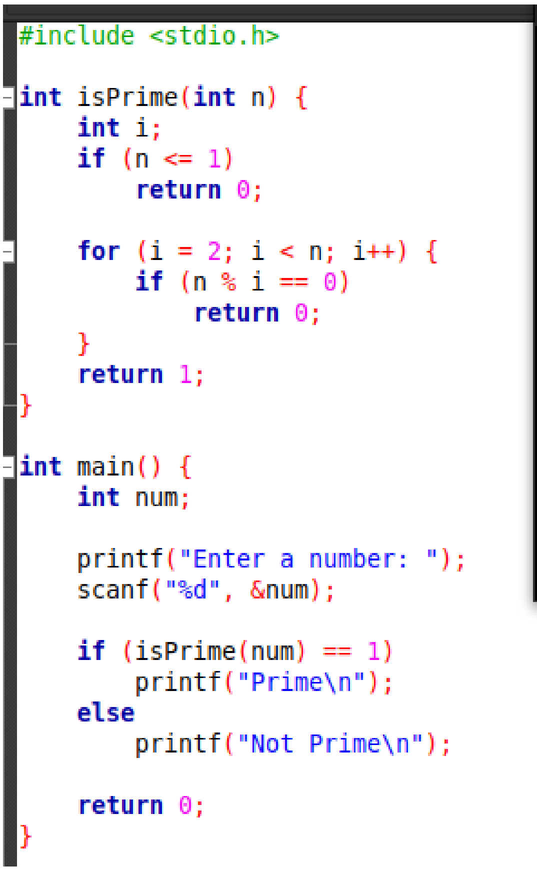
Create a function to swap two numbers using call by reference.



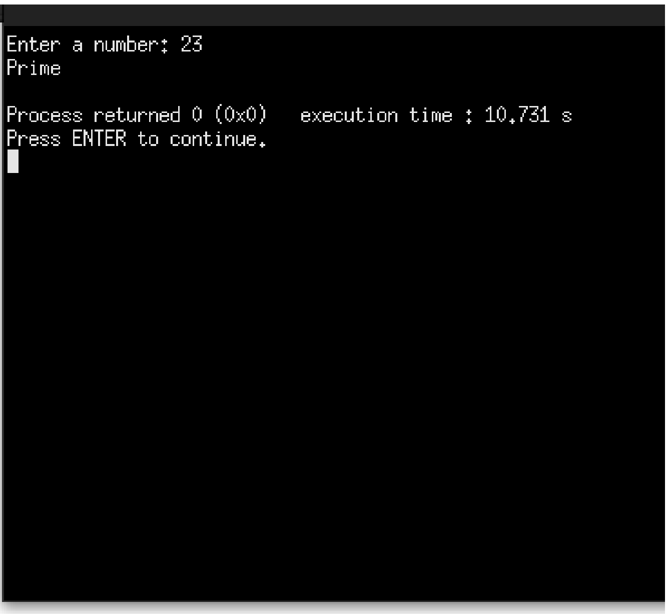
```c
#include <stdio.h>

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 swap: x = %d, y = %d\n", x, y);

    swap(&x, &y);

    printf("After swap: x = %d, y = %d\n", x, y);

    return 0;
}
```

```
Enter two numbers: 45 60
Before swap: x = 45, y = 60
After swap: x = 60, y = 45

Process returned 0 (0x0)   execution time : 13.539 s
Press ENTER to continue.
```

Write a function to check if a number is prime.

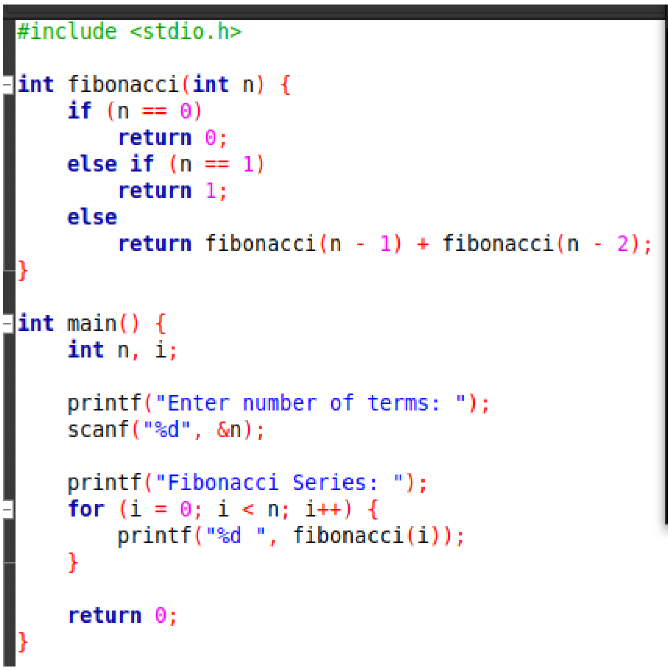


```c
#include <stdio.h>

int isPrime(int n) {
    int i;
    if (n <= 1)
        return 0;

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

int main() {
    int num;

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

    if (isPrime(num) == 1)
        printf("Prime\n");
    else
        printf("Not Prime\n");

    return 0;
}
```

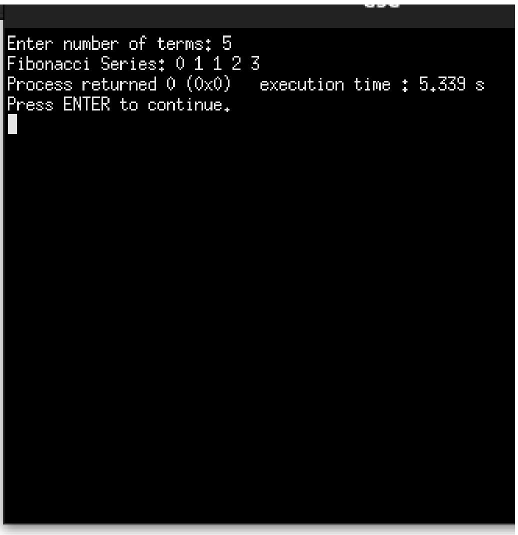


```
Enter a number: 23
Prime

Process returned 0 (0x0)   execution time : 10.731 s
Press ENTER to continue.
```

Write a recursive function to calculate the Fibonacci series.

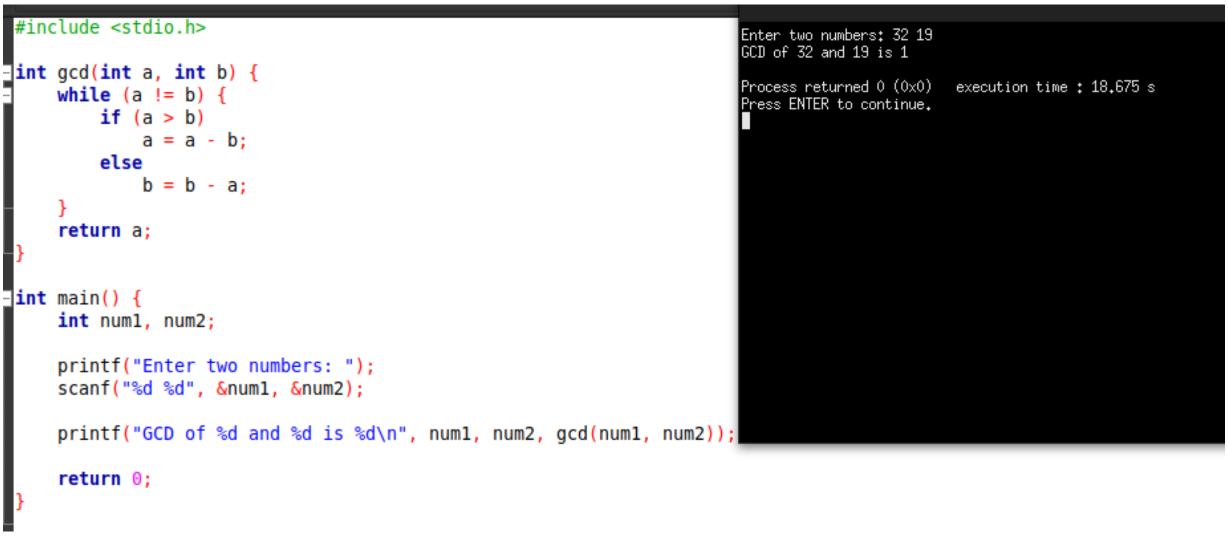


```c
#include <stdio.h>

int fibonacci(int n) {
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return fibonacci(n - 1) + fibonacci(n - 2);
}

int main() {
    int n, i;

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

    printf("Fibonacci Series: ");
    for (i = 0; i < n; i++) {
        printf("%d ", fibonacci(i));
    }

    return 0;
}
```



```
Enter number of terms: 5
Fibonacci Series: 0 1 1 2 3
Process returned 0 (0x0)   execution time : 5.339 s
Press ENTER to continue.
```

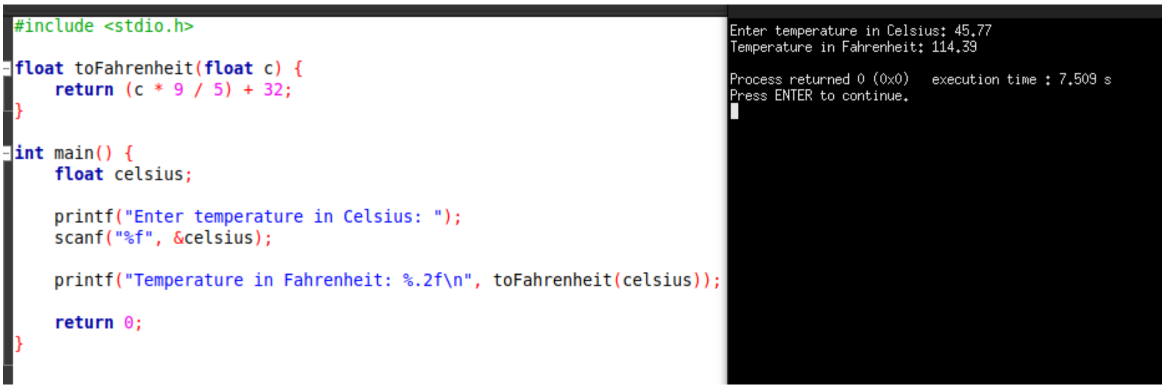
Write a function to calculate the greatest common divisor (GCD) of two numbers.



```c
#include <stdio.h>

int gcd(int a, int b) {
    while (a != b) {
        if (a > b)
            a = a - b;
        else
            b = b - a;
    }
    return a;
}

int main() {
    int num1, num2;

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

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

    return 0;
}
```

```
Enter two numbers: 32 19
GCD of 32 and 19 is 1

Process returned 0 (0x0)   execution time : 18.675 s
Press ENTER to continue.
```

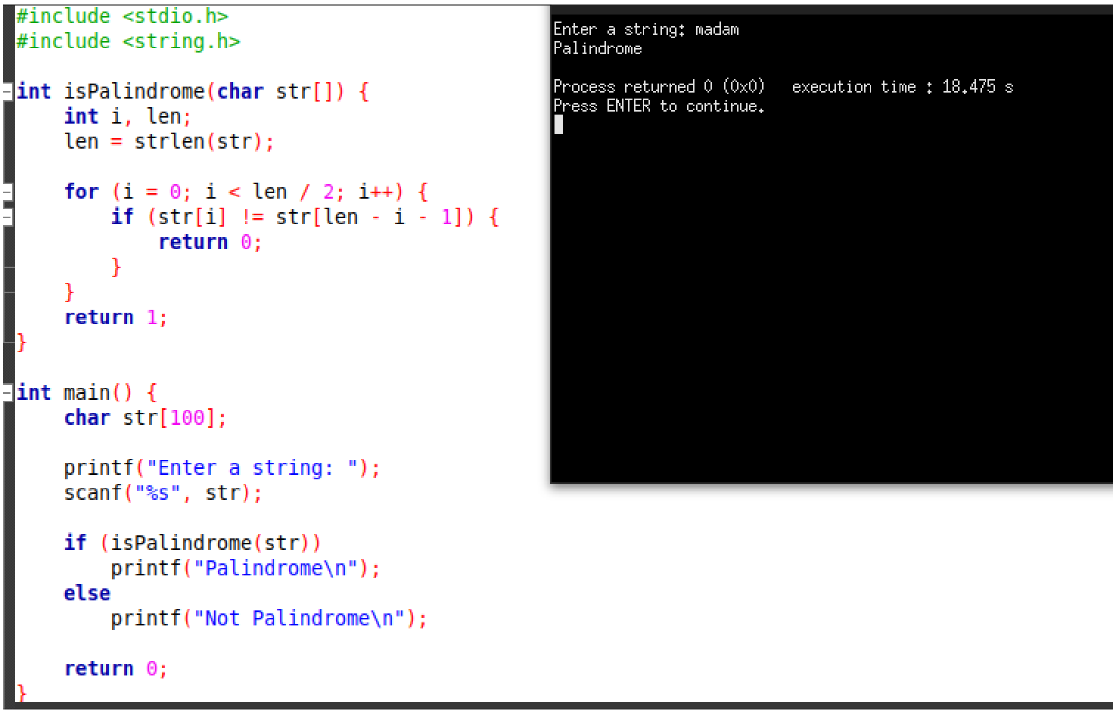
Write a function to convert Celsius to Fahrenheit.



```c
#include <stdio.h>

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

int main() {
    float celsius;

    printf("Enter temperature in Celsius: ");
    scanf("%f", &celsius);

    printf("Temperature in Fahrenheit: %.2f\n", toFahrenheit(celsius));

    return 0;
}
```

```
Enter temperature in Celsius: 45.77
Temperature in Fahrenheit: 114.39

Process returned 0 (0x0)   execution time : 7.509 s
Press ENTER to continue.
```

