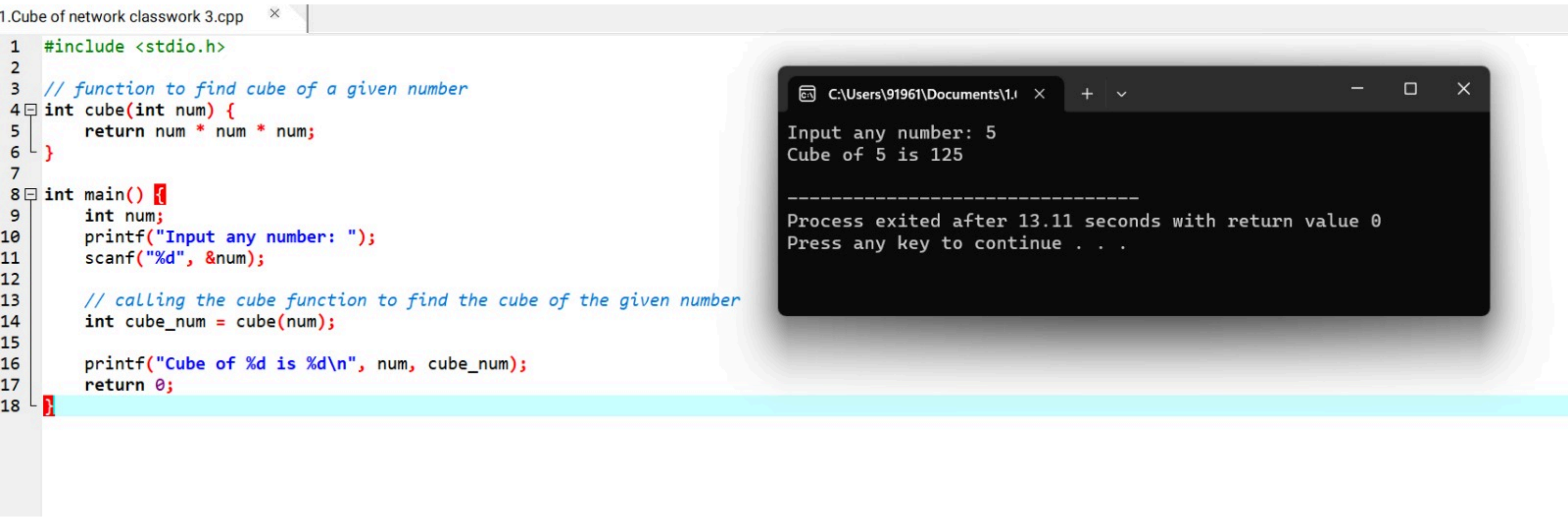
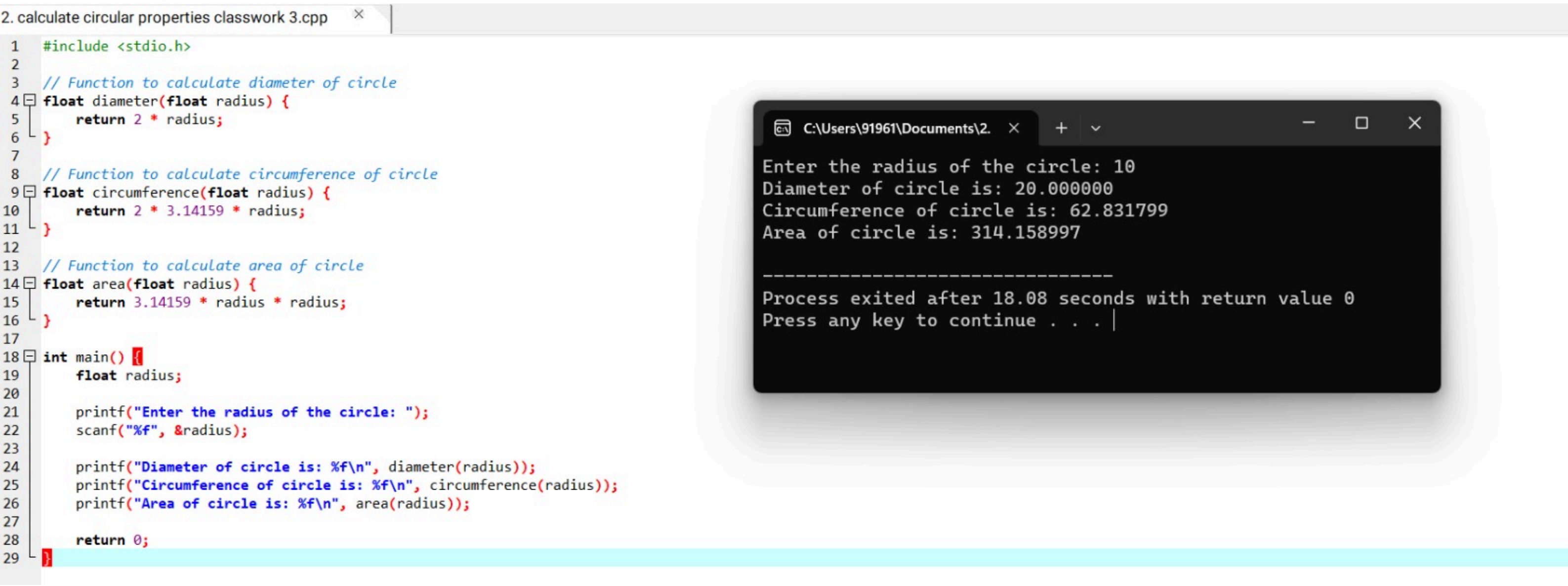