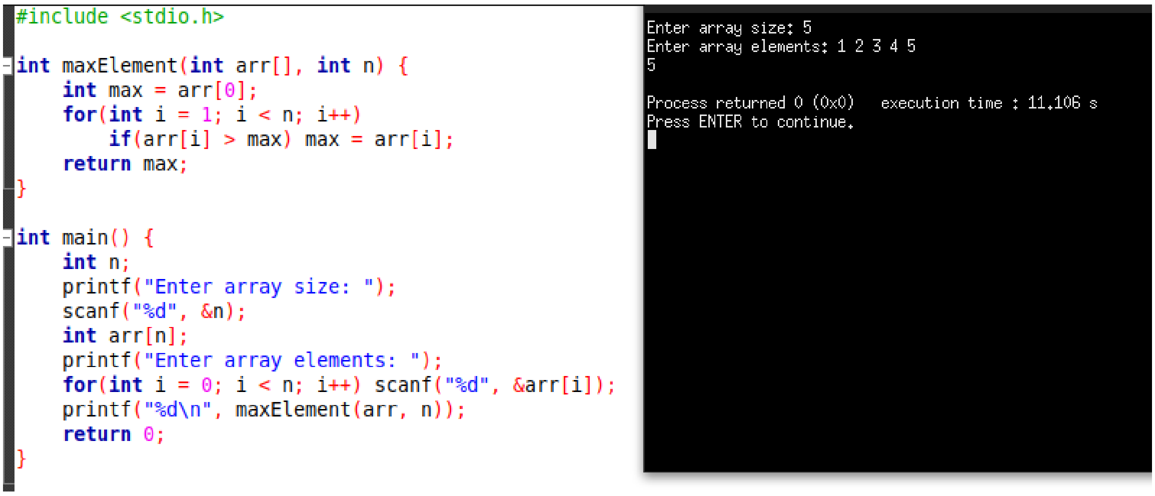
Create a function to check whether a string is a palindrome.



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

int isPalindrome(char str[]) {
    int i, len;
    len = strlen(str);

    for (i = 0; i < len / 2; i++) {
        if (str[i] != str[len - i - 1]) {
            return 0;
        }
    }
    return 1;
}

int main() {
    char str[100];

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

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

    return 0;
}
```

```
Enter a string: madam
Palindrome

Process returned 0 (0x0)   execution time : 18.475 s
Press ENTER to continue.
```

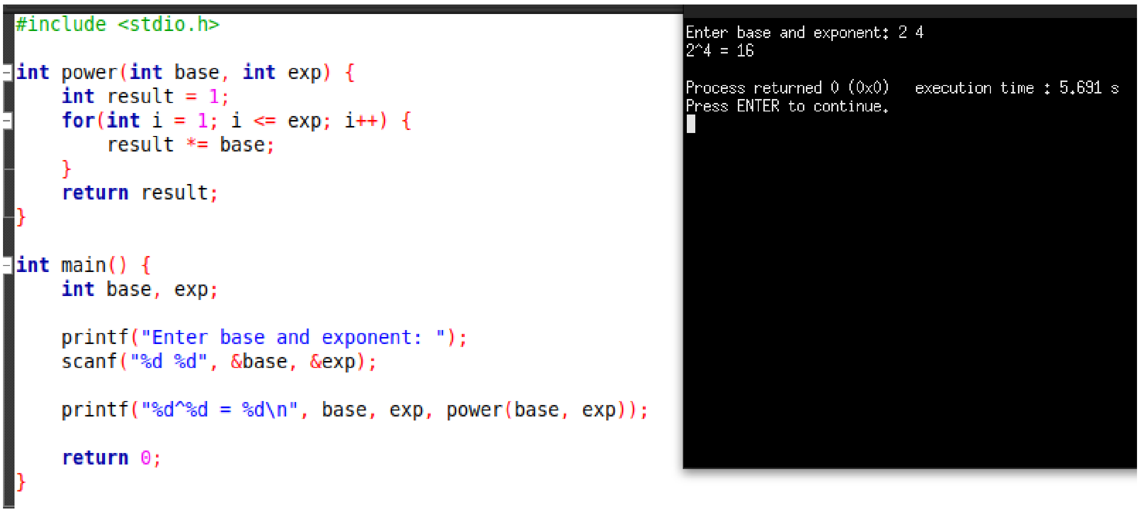
Write a function to return the maximum element in an array.



```c
#include <stdio.h>

int maxElement(int arr[], int n) {
    int max = arr[0];
    for(int i = 1; i < n; i++)
        if(arr[i] > max) max = arr[i];
    return max;
}

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

```
Enter array size: 5
Enter array elements: 1 2 3 4 5
5

Process returned 0 (0x0)   execution time : 11.106 s
Press ENTER to continue.
```

Write a function to calculate the power of a number.

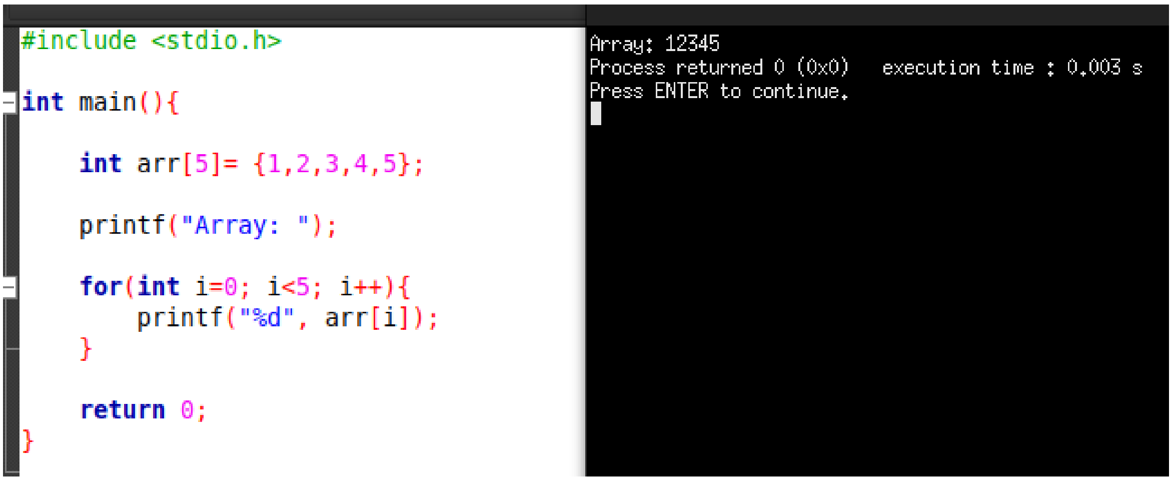


```c
#include <stdio.h>

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

int main() {
    int base, exp;

    printf("Enter base and exponent: ");
    scanf("%d %d", &base, &exp);

    printf("%d^%d = %d\n", base, exp, power(base, exp));

    return 0;
}
```

```
Enter base and exponent: 2 4
2^4 = 16

Process returned 0 (0x0)   execution time : 5.691 s
Press ENTER to continue.
```

6. Arrays and Strings

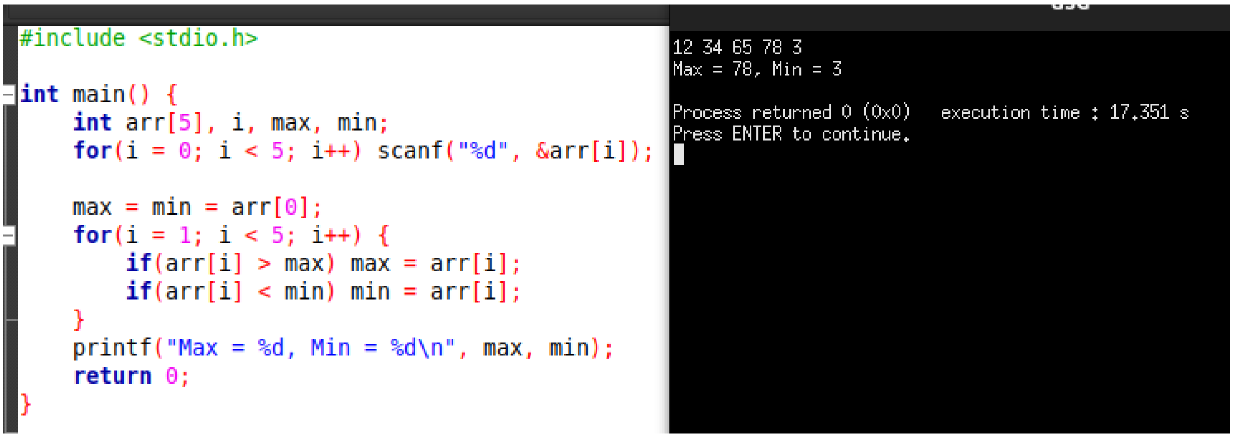
Write a program to take an array of 5 integers and print them.



```c
#include <stdio.h>

int main(){

    int arr[5]= {1,2,3,4,5};

    printf("Array: ");

    for(int i=0; i<5; i++){
        printf("%d", arr[i]);
    }

    return 0;
}
```

```
Array: 12345
Process returned 0 (0x0)   execution time : 0.003 s
Press ENTER to continue.
```

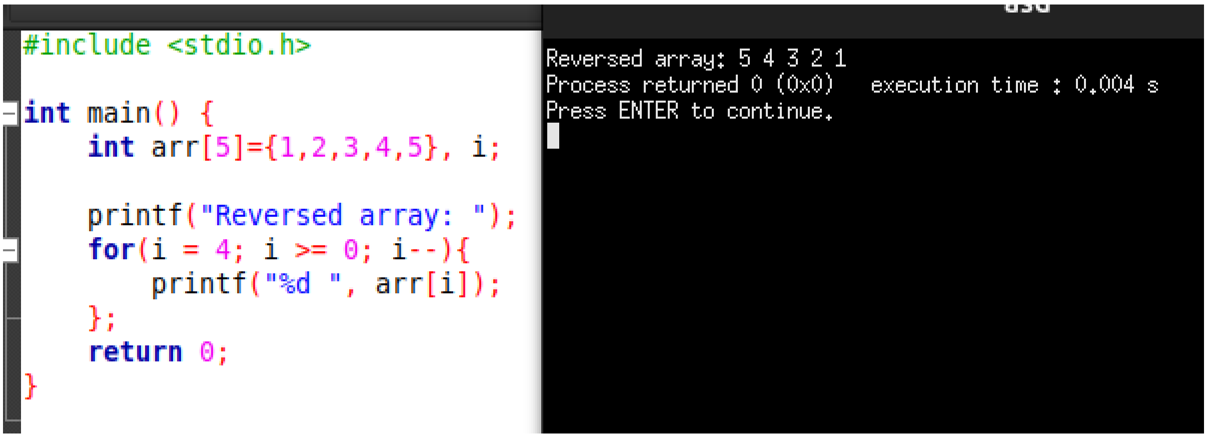
Write a program to find the maximum and minimum element in an array.



```c
#include <stdio.h>

int main() {
    int arr[5], i, max, min;
    for(i = 0; i < 5; i++) scanf("%d", &arr[i]);

    max = min = arr[0];
    for(i = 1; i < 5; i++) {
        if(arr[i] > max) max = arr[i];
        if(arr[i] < min) min = arr[i];
    }
    printf("Max = %d, Min = %d\n", max, min);
    return 0;
}
```

```
12 34 65 78 3
Max = 78, Min = 3

Process returned 0 (0x0)   execution time : 17.351 s
Press ENTER to continue.
```

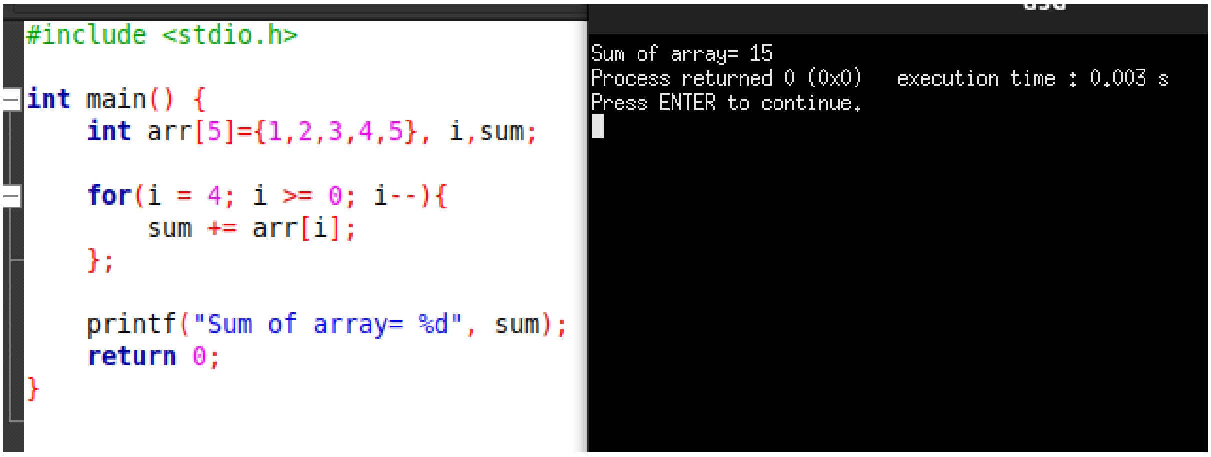
Write a program to reverse an array.



```c
#include <stdio.h>

int main() {
    int arr[5]={1,2,3,4,5}, i;

    printf("Reversed array: ");
    for(i = 4; i >= 0; i--){
        printf("%d ", arr[i]);
    };
    return 0;
}
```

```
Reversed array: 5 4 3 2 1
Process returned 0 (0x0)   execution time : 0.004 s
Press ENTER to continue.
```

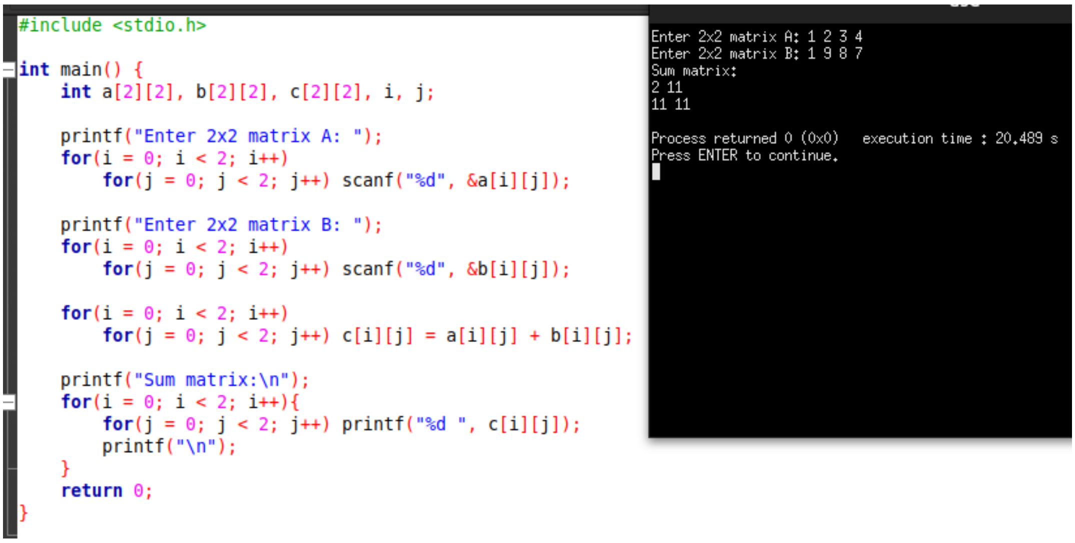
Create a program that finds the sum of all elements in an array.



```c
#include <stdio.h>

int main() {
    int arr[5]={1,2,3,4,5}, i,sum;

    for(i = 4; i >= 0; i--){
        sum += arr[i];
    };

    printf("Sum of array= %d", sum);
    return 0;
}
```

```
Sum of array= 15
Process returned 0 (0x0)   execution time : 0.003 s
Press ENTER to continue.
```

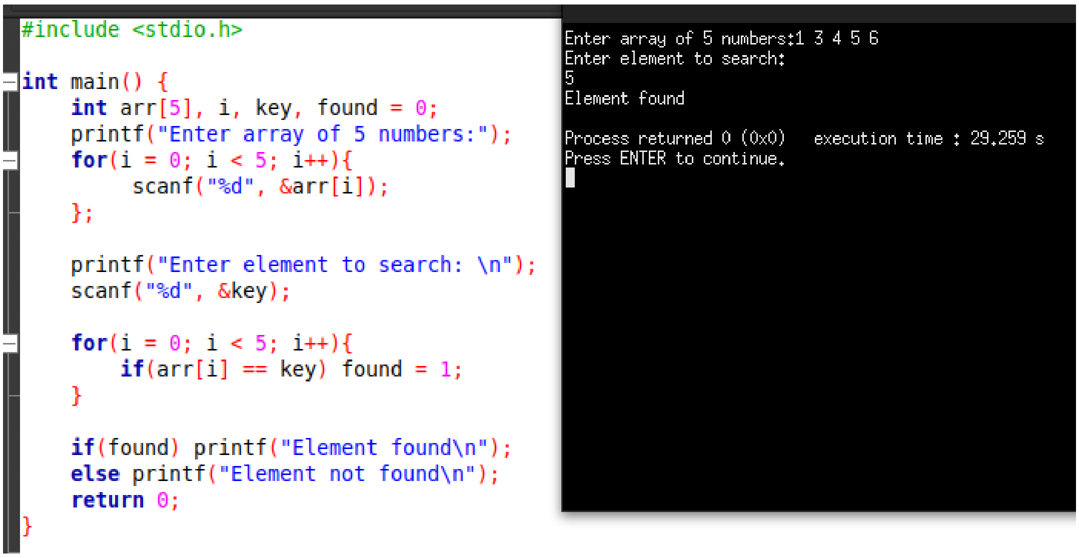
Write a program to perform matrix addition.



```c
#include <stdio.h>

int main() {
    int a[2][2], b[2][2], c[2][2], i, j;

    printf("Enter 2x2 matrix A: ");
    for(i = 0; i < 2; i++)
        for(j = 0; j < 2; j++) scanf("%d", &a[i][j]);

    printf("Enter 2x2 matrix B: ");
    for(i = 0; i < 2; i++)
        for(j = 0; j < 2; j++) scanf("%d", &b[i][j]);

    for(i = 0; i < 2; i++)
        for(j = 0; j < 2; j++) c[i][j] = a[i][j] + b[i][j];

    printf("Sum matrix:\n");
    for(i = 0; i < 2; i++){
        for(j = 0; j < 2; j++) printf("%d ", c[i][j]);
        printf("\n");
    }
    return 0;
}
```

```
Enter 2x2 matrix A: 1 2 3 4
Enter 2x2 matrix B: 1 9 8 7
Sum matrix:
2 11
11 11

Process returned 0 (0x0)   execution time : 20.489 s
Press ENTER to continue.
```

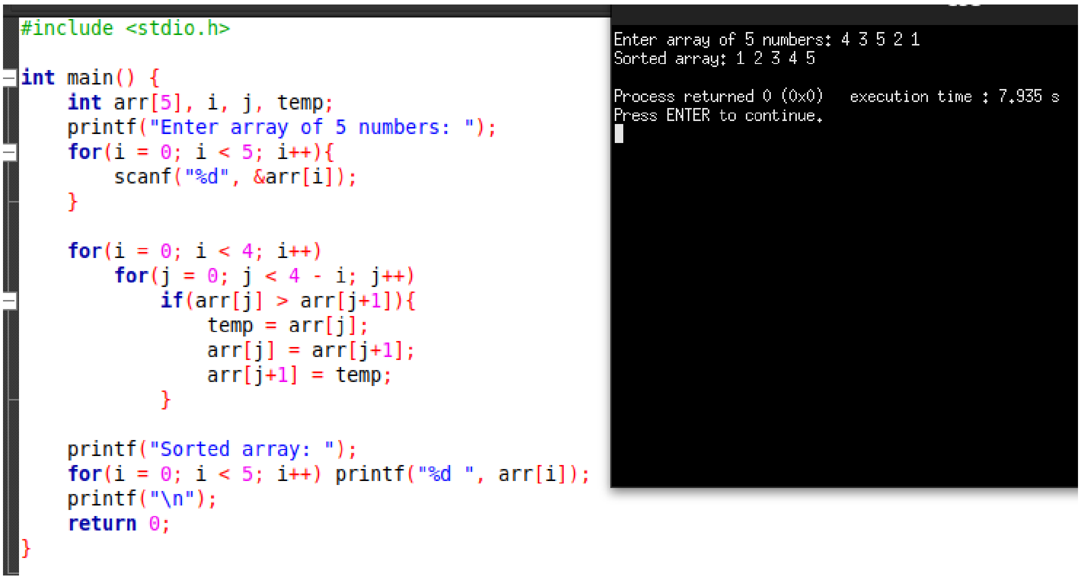
Write a program to search for an element in an array.



```c
#include <stdio.h>

int main() {
    int arr[5], i, key, found = 0;
    printf("Enter array of 5 numbers:");
    for(i = 0; i < 5; i++){
          scanf("%d", &arr[i]);
    };

    printf("Enter element to search: \n");
    scanf("%d", &key);

    for(i = 0; i < 5; i++){
        if(arr[i] == key) found = 1;
    }

    if(found) printf("Element found\n");
    else printf("Element not found\n");
    return 0;
}
```

```
Enter array of 5 numbers:1 3 4 5 6
Enter element to search:
5
Element found

Process returned 0 (0x0)   execution time : 29.259 s
Press ENTER to continue.
```

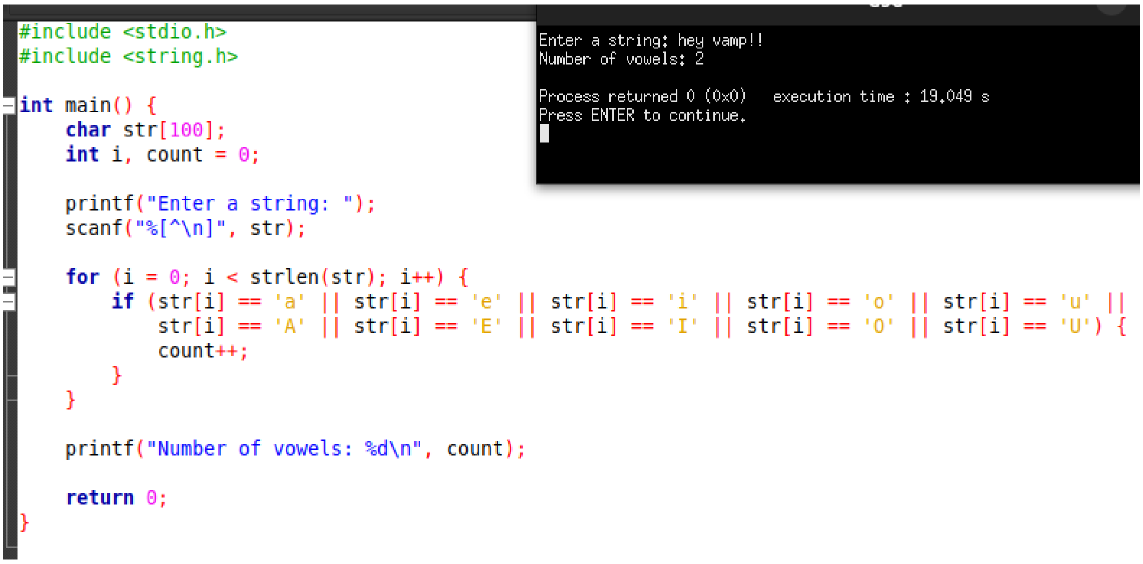
Write a program to sort an array using bubble sort.



```c
#include <stdio.h>

int main() {
    int arr[5], i, j, temp;
    printf("Enter array of 5 numbers: ");
    for(i = 0; i < 5; i++){
        scanf("%d", &arr[i]);
    }

    for(i = 0; i < 4; i++)
        for(j = 0; j < 4 - i; j++)
            if(arr[j] > arr[j+1]){
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }

    printf("Sorted array: ");
    for(i = 0; i < 5; i++) printf("%d ", arr[i]);
    printf("\n");
    return 0;
}
```

```
Enter array of 5 numbers: 4 3 5 2 1
Sorted array: 1 2 3 4 5

Process returned 0 (0x0)   execution time : 7.935 s
Press ENTER to continue.
```