1.Cube of network classwork 3.cpp
C:\Users\91961\Documents\1.(

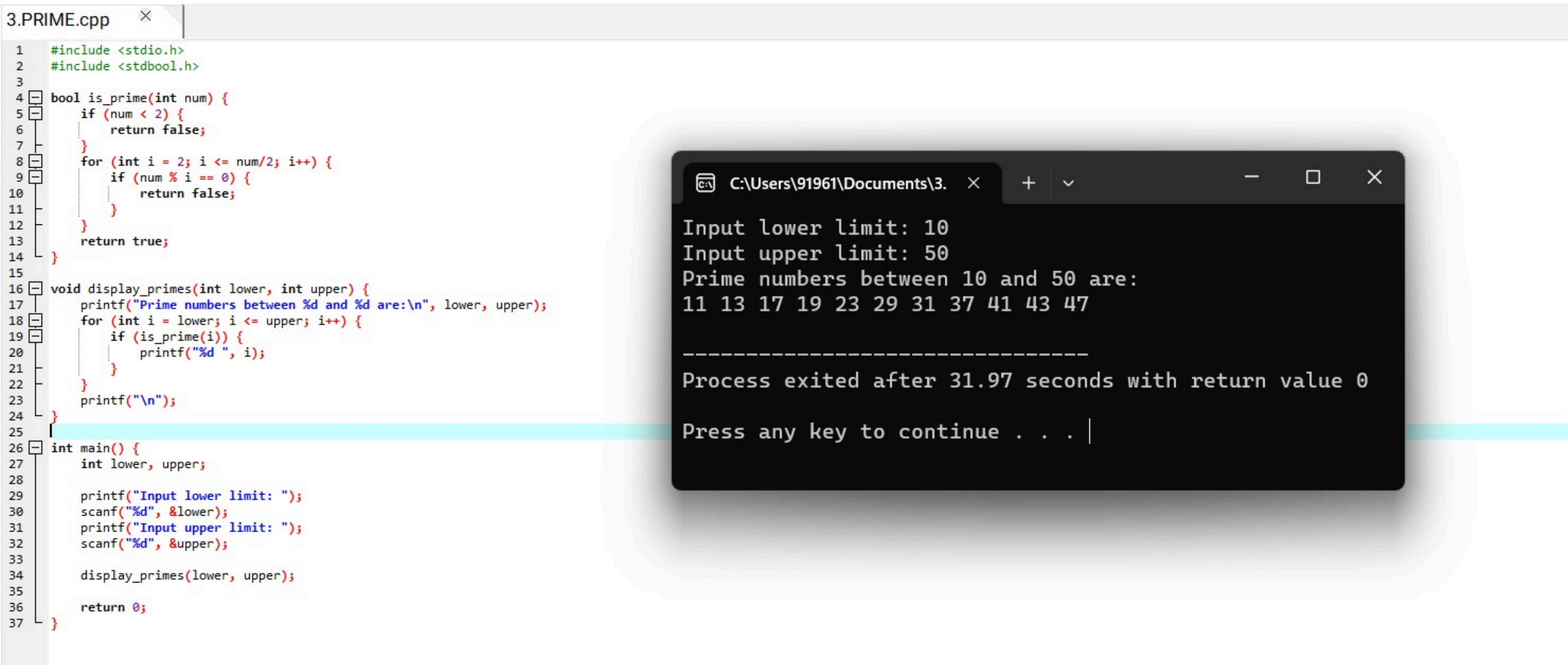
```c
#include <stdio.h>

// function to find cube of a given number
int cube(int num) {
    return num * num * num;
}

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

    // calling the cube function to find the cube of the given number
    int cube_num = cube(num);

    printf("Cube of %d is %d\n", num, cube_num);
    return 0;
}
```

```
Input any number: 5
Cube of 5 is 125

--------------------------------
Process exited after 13.11 seconds with return value 0
Press any key to continue . . .
```


2. calculate circular properties classwork 3.cpp
C:\Users\91961\Documents\2.

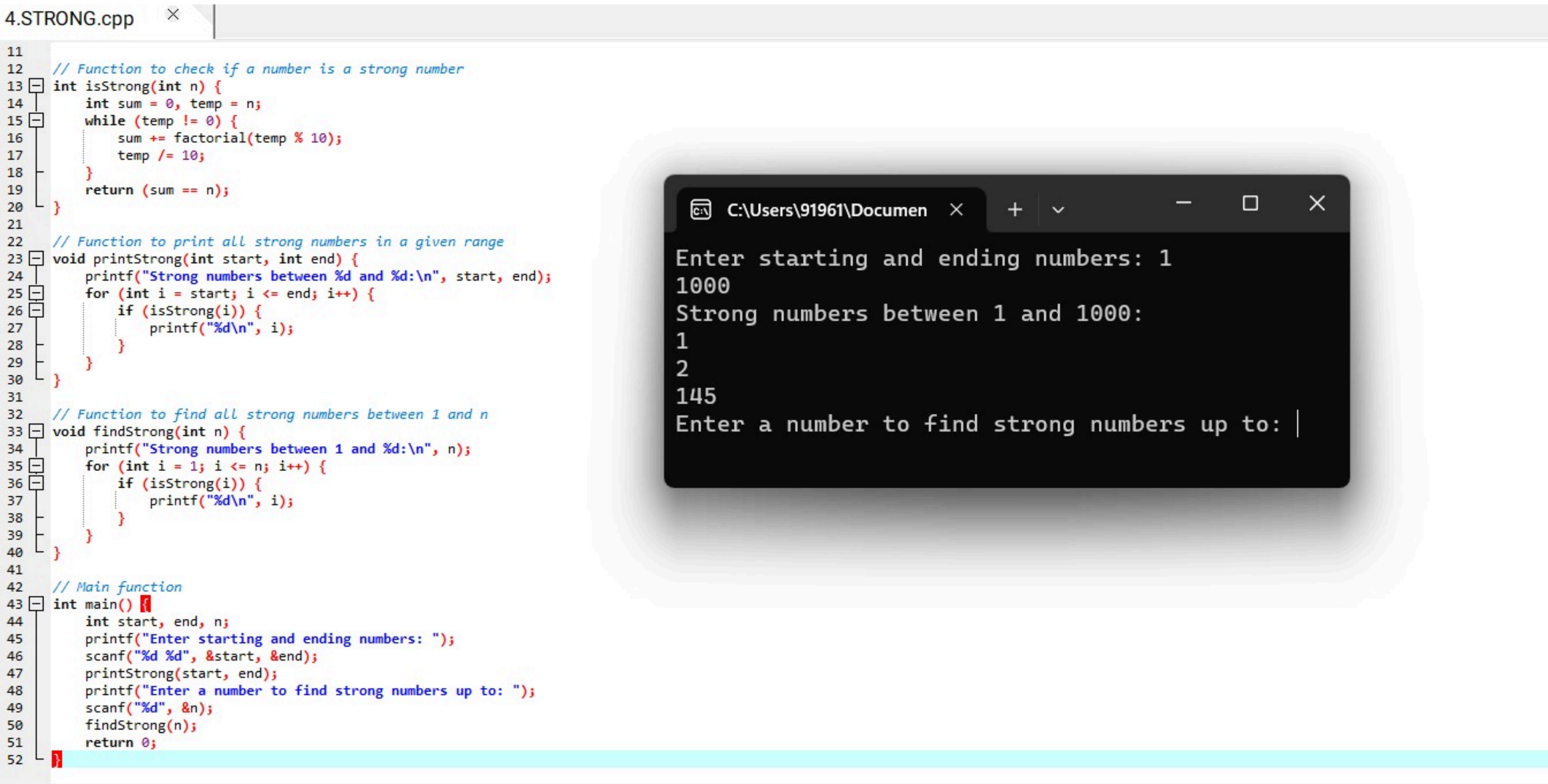
```c
#include <stdio.h>

// Function to calculate diameter of circle
float diameter(float radius) {
    return 2 * radius;
}

// Function to calculate circumference of circle
float circumference(float radius) {
    return 2 * 3.14159 * radius;
}

// Function to calculate area of circle
float area(float radius) {
    return 3.14159 * radius * radius;
}

int main() {
    float radius;

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

    printf("Diameter of circle is: %f\n", diameter(radius));
    printf("Circumference of circle is: %f\n", circumference(radius));
    printf("Area of circle is: %f\n", area(radius));

    return 0;
}
```

```
Enter the radius of the circle: 10
Diameter of circle is: 20.000000
Circumference of circle is: 62.831799
Area of circle is: 314.158997

--------------------------------
Process exited after 18.08 seconds with return value 0
Press any key to continue . . .
```