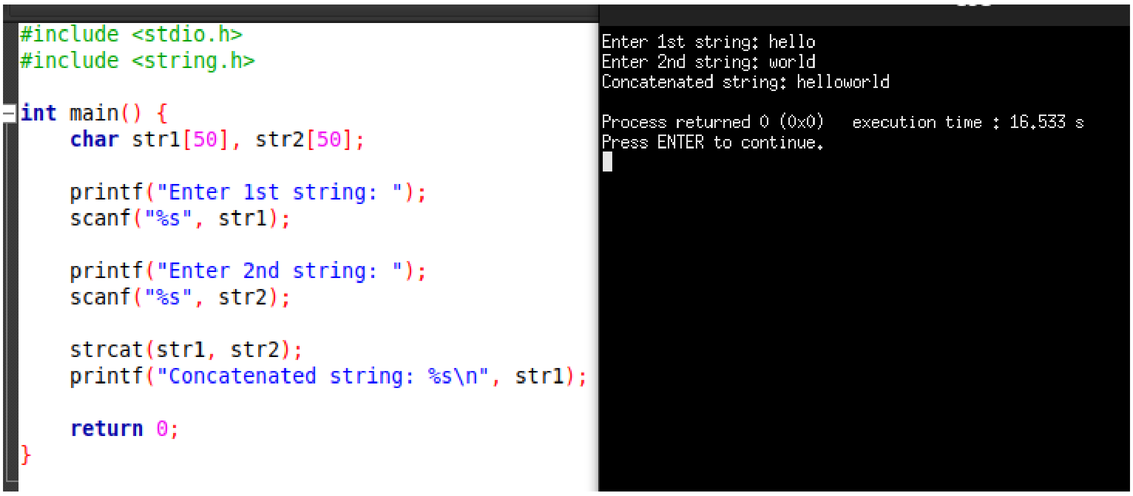
Create a program that counts the number of vowels in a string.



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

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

    printf("Enter a string: ");
    scanf("%[^\n]", str);

    for (i = 0; i < strlen(str); 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++;
        }
    }

    printf("Number of vowels: %d\n", count);

    return 0;
}
```

```
Enter a string: hey vamp!!
Number of vowels: 2

Process returned 0 (0x0)   execution time : 19.049 s
Press ENTER to continue.
```

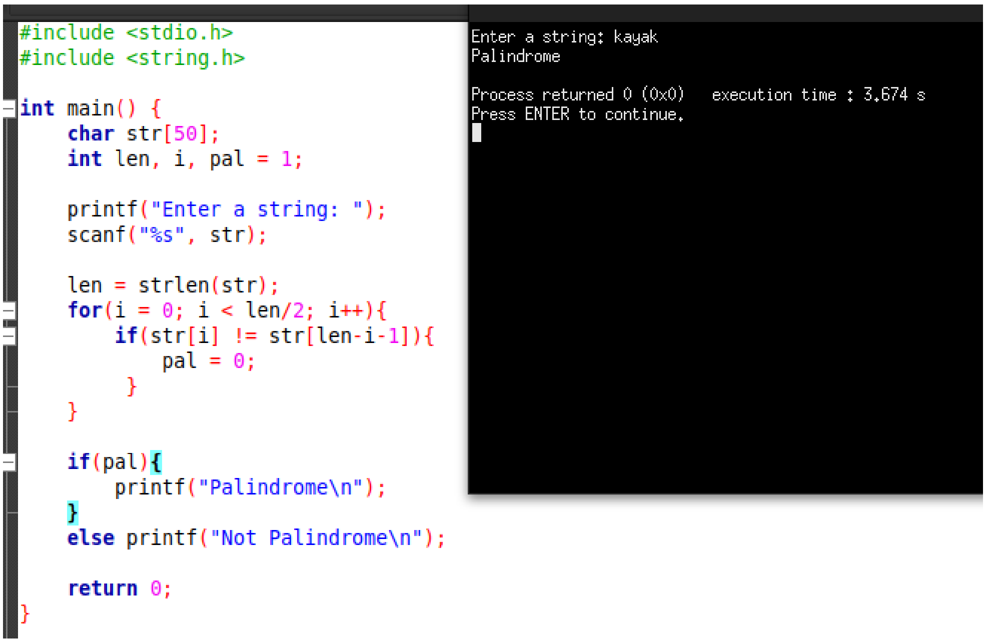
Write a program to concatenate two strings.



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

int main() {
    char str1[50], str2[50];

    printf("Enter 1st string: ");
    scanf("%s", str1);

    printf("Enter 2nd string: ");
    scanf("%s", str2);

    strcat(str1, str2);
    printf("Concatenated string: %s\n", str1);

    return 0;
}
```

```
Enter 1st string: hello
Enter 2nd string: world
Concatenated string: helloworld

Process returned 0 (0x0)   execution time : 16.533 s
Press ENTER to continue.
```

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

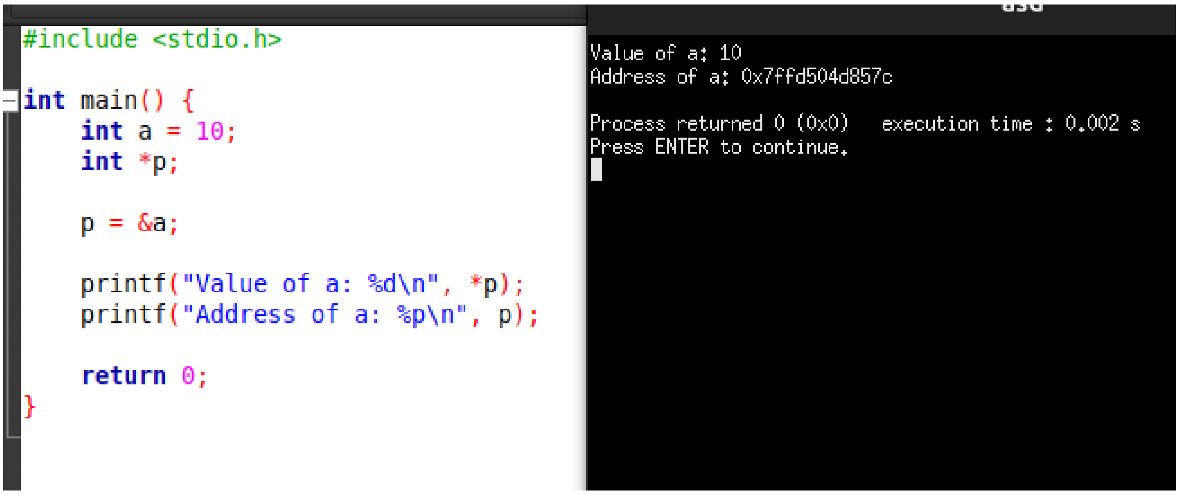


```c
#include <stdio.h>
#include <string.h>

int main() {
    char str[50];
    int len, i, pal = 1;

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

    len = strlen(str);
    for(i = 0; i < len/2; i++){
        if(str[i] != str[len-i-1]){
            pal = 0;
         }
    }

    if(pal){
        printf("Palindrome\n");
    }
    else printf("Not Palindrome\n");

    return 0;
}
```

```
Enter a string: kayak
Palindrome

Process returned 0 (0x0)   execution time : 3.674 s
Press ENTER to continue.
```

7. Pointers

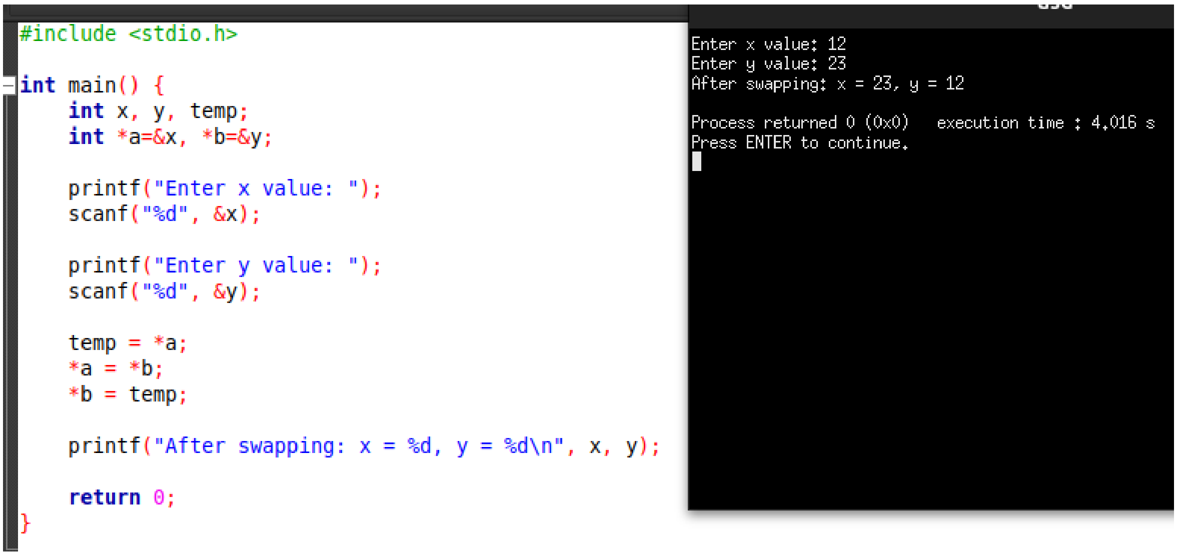
Write a program to declare a pointer and print its address.



```c
#include <stdio.h>

int main() {
    int a = 10;
    int *p;

    p = &a;

    printf("Value of a: %d\n", *p);
    printf("Address of a: %p\n", p);

    return 0;
}
```

```
Value of a: 10
Address of a: 0x7ffd504d857c

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

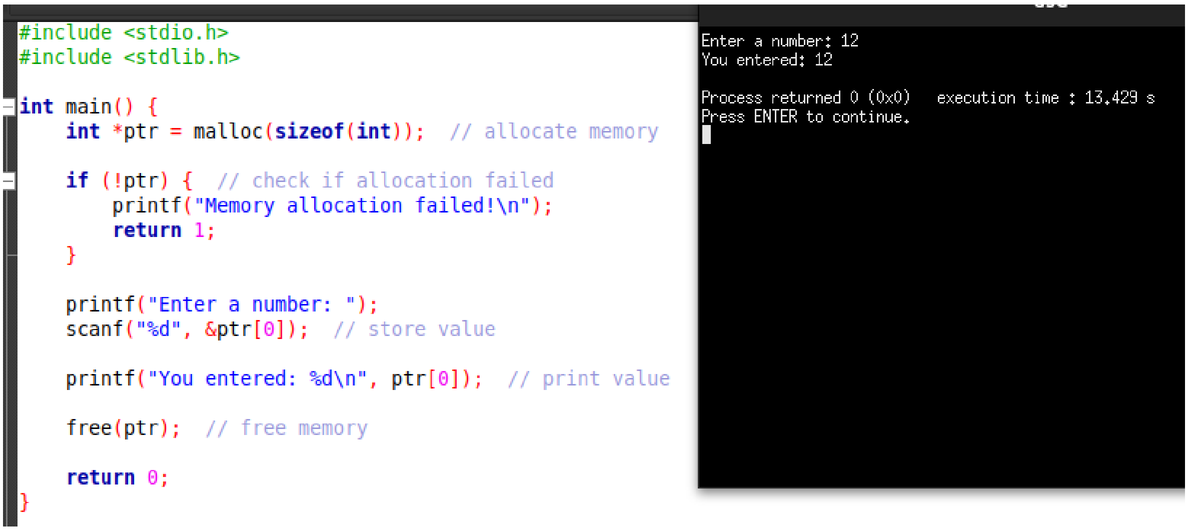
Write a program to swap two numbers using pointers.



```c
#include <stdio.h>

int main() {
    int x, y, temp;
    int *a=&x, *b=&y;

    printf("Enter x value: ");
    scanf("%d", &x);

    printf("Enter y value: ");
    scanf("%d", &y);

    temp = *a;
    *a = *b;
    *b = temp;

    printf("After swapping: x = %d, y = %d\n", x, y);

    return 0;
}
```

```
Enter x value: 12
Enter y value: 23
After swapping: x = 23, y = 12

Process returned 0 (0x0)   execution time : 4.016 s
Press ENTER to continue.
```

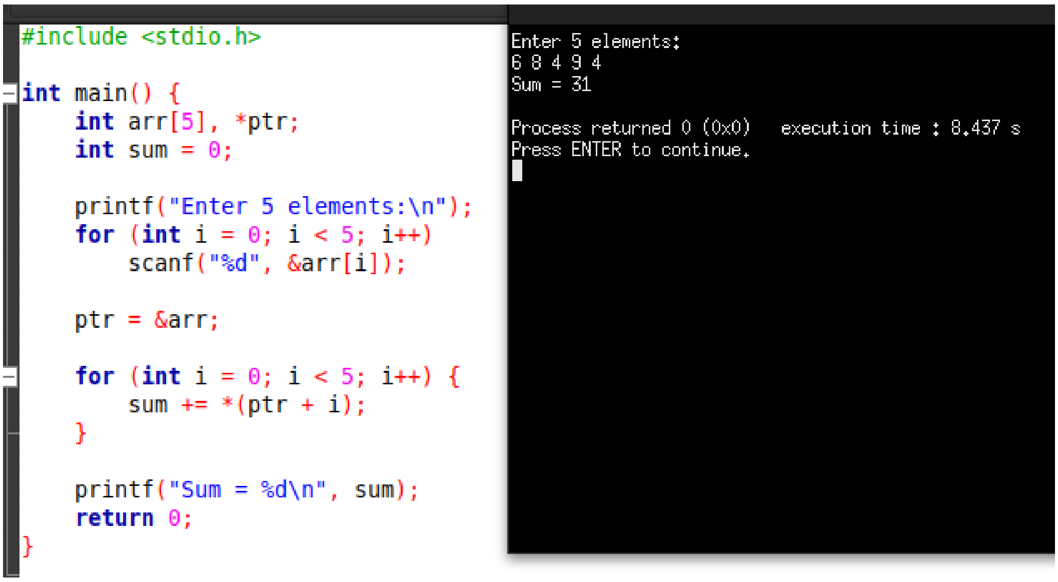
Create a program that dynamically allocates an integer using new.



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

int main() {
    int *ptr = malloc(sizeof(int));  // allocate memory

    if (!ptr) {  // check if allocation failed
        printf("Memory allocation failed!\n");
        return 1;
    }

    printf("Enter a number: ");
    scanf("%d", &ptr[0]);  // store value

    printf("You entered: %d\n", ptr[0]);  // print value

    free(ptr);  // free memory

    return 0;
}
```

```
Enter a number: 12
You entered: 12

Process returned 0 (0x0)   execution time : 13.429 s
Press ENTER to continue.
```

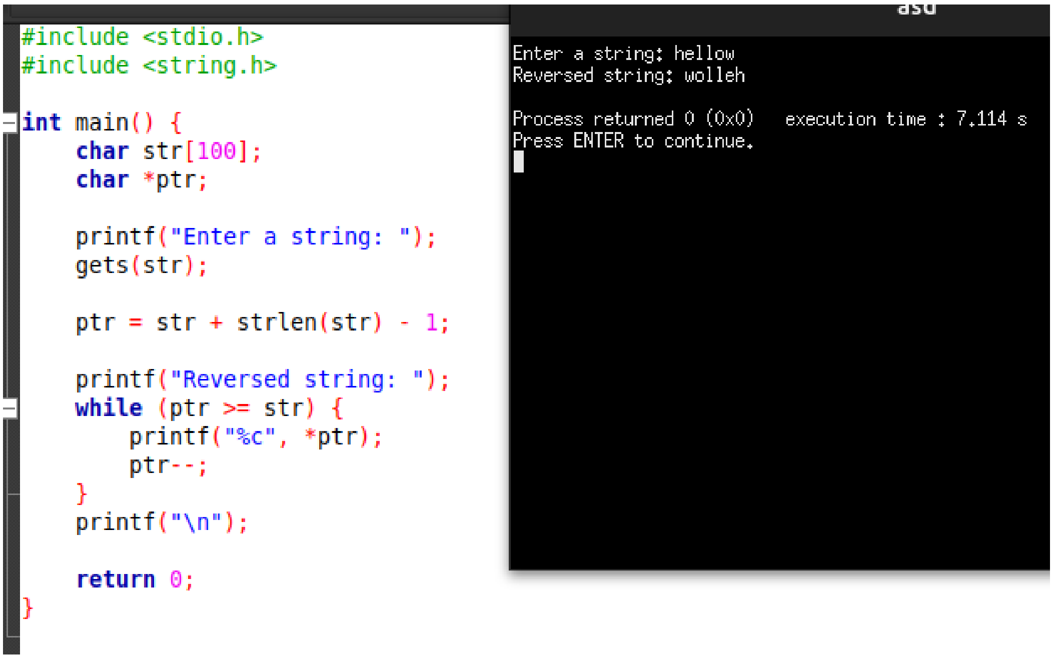
Write a program to find the sum of array elements using pointers.



```c
#include <stdio.h>

int main() {
    int arr[5], *ptr;
    int sum = 0;

    printf("Enter 5 elements:\n");
    for (int i = 0; i < 5; i++)
        scanf("%d", &arr[i]);

    ptr = &arr;

    for (int i = 0; i < 5; i++) {
        sum += *(ptr + i);
    }

    printf("Sum = %d\n", sum);
    return 0;
}
```

```
Enter 5 elements:
6 8 4 9 4
Sum = 31

Process returned 0 (0x0)   execution time : 8.437 s
Press ENTER to continue.
```

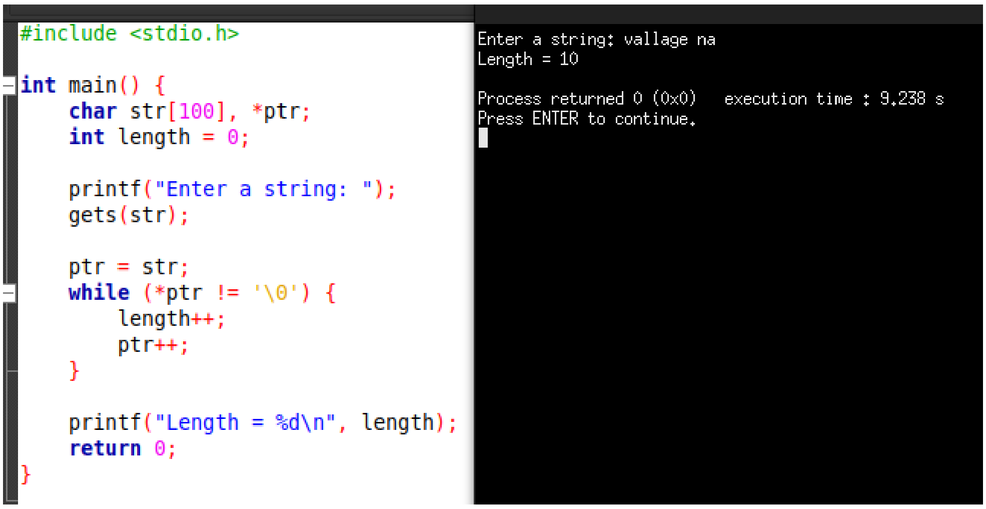
Write a program to reverse a string using pointers.



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

int main() {
    char str[100];
    char *ptr;

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

    ptr = str + strlen(str) - 1;

    printf("Reversed string: ");
    while (ptr >= str) {
        printf("%c", *ptr);
        ptr--;
    }
    printf("\n");

    return 0;
}
```

```
Enter a string: hellow
Reversed string: wolleh

Process returned 0 (0x0)   execution time : 7.114 s
Press ENTER to continue.
```

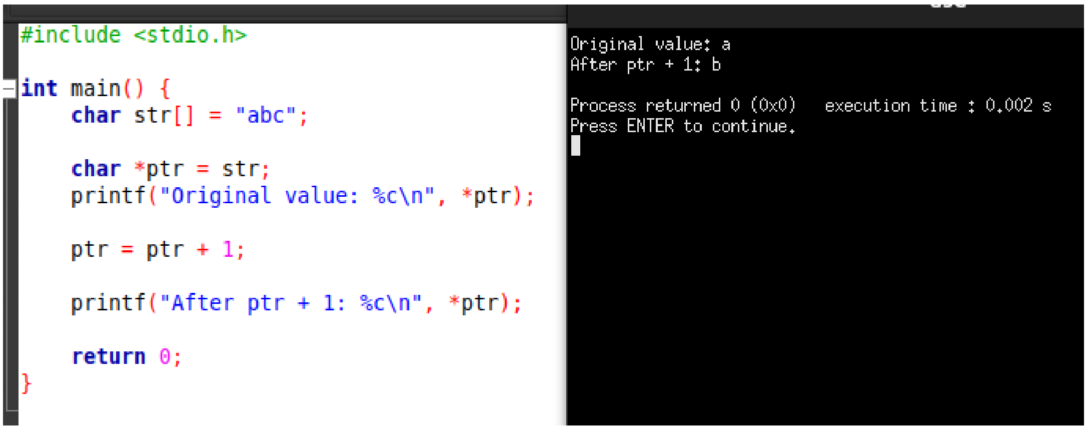
Write a program to find the length of a string using pointers.



```c
#include <stdio.h>

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

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

    ptr = str;
    while (*ptr != '\0') {
        length++;
        ptr++;
    }

    printf("Length = %d\n", length);
    return 0;
}
```

```
Enter a string: vallage na
Length = 10

Process returned 0 (0x0)   execution time : 9.238 s
Press ENTER to continue.
```

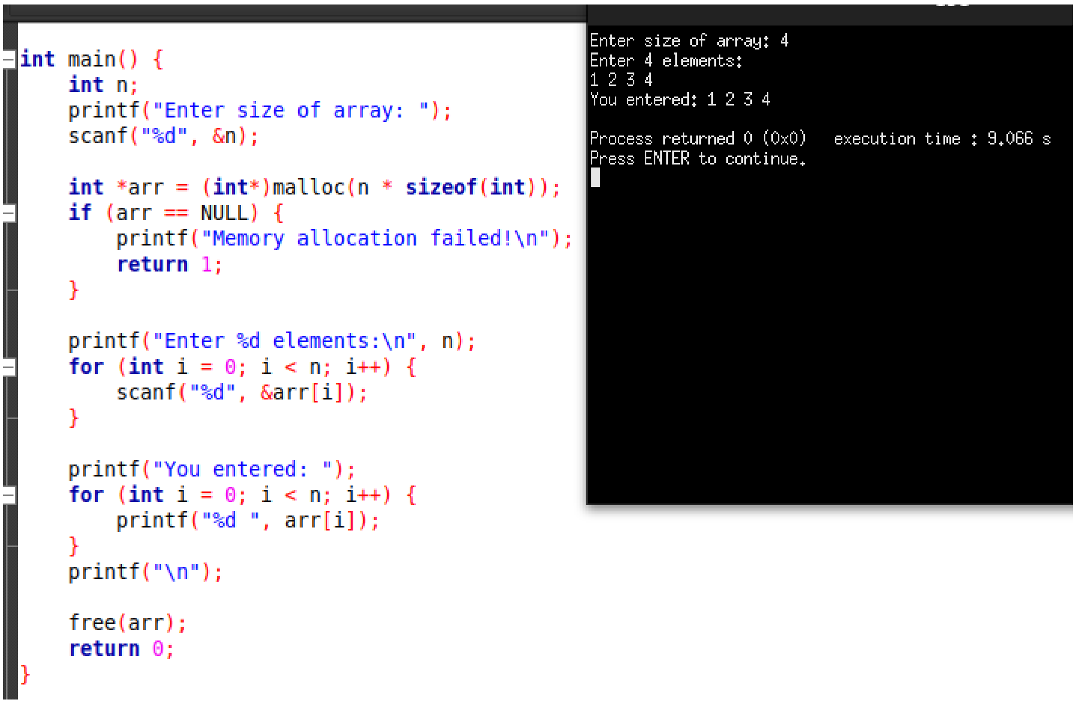
Write a program to demonstrate pointer arithmetic.