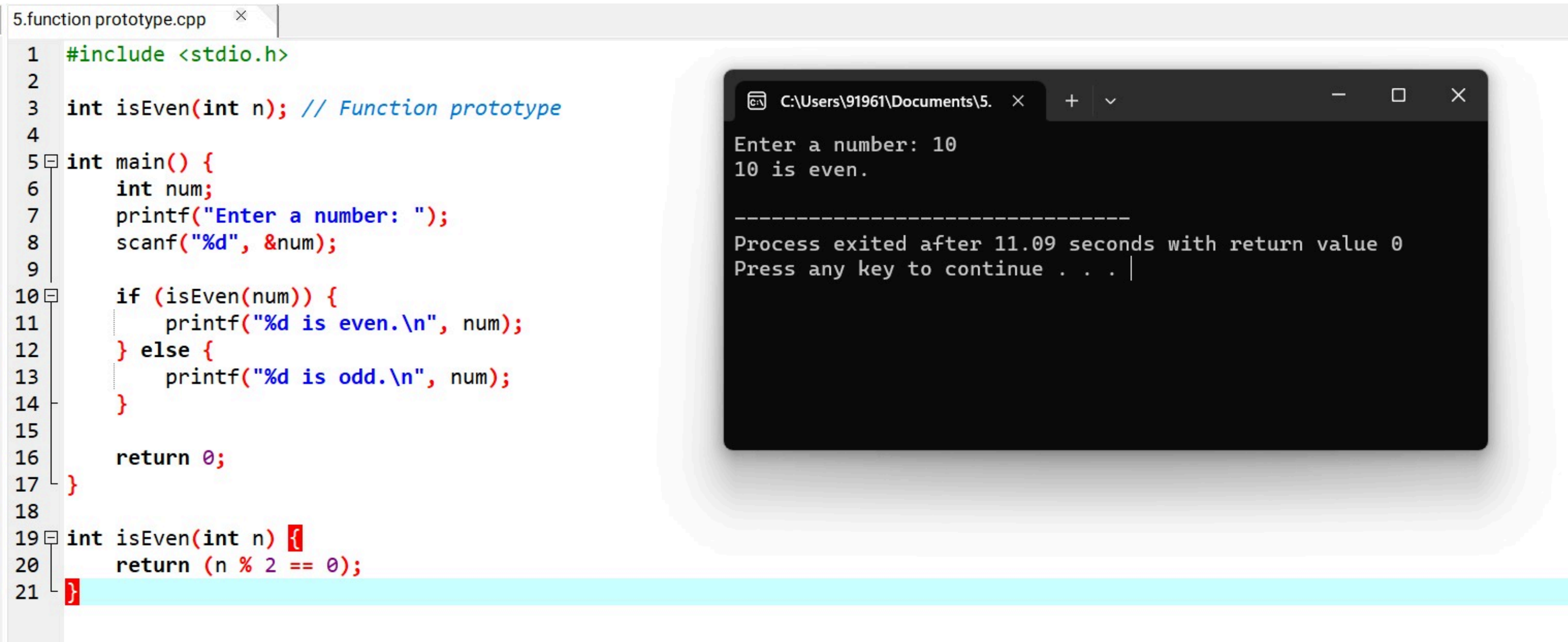
```c
#include <stdio.h>
#include <stdbool.h>

bool is_prime(int num) {
    if (num < 2) {
        return false;
    }
    for (int i = 2; i <= num/2; i++) {
        if (num % i == 0) {
            return false;
        }
    }
    return true;
}

void display_primes(int lower, int upper) {
    printf("Prime numbers between %d and %d are:\n", lower, upper);
    for (int i = lower; i <= upper; i++) {
        if (is_prime(i)) {
            printf("%d ", i);
        }
    }
    printf("\n");
}

int main() {
    int lower, upper;

    printf("Input lower limit: ");
    scanf("%d", &lower);
    printf("Input upper limit: ");
    scanf("%d", &upper);

    display_primes(lower, upper);

    return 0;
}
```

```
Input lower limit: 10
Input upper limit: 50
Prime numbers between 10 and 50 are:
11 13 17 19 23 29 31 37 41 43 47

--------------------------------
Process exited after 31.97 seconds with return value 0

Press any key to continue . . .
```



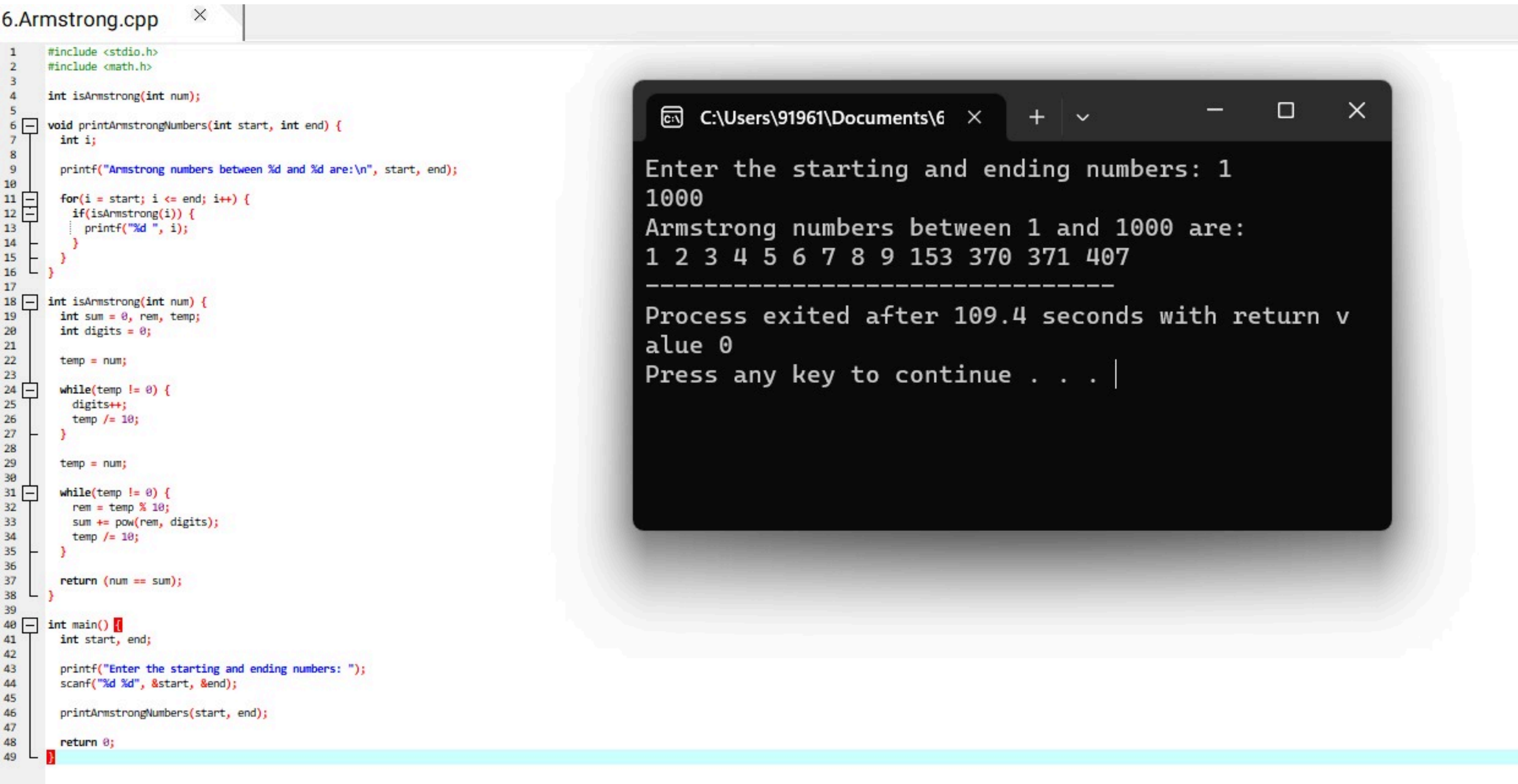
```

// Function to check if a number is a strong number
int isStrong(int n) {
    int sum = 0, temp = n;
    while (temp != 0) {
        sum += factorial(temp % 10);
        temp /= 10;
    }
    return (sum == n);
}

// Function to print all strong numbers in a given range
void printStrong(int start, int end) {
    printf("Strong numbers between %d and %d:\n", start, end);
    for (int i = start; i <= end; i++) {
        if (isStrong(i)) {
            printf("%d\n", i);
        }
    }
}

// Function to find all strong numbers between 1 and n
void findStrong(int n) {
    printf("Strong numbers between 1 and %d:\n", n);
    for (int i = 1; i <= n; i++) {
        if (isStrong(i)) {
            printf("%d\n", i);
        }
    }
}

// Main function
int main() {
    int start, end, n;
    printf("Enter starting and ending numbers: ");
    scanf("%d %d", &start, &end);
    printStrong(start, end);
    printf("Enter a number to find strong numbers up to: ");
    scanf("%d", &n);
    findStrong(n);
    return 0;
}
```

```
Enter starting and ending numbers: 1
1000
Strong numbers between 1 and 1000:
1
2
145
Enter a number to find strong numbers up to:
```