```c
#include <stdio.h>

int main() {
    char str[] = "abc";

    char *ptr = str;
    printf("Original value: %c\n", *ptr);

    ptr = ptr + 1;

    printf("After ptr + 1: %c\n", *ptr);

    return 0;
}
```

```
Original value: a
After ptr + 1: b

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

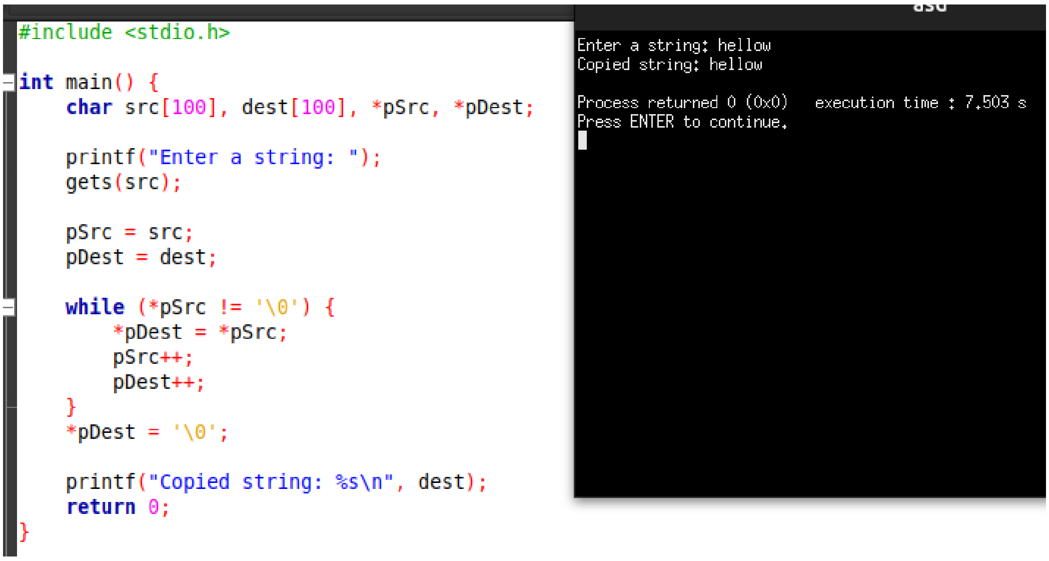
Write a program to create and delete a dynamic array using new and delete.



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

    int *arr = (int*)malloc(n * sizeof(int));
    if (arr == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }

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

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

    free(arr);
    return 0;
}
```

```
Enter size of array: 4
Enter 4 elements:
1 2 3 4
You entered: 1 2 3 4

Process returned 0 (0x0)   execution time : 9.066 s
Press ENTER to continue.
```

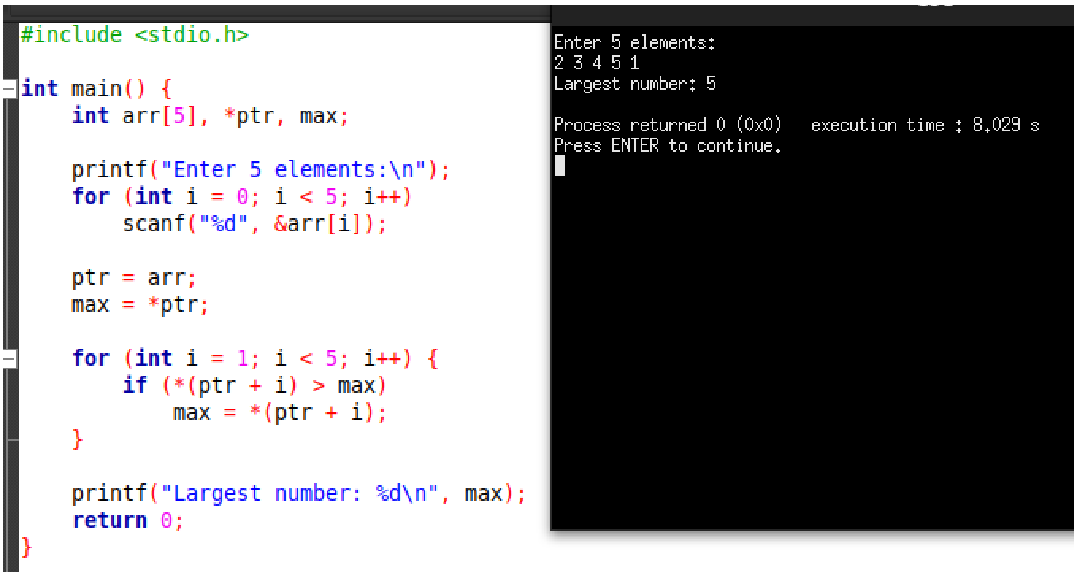
Write a program to copy a string using pointers.



```c
#include <stdio.h>

int main() {
    char src[100], dest[100], *pSrc, *pDest;

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

    pSrc = src;
    pDest = dest;

    while (*pSrc != '\0') {
        *pDest = *pSrc;
        pSrc++;
        pDest++;
    }
    *pDest = '\0';

    printf("Copied string: %s\n", dest);
    return 0;
}
```

```
Enter a string: hellow
Copied string: hellow

Process returned 0 (0x0)   execution time : 7.503 s
Press ENTER to continue.
```

Write a program to find the largest number in an array using pointers.

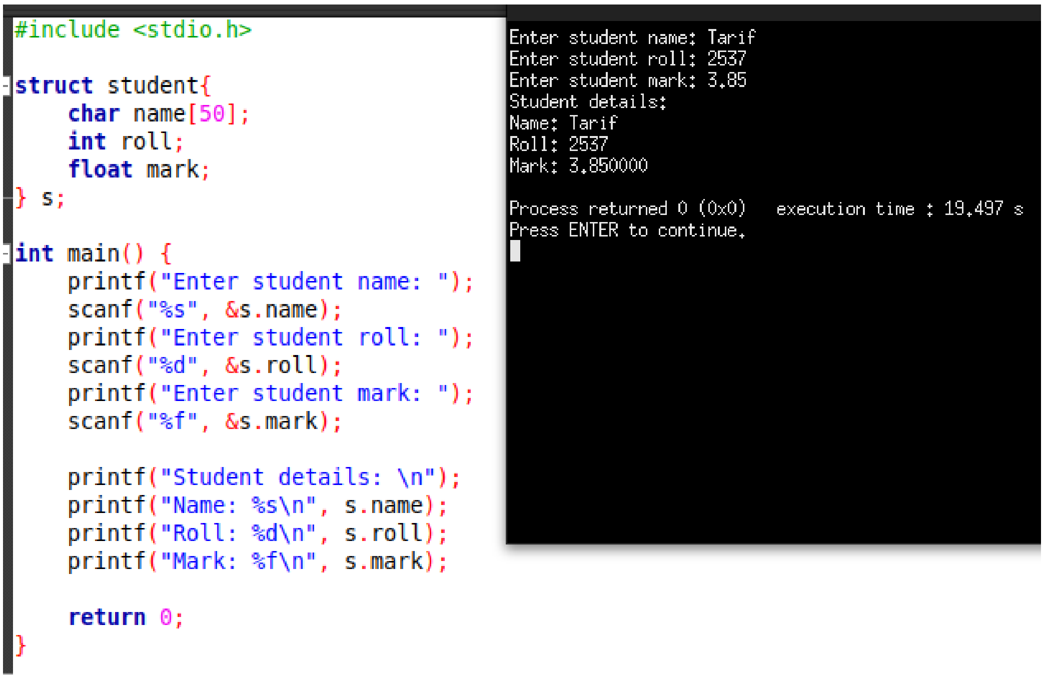


```
#include <stdio.h>

int main() {
    int arr[5], *ptr, max;

    printf("Enter 5 elements:\n");
    for (int i = 0; i < 5; i++)
        scanf("%d", &arr[i]);

    ptr = arr;
    max = *ptr;

    for (int i = 1; i < 5; i++) {
        if (*(ptr + i) > max)
            max = *(ptr + i);
    }

    printf("Largest number: %d\n", max);
    return 0;
}
```

```
Enter 5 elements:
2 3 4 5 1
Largest number: 5

Process returned 0 (0x0)   execution time : 8.029 s
Press ENTER to continue.
```

8. Structures and Unions

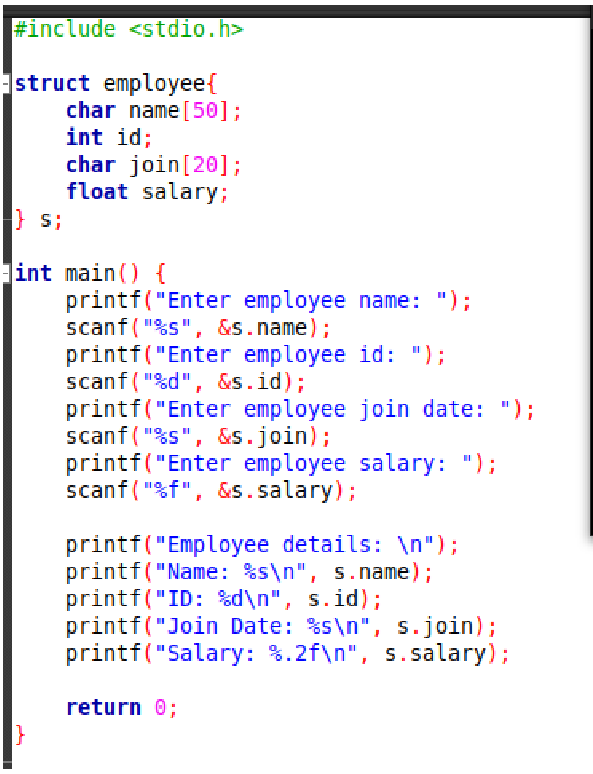
Write a program to define a structure for student details (name, roll, marks) and display them.



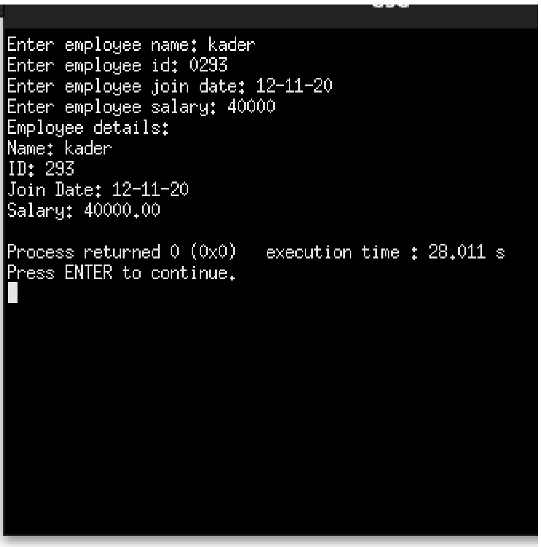
```
#include <stdio.h>

struct student{
    char name[50];
    int roll;
    float mark;
} s;

int main() {
    printf("Enter student name: ");
    scanf("%s", &s.name);
    printf("Enter student roll: ");
    scanf("%d", &s.roll);
    printf("Enter student mark: ");
    scanf("%f", &s.mark);

    printf("Student details: \n");
    printf("Name: %s\n", s.name);
    printf("Roll: %d\n", s.roll);
    printf("Mark: %f\n", s.mark);

    return 0;
}
```

```
Enter student name: Tarif
Enter student roll: 2537
Enter student mark: 3.85
Student details:
Name: Tarif
Roll: 2537
Mark: 3.850000

Process returned 0 (0x0)   execution time : 19.497 s
Press ENTER to continue.
```

Create a structure to store the details of an employee and print them.

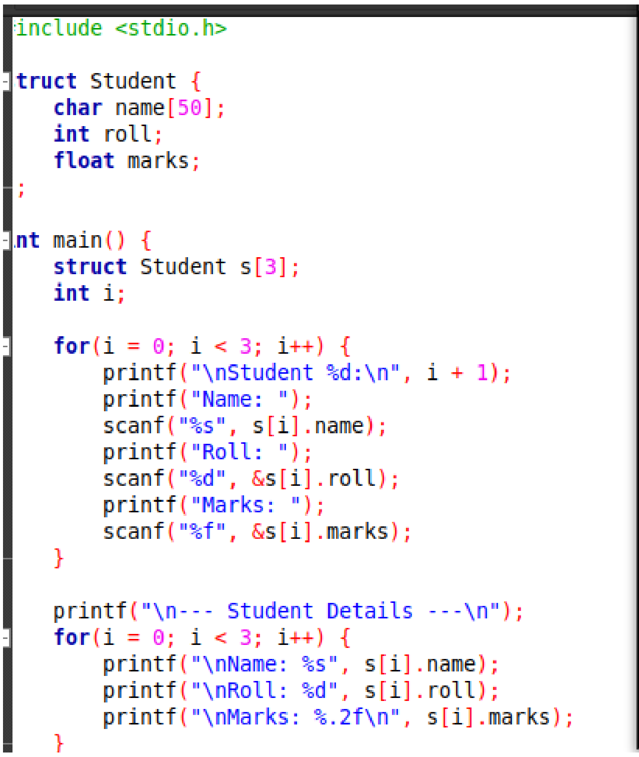