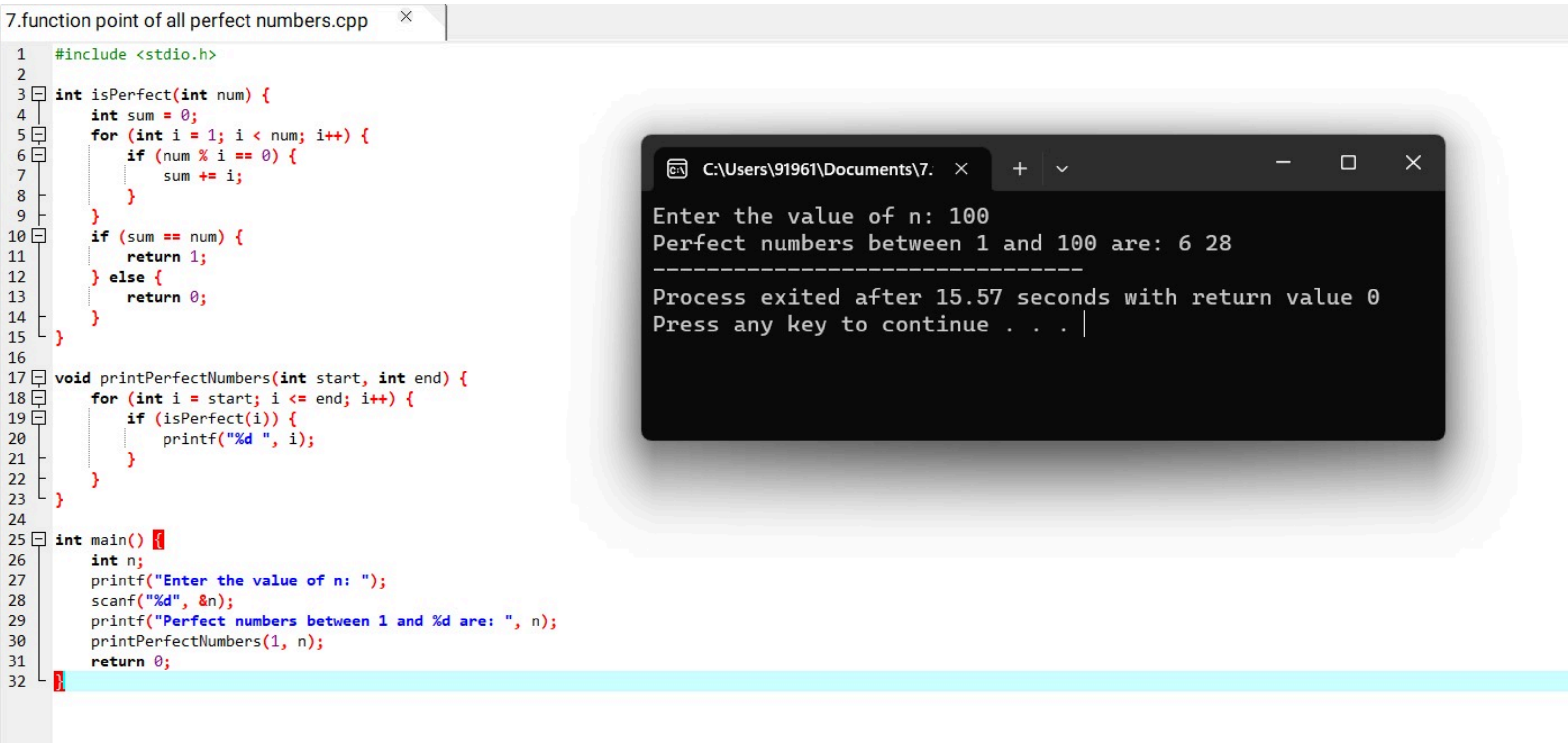
```c
#include <stdio.h>

int isEven(int n); // Function prototype

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

    if (isEven(num)) {
        printf("%d is even.\n", num);
    } else {
        printf("%d is odd.\n", num);
    }

    return 0;
}

int isEven(int n) {
    return (n % 2 == 0);
}
```

```
Enter a number: 10
10 is even.

--------------------------------
Process exited after 11.09 seconds with return value 0
Press any key to continue . . .
```



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

int isArmstrong(int num);

void printArmstrongNumbers(int start, int end) {
  int i;

  printf("Armstrong numbers between %d and %d are:\n", start, end);

  for(i = start; i <= end; i++) {
    if(isArmstrong(i)) {
      printf("%d ", i);
    }
  }
}

int isArmstrong(int num) {
  int sum = 0, rem, temp;
  int digits = 0;

  temp = num;

  while(temp != 0) {
    digits++;
    temp /= 10;
  }

  temp = num;

  while(temp != 0) {
    rem = temp % 10;
    sum += pow(rem, digits);
    temp /= 10;
  }

  return (num == sum);
}

int main() {
  int start, end;

  printf("Enter the starting and ending numbers: ");
  scanf("%d %d", &start, &end);

  printArmstrongNumbers(start, end);