```c
#include <stdio.h>

struct employee{
    char name[50];
    int id;
    char join[20];
    float salary;
} s;

int main() {
    printf("Enter employee name: ");
    scanf("%s", &s.name);
    printf("Enter employee id: ");
    scanf("%d", &s.id);
    printf("Enter employee join date: ");
    scanf("%s", &s.join);
    printf("Enter employee salary: ");
    scanf("%f", &s.salary);

    printf("Employee details: \n");
    printf("Name: %s\n", s.name);
    printf("ID: %d\n", s.id);
    printf("Join Date: %s\n", s.join);
    printf("Salary: %.2f\n", s.salary);

    return 0;
}
```

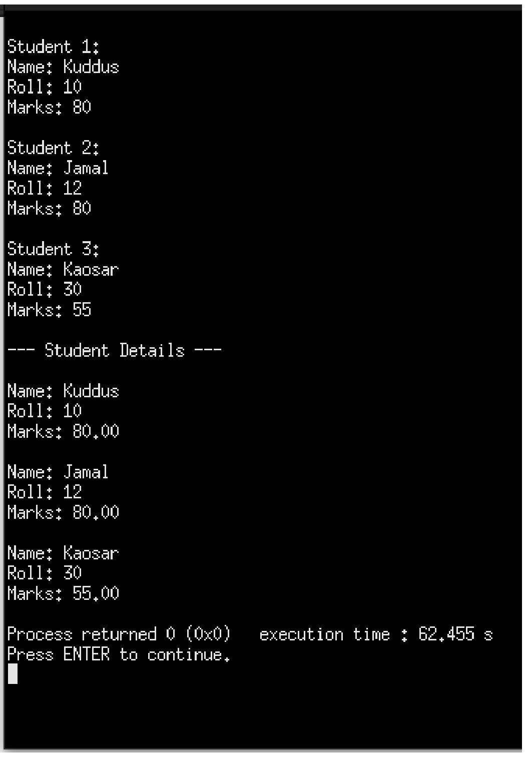


```
Enter employee name: kader
Enter employee id: 0293
Enter employee join date: 12-11-20
Enter employee salary: 40000
Employee details:
Name: kader
ID: 293
Join Date: 12-11-20
Salary: 40000.00

Process returned 0 (0x0)   execution time : 28.011 s
Press ENTER to continue.
```

Write a program to store multiple students’ data using an array of structures.

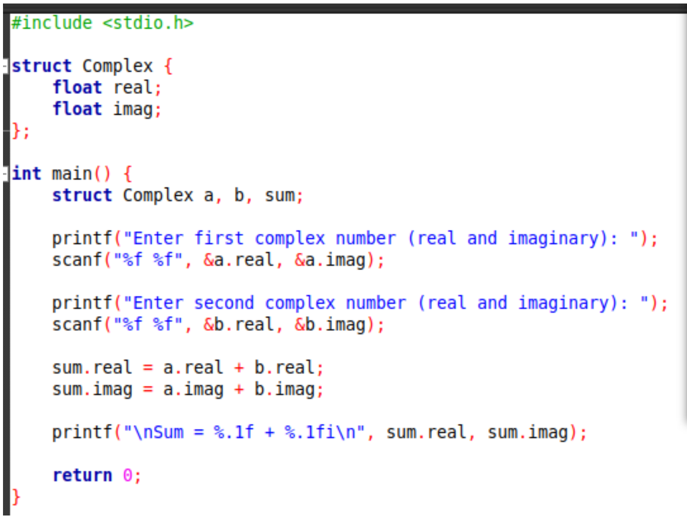


```c
include <stdio.h>

truct Student {
   char name[50];
   int roll;
   float marks;
;

nt main() {
   struct Student s[3];
   int i;

   for(i = 0; i < 3; i++) {
       printf("\nStudent %d:\n", i + 1);
       printf("Name: ");
       scanf("%s", s[i].name);
       printf("Roll: ");
       scanf("%d", &s[i].roll);
       printf("Marks: ");
       scanf("%f", &s[i].marks);
   }

   printf("\n--- Student Details ---\n");
   for(i = 0; i < 3; i++) {
       printf("\nName: %s", s[i].name);
       printf("\nRoll: %d", s[i].roll);
       printf("\nMarks: %.2f\n", s[i].marks);
   }
```

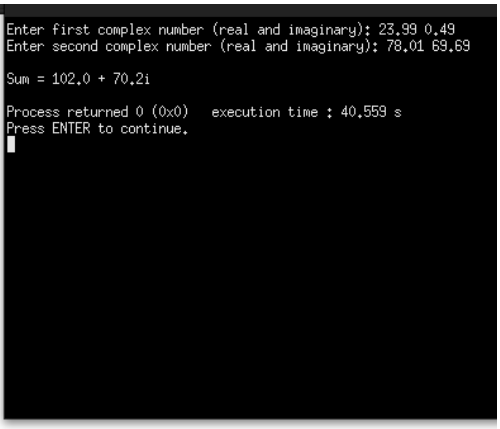


```
Student 1:
Name: Kuddus
Roll: 10
Marks: 80

Student 2:
Name: Jamal
Roll: 12
Marks: 80

Student 3:
Name: Kaosar
Roll: 30
Marks: 55

--- Student Details ---

Name: Kuddus
Roll: 10
Marks: 80.00

Name: Jamal
Roll: 12
Marks: 80.00

Name: Kaosar
Roll: 30
Marks: 55.00

Process returned 0 (0x0)   execution time : 62.455 s
Press ENTER to continue.
```

Create a structure for complex numbers and perform addition.

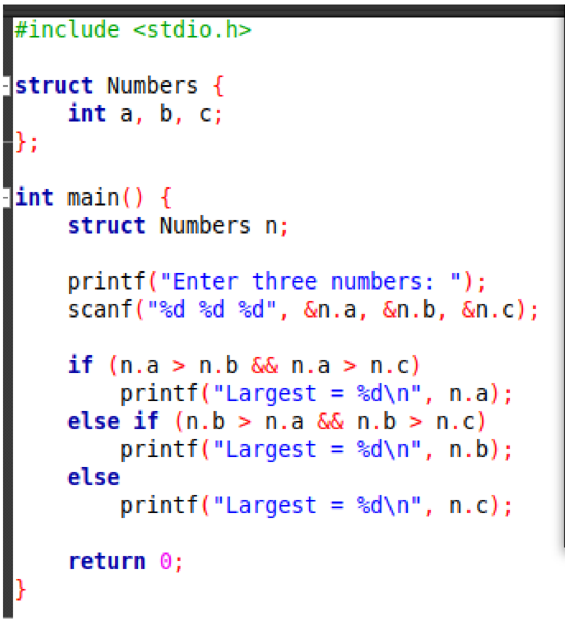


```c
#include <stdio.h>

struct Complex {
    float real;
    float imag;
};

int main() {
    struct Complex a, b, sum;

    printf("Enter first complex number (real and imaginary): ");
    scanf("%f %f", &a.real, &a.imag);

    printf("Enter second complex number (real and imaginary): ");
    scanf("%f %f", &b.real, &b.imag);

    sum.real = a.real + b.real;
    sum.imag = a.imag + b.imag;

    printf("\nSum = %.1f + %.1fi\n", sum.real, sum.imag);

    return 0;
}
```

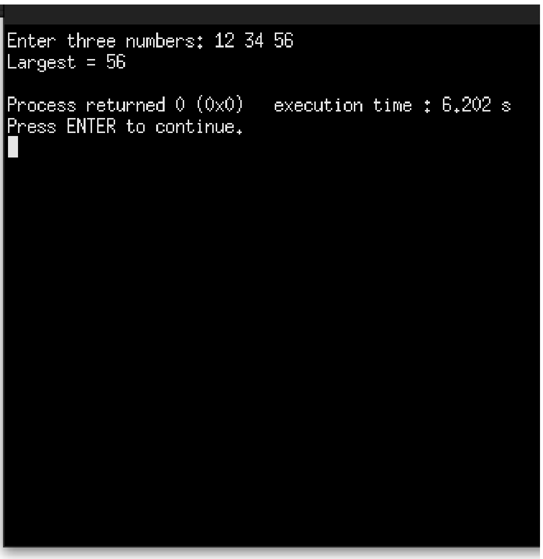


```
Enter first complex number (real and imaginary): 23.99 0.49
Enter second complex number (real and imaginary): 78.01 69.69

Sum = 102.0 + 70.2i

Process returned 0 (0x0)   execution time : 40.559 s
Press ENTER to continue.
```

Write a program to find the largest of three numbers using a structure.

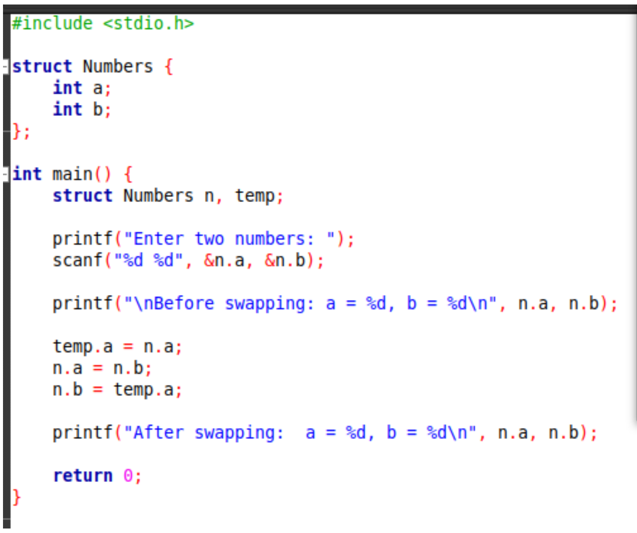


```c
#include <stdio.h>

struct Numbers {
    int a, b, c;
};

int main() {
    struct Numbers n;

    printf("Enter three numbers: ");
    scanf("%d %d %d", &n.a, &n.b, &n.c);

    if (n.a > n.b && n.a > n.c)
        printf("Largest = %d\n", n.a);
    else if (n.b > n.a && n.b > n.c)
        printf("Largest = %d\n", n.b);
    else
        printf("Largest = %d\n", n.c);

    return 0;
}
```

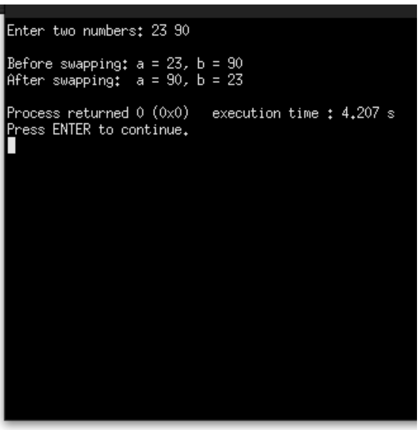


```
Enter three numbers: 12 34 56
Largest = 56

Process returned 0 (0x0)   execution time : 6.202 s
Press ENTER to continue.
```

Write a program to swap two numbers using structures.

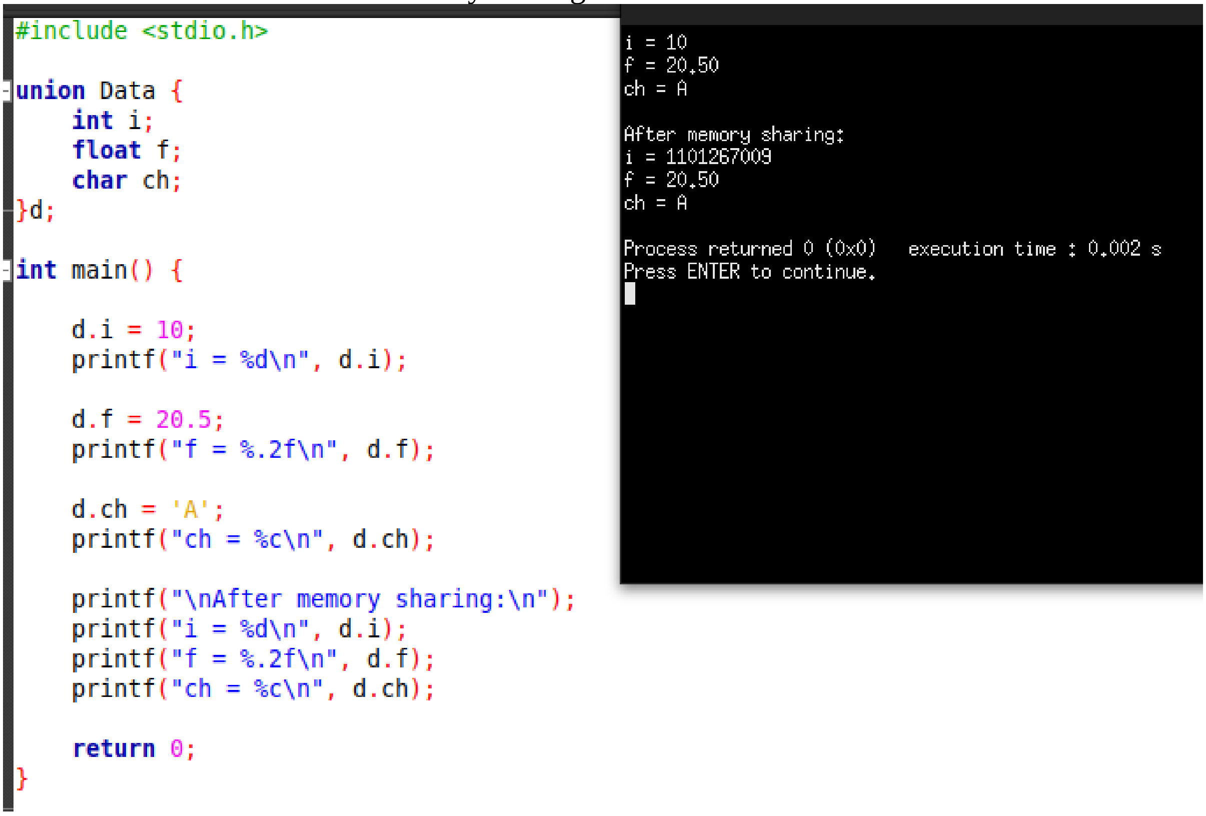


```c
#include <stdio.h>

struct Numbers {
    int a;
    int b;
};

int main() {
    struct Numbers n, temp;

    printf("Enter two numbers: ");
    scanf("%d %d", &n.a, &n.b);

    printf("\nBefore swapping: a = %d, b = %d\n", n.a, n.b);

    temp.a = n.a;
    n.a = n.b;
    n.b = temp.a;

    printf("After swapping:  a = %d, b = %d\n", n.a, n.b);

    return 0;
}
```



```
Enter two numbers: 23 90

Before swapping: a = 23, b = 90
After swapping:  a = 90, b = 23

Process returned 0 (0x0)   execution time : 4.207 s
Press ENTER to continue.
```

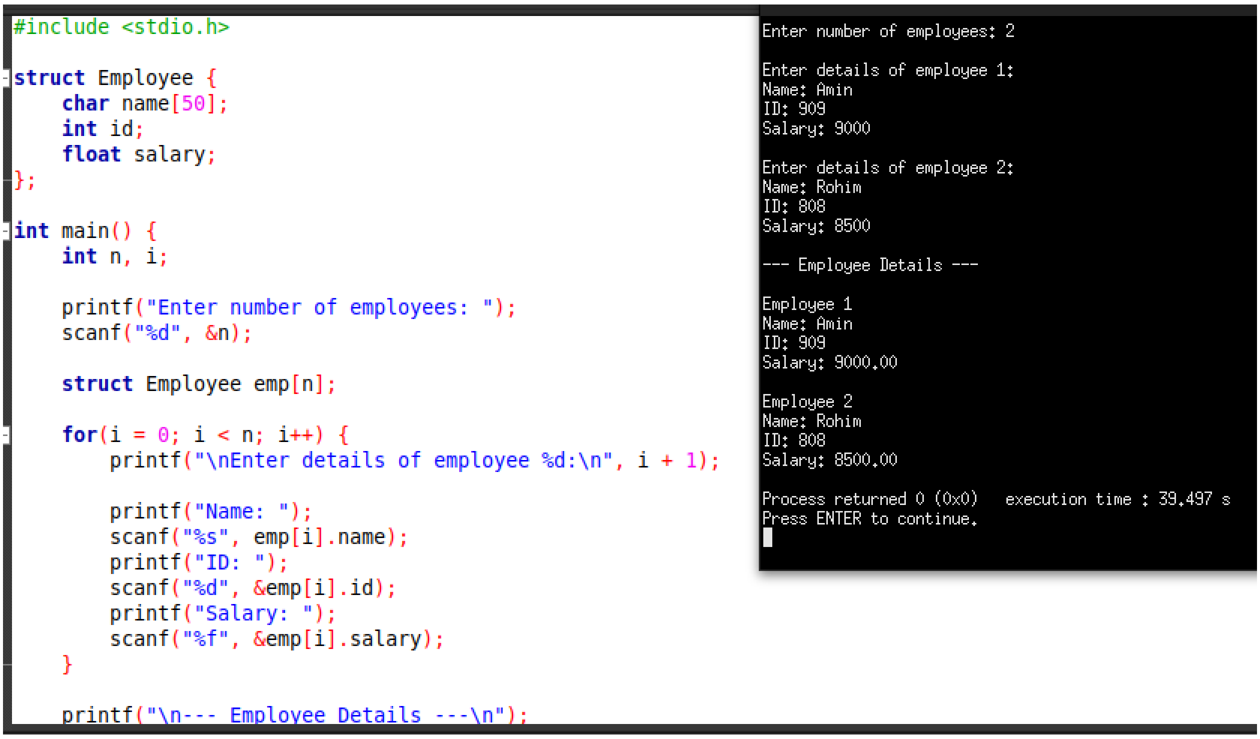
Define a union and demonstrate memory sharing between members.



```c
#include <stdio.h>

union Data {
    int i;
    float f;
    char ch;
}d;

int main() {

    d.i = 10;
    printf("i = %d\n", d.i);

    d.f = 20.5;
    printf("f = %.2f\n", d.f);

    d.ch = 'A';
    printf("ch = %c\n", d.ch);

    printf("\nAfter memory sharing:\n");
    printf("i = %d\n", d.i);
    printf("f = %.2f\n", d.f);
    printf("ch = %c\n", d.ch);

    return 0;
}
```

```
i = 10
f = 20.50
ch = A

After memory sharing:
i = 1101267009
f = 20.50
ch = A

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

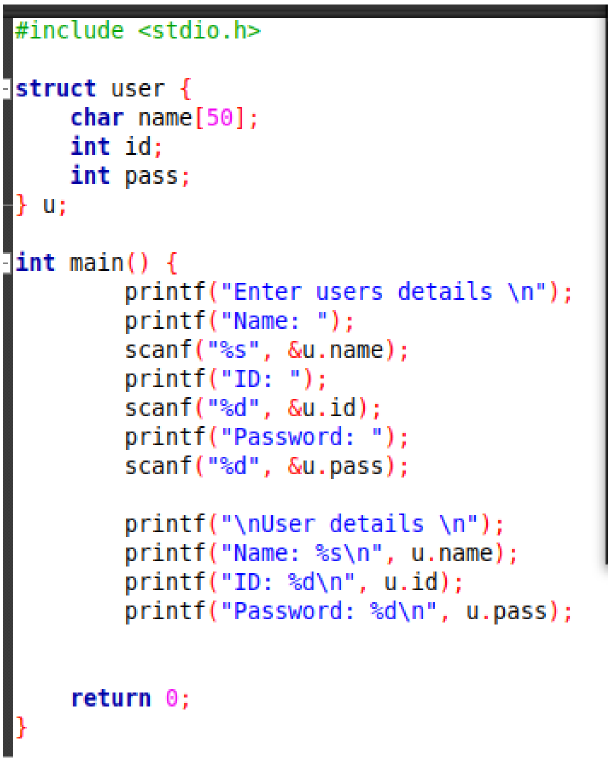
Create a program to store the details of multiple employees using structures.



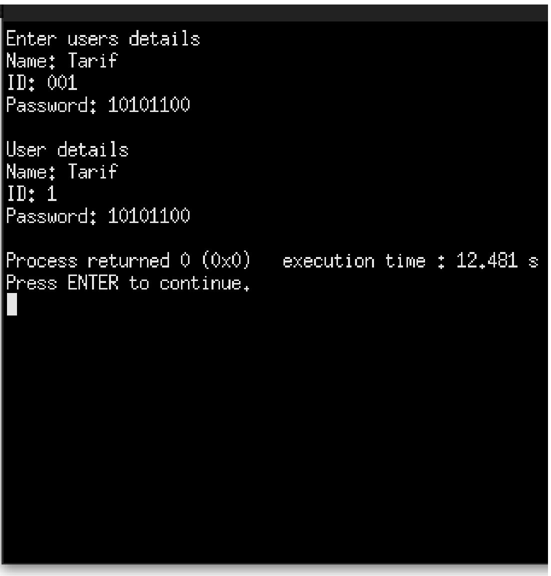
```c
#include <stdio.h>

struct Employee {
    char name[50];
    int id;
    float salary;
};

int main() {
    int n, i;

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

    struct Employee emp[n];

    for(i = 0; i < n; i++) {
        printf("\nEnter details of employee %d:\n", i + 1);

        printf("Name: ");
        scanf("%s", emp[i].name);
        printf("ID: ");
        scanf("%d", &emp[i].id);
        printf("Salary: ");
        scanf("%f", &emp[i].salary);
    }

    printf("\n--- Employee Details ---\n");
```

```
Enter number of employees: 2

Enter details of employee 1:
Name: Amin
ID: 909
Salary: 9000

Enter details of employee 2:
Name: Rohim
ID: 808
Salary: 8500

--- Employee Details ---

Employee 1
Name: Amin
ID: 909
Salary: 9000.00

Employee 2
Name: Rohim
ID: 808
Salary: 8500.00

Process returned 0 (0x0)   execution time : 39.497 s
Press ENTER to continue.
```

Write a program to take user input and display the details using structures.

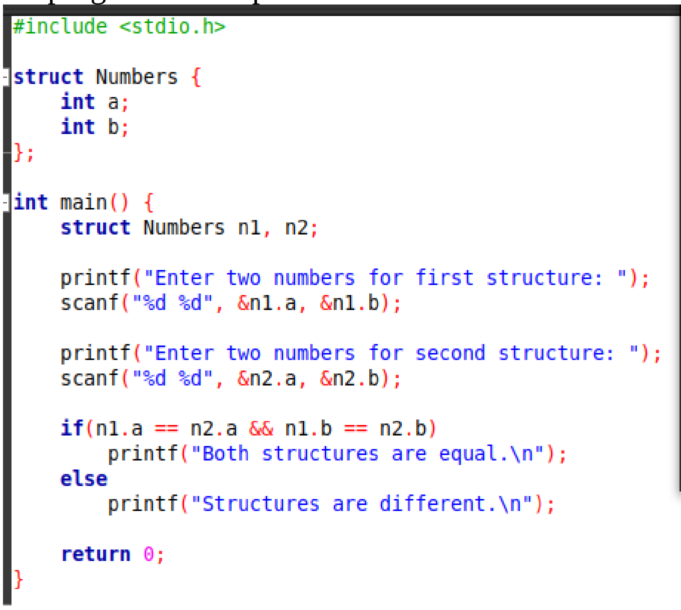


```c
#include <stdio.h>

struct user {
    char name[50];
    int id;
    int pass;
} u;

int main() {
        printf("Enter users details \n");
        printf("Name: ");
        scanf("%s", &u.name);
        printf("ID: ");
        scanf("%d", &u.id);
        printf("Password: ");
        scanf("%d", &u.pass);

        printf("\nUser details \n");
        printf("Name: %s\n", u.name);
        printf("ID: %d\n", u.id);
        printf("Password: %d\n", u.pass);


    return 0;
}
```

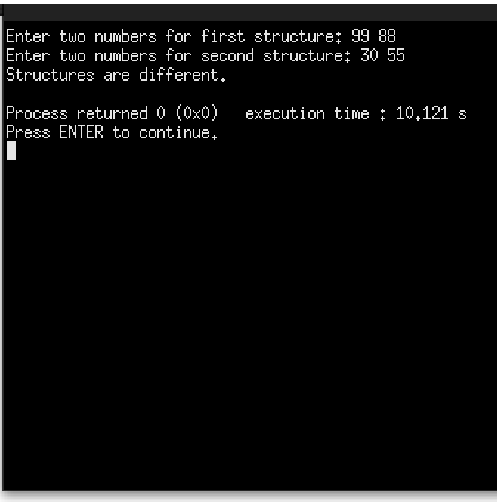


```
Enter users details
Name: Tarif
ID: 001
Password: 10101100

User details
Name: Tarif
ID: 1
Password: 10101100

Process returned 0 (0x0)   execution time : 12.481 s
Press ENTER to continue.
```

Write a program to compare two structures.



```c
#include <stdio.h>

struct Numbers {
    int a;
    int b;
};

int main() {
    struct Numbers n1, n2;

    printf("Enter two numbers for first structure: ");
    scanf("%d %d", &n1.a, &n1.b);

    printf("Enter two numbers for second structure: ");
    scanf("%d %d", &n2.a, &n2.b);

    if(n1.a == n2.a && n1.b == n2.b)
        printf("Both structures are equal.\n");
    else
        printf("Structures are different.\n");

    return 0;
}
```



```
Enter two numbers for first structure: 99 88
Enter two numbers for second structure: 30 55
Structures are different.

Process returned 0 (0x0)   execution time : 10.121 s
Press ENTER to continue.
```