  return 0;
}
```

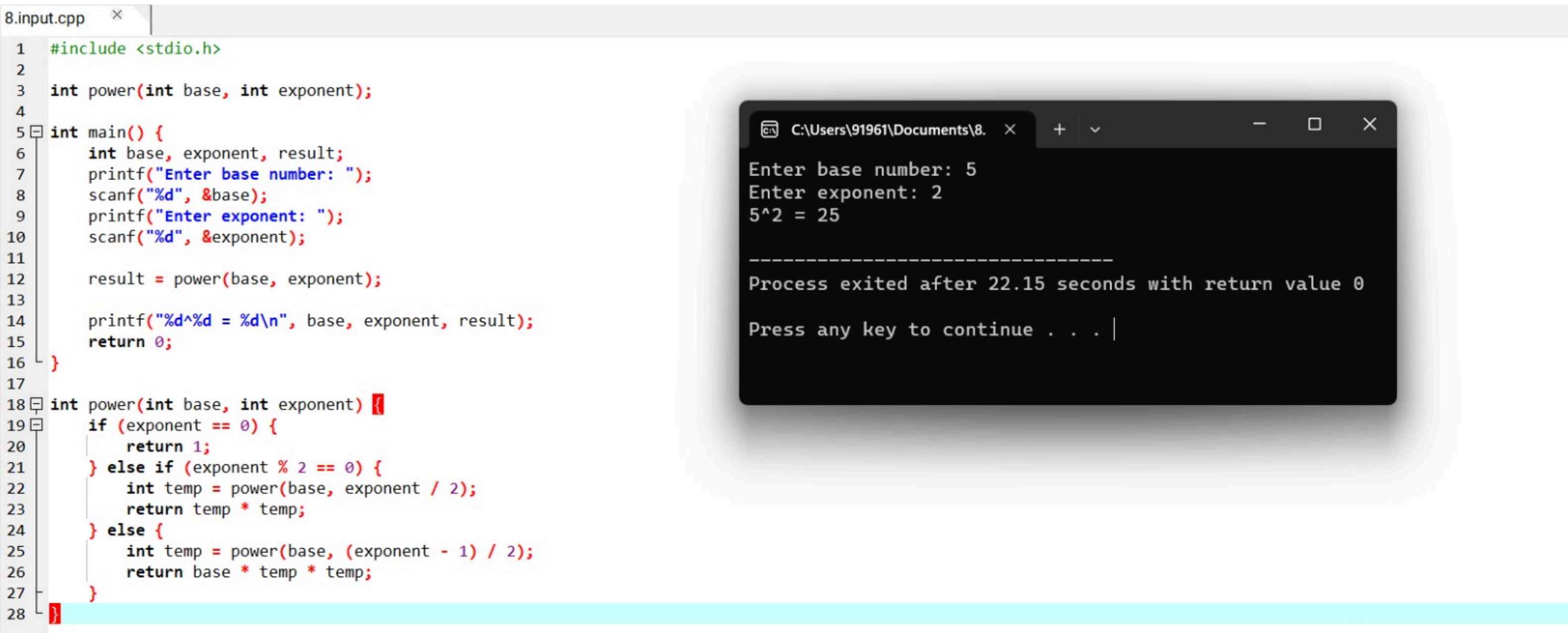
```
Enter the starting and ending numbers: 1
1000
Armstrong numbers between 1 and 1000 are:
1 2 3 4 5 6 7 8 9 153 370 371 407
--------------------------------
Process exited after 109.4 seconds with return v
alue 0
Press any key to continue . . .
```



7.function point of all perfect numbers.cpp
C:\Users\91961\Documents\7.

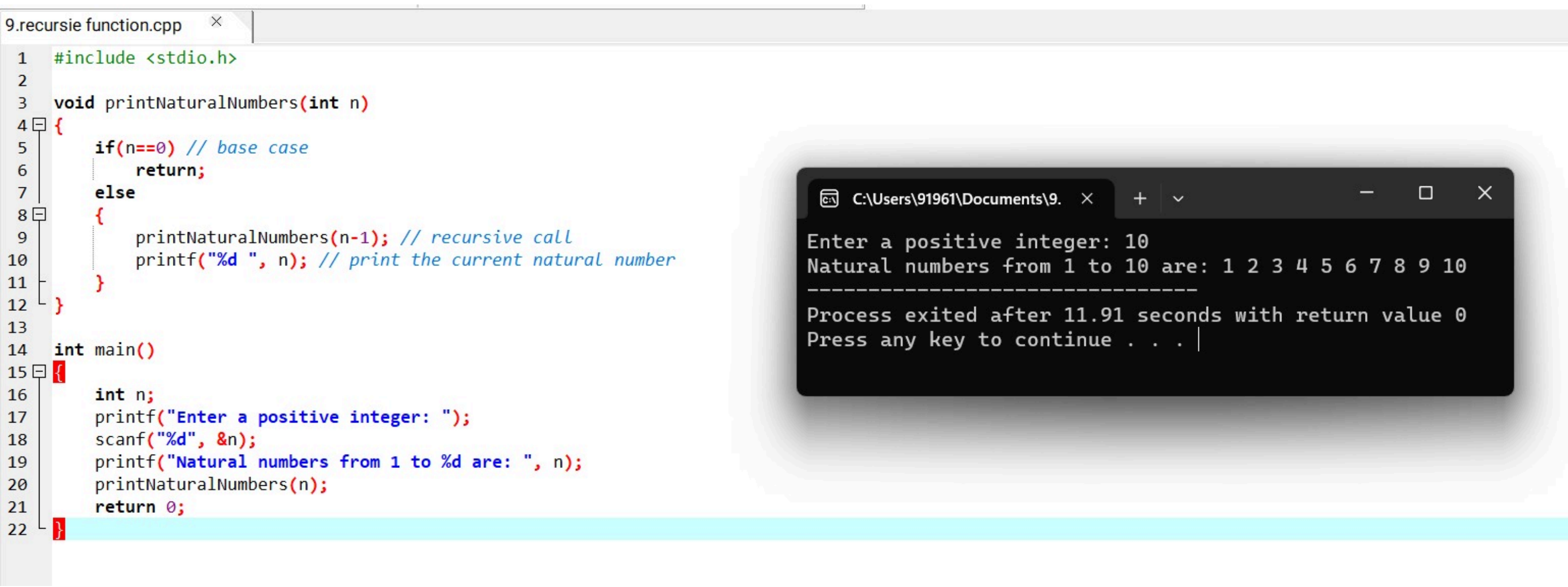
```c
#include <stdio.h>

int isPerfect(int num) {
    int sum = 0;
    for (int i = 1; i < num; i++) {
        if (num % i == 0) {
            sum += i;
        }
    }
    if (sum == num) {
        return 1;
    } else {
        return 0;
    }
}

void printPerfectNumbers(int start, int end) {
    for (int i = start; i <= end; i++) {
        if (isPerfect(i)) {
            printf("%d ", i);
        }
    }
}

int main() {
    int n;
    printf("Enter the value of n: ");
    scanf("%d", &n);
    printf("Perfect numbers between 1 and %d are: ", n);
    printPerfectNumbers(1, n);
    return 0;
}
```

```
Enter the value of n: 100
Perfect numbers between 1 and 100 are: 6 28
--------------------------------
Process exited after 15.57 seconds with return value 0
Press any key to continue . . .
```



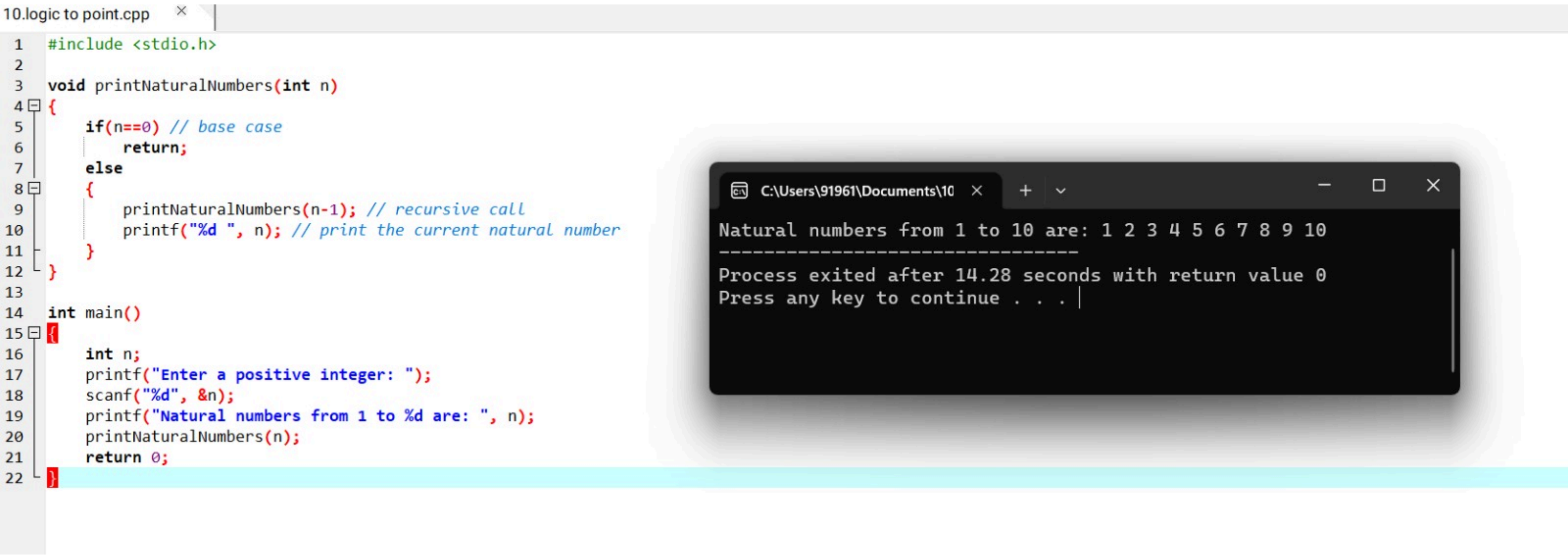
```c
#include <stdio.h>

int power(int base, int exponent);

int main() {
    int base, exponent, result;
    printf("Enter base number: ");
    scanf("%d", &base);
    printf("Enter exponent: ");
    scanf("%d", &exponent);

    result = power(base, exponent);

    printf("%d^%d = %d\n", base, exponent, result);
    return 0;
}

int power(int base, int exponent) {
    if (exponent == 0) {
        return 1;
    } else if (exponent % 2 == 0) {
        int temp = power(base, exponent / 2);
        return temp * temp;
    } else {
        int temp = power(base, (exponent - 1) / 2);
        return base * temp * temp;
    }
}
```

```
Enter base number: 5
Enter exponent: 2
5^2 = 25

--------------------------------
Process exited after 22.15 seconds with return value 0

Press any key to continue . . .
```



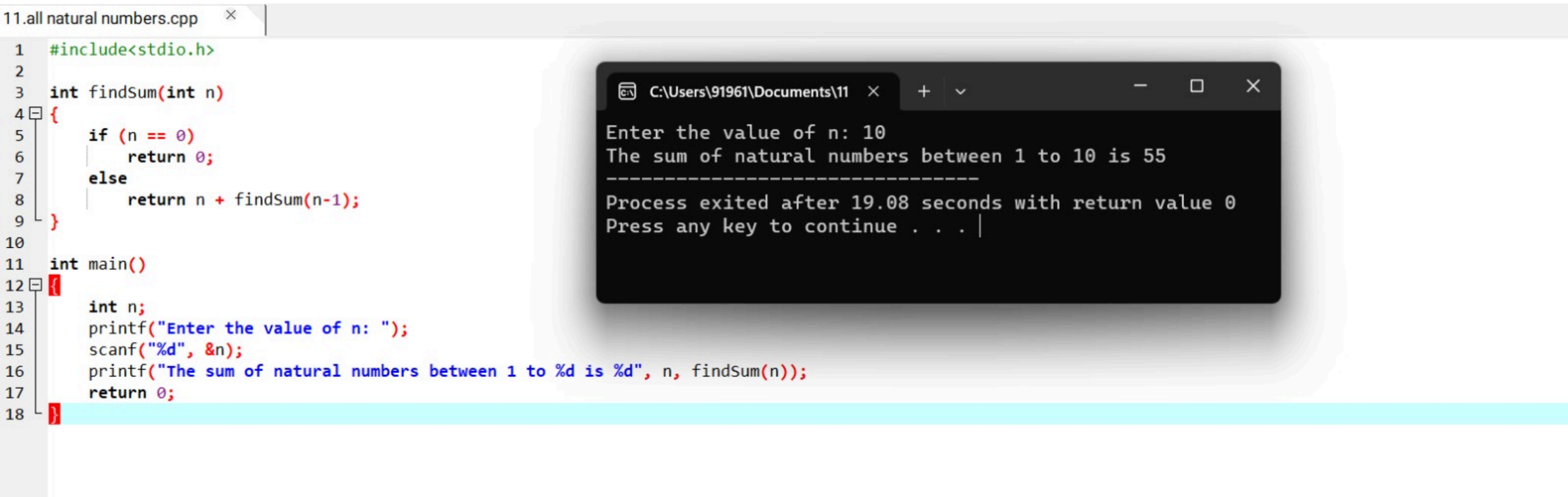
```c
#include <stdio.h>

void printNaturalNumbers(int n)
{
    if(n==0) // base case
        return;
    else
    {
        printNaturalNumbers(n-1); // recursive call
        printf("%d ", n); // print the current natural number
    }
}

int main()
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Natural numbers from 1 to %d are: ", n);
    printNaturalNumbers(n);
    return 0;
}
```

```
Enter a positive integer: 10
Natural numbers from 1 to 10 are: 1 2 3 4 5 6 7 8 9 10
--------------------------------
Process exited after 11.91 seconds with return value 0
Press any key to continue . . .
```



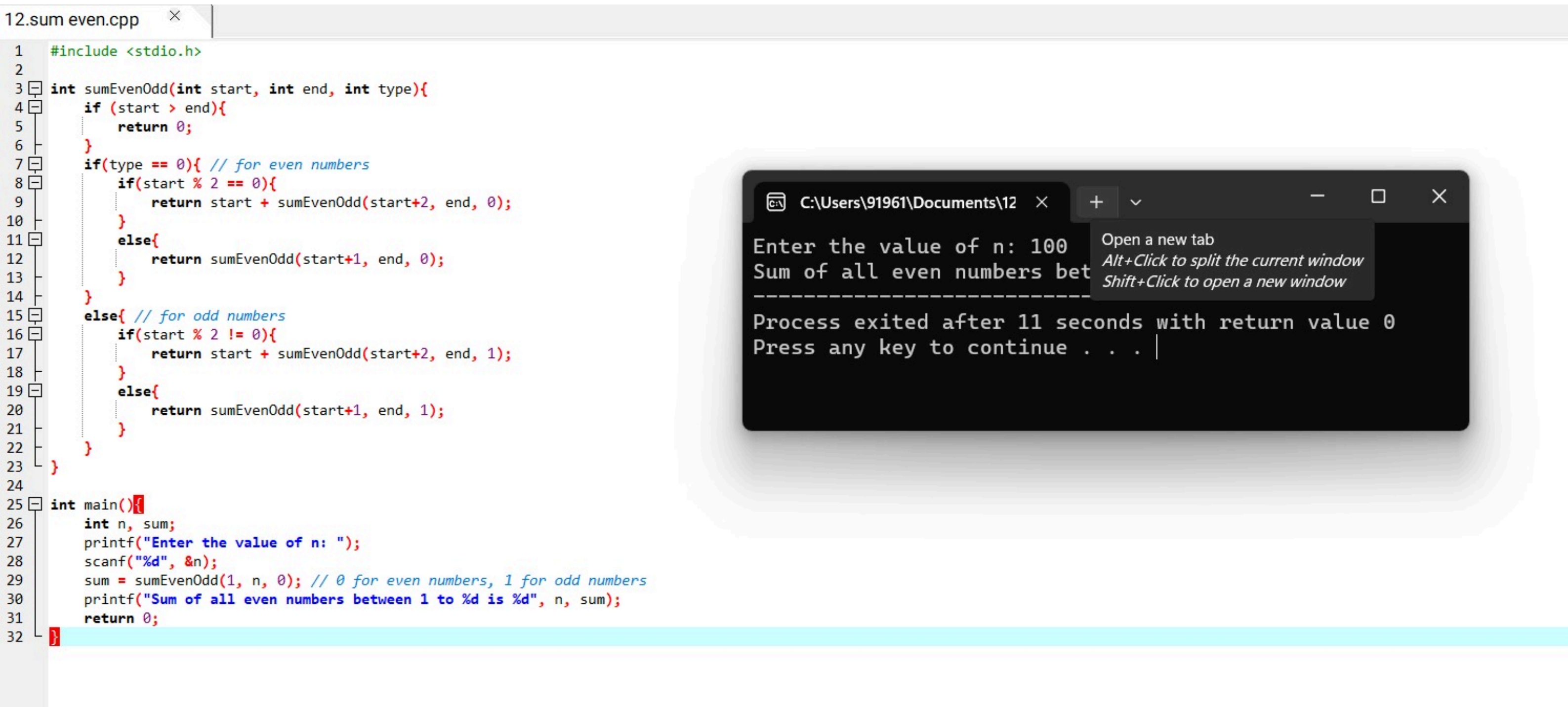
```c
#include <stdio.h>

void printNaturalNumbers(int n)
{
    if(n==0) // base case
        return;
    else
    {
        printNaturalNumbers(n-1); // recursive call
        printf("%d ", n); // print the current natural number
    }
}

int main()
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Natural numbers from 1 to %d are: ", n);
    printNaturalNumbers(n);
    return 0;
}
```

```
Natural numbers from 1 to 10 are: 1 2 3 4 5 6 7 8 9 10
--------------------------------
Process exited after 14.28 seconds with return value 0
Press any key to continue . . .
```



```c
#include<stdio.h>

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

int main()
{
    int n;
    printf("Enter the value of n: ");
    scanf("%d", &n);
    printf("The sum of natural numbers between 1 to %d is %d", n, findSum(n));
    return 0;
}
```

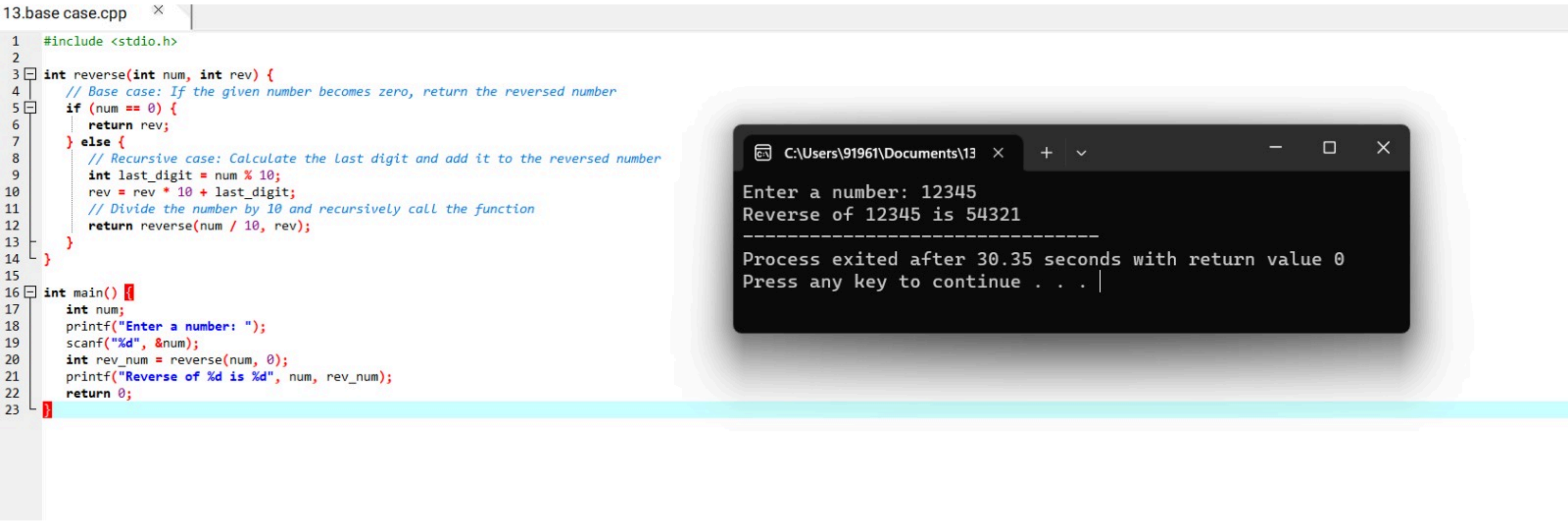
```
Enter the value of n: 10
The sum of natural numbers between 1 to 10 is 55
--------------------------------
Process exited after 19.08 seconds with return value 0
Press any key to continue . . .
```



12.sum even.cpp
C:\Users\91961\Documents\12
Open a new tab
Alt+Click to split the current window
Shift+Click to open a new window

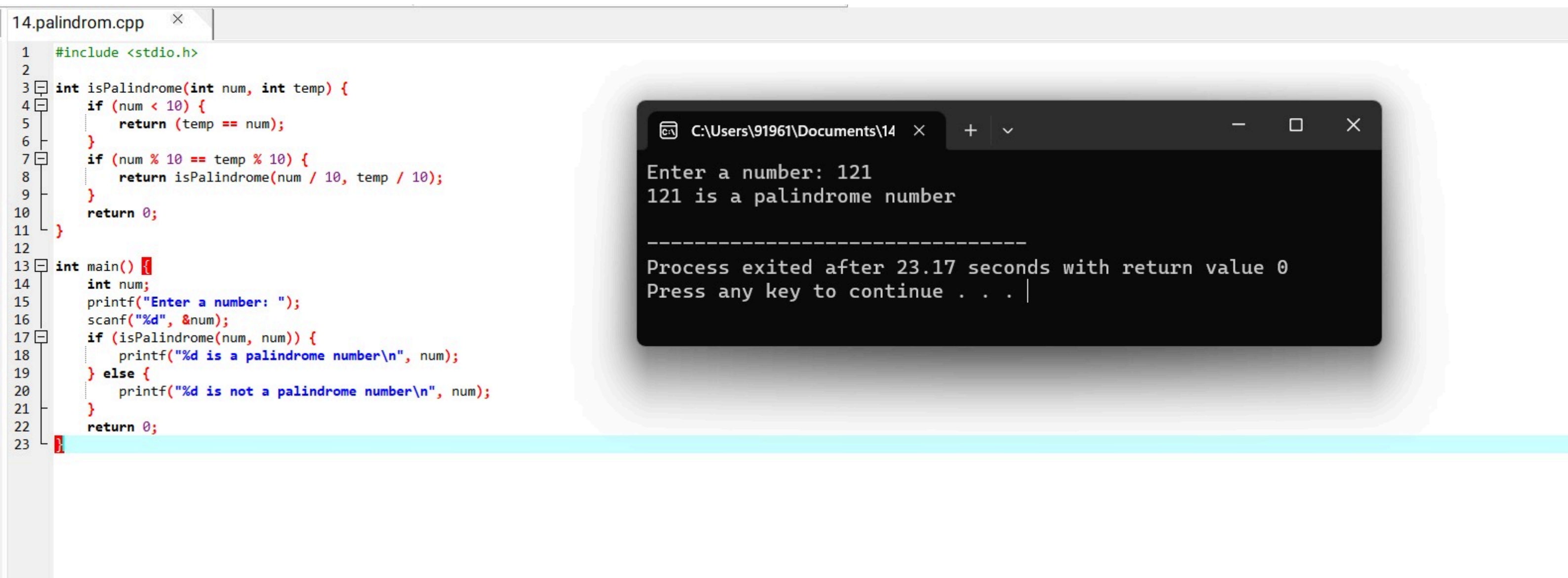
```c
#include <stdio.h>

int sumEvenOdd(int start, int end, int type){
    if (start > end){
        return 0;
    }
    if(type == 0){ // for even numbers
        if(start % 2 == 0){
            return start + sumEvenOdd(start+2, end, 0);
        }
        else{
            return sumEvenOdd(start+1, end, 0);
        }
    }
    else{ // for odd numbers
        if(start % 2 != 0){
            return start + sumEvenOdd(start+2, end, 1);
        }
        else{
            return sumEvenOdd(start+1, end, 1);
        }
    }
}

int main(){
    int n, sum;
    printf("Enter the value of n: ");
    scanf("%d", &n);
    sum = sumEvenOdd(1, n, 0); // 0 for even numbers, 1 for odd numbers
    printf("Sum of all even numbers between 1 to %d is %d", n, sum);
    return 0;
}
```

```
Enter the value of n: 100
Sum of all even numbers bet
---------------------------
Process exited after 11 seconds with return value 0
Press any key to continue . . .
```



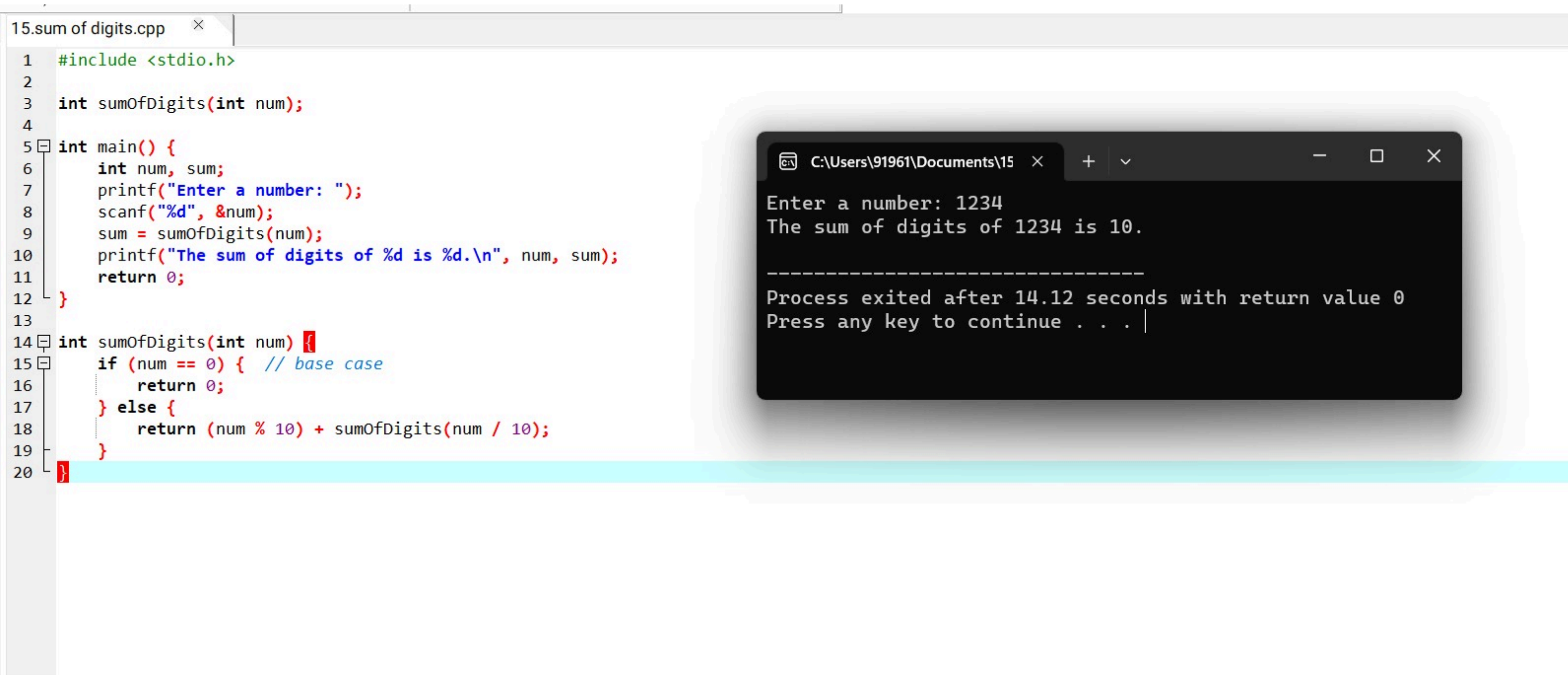
```c
#include <stdio.h>

int reverse(int num, int rev) {
    // Base case: If the given number becomes zero, return the reversed number
    if (num == 0) {
        return rev;
    } else {
        // Recursive case: Calculate the last digit and add it to the reversed number
        int last_digit = num % 10;
        rev = rev * 10 + last_digit;
        // Divide the number by 10 and recursively call the function
        return reverse(num / 10, rev);
    }
}

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

```
Enter a number: 12345
Reverse of 12345 is 54321
--------------------------------
Process exited after 30.35 seconds with return value 0
Press any key to continue . . .
```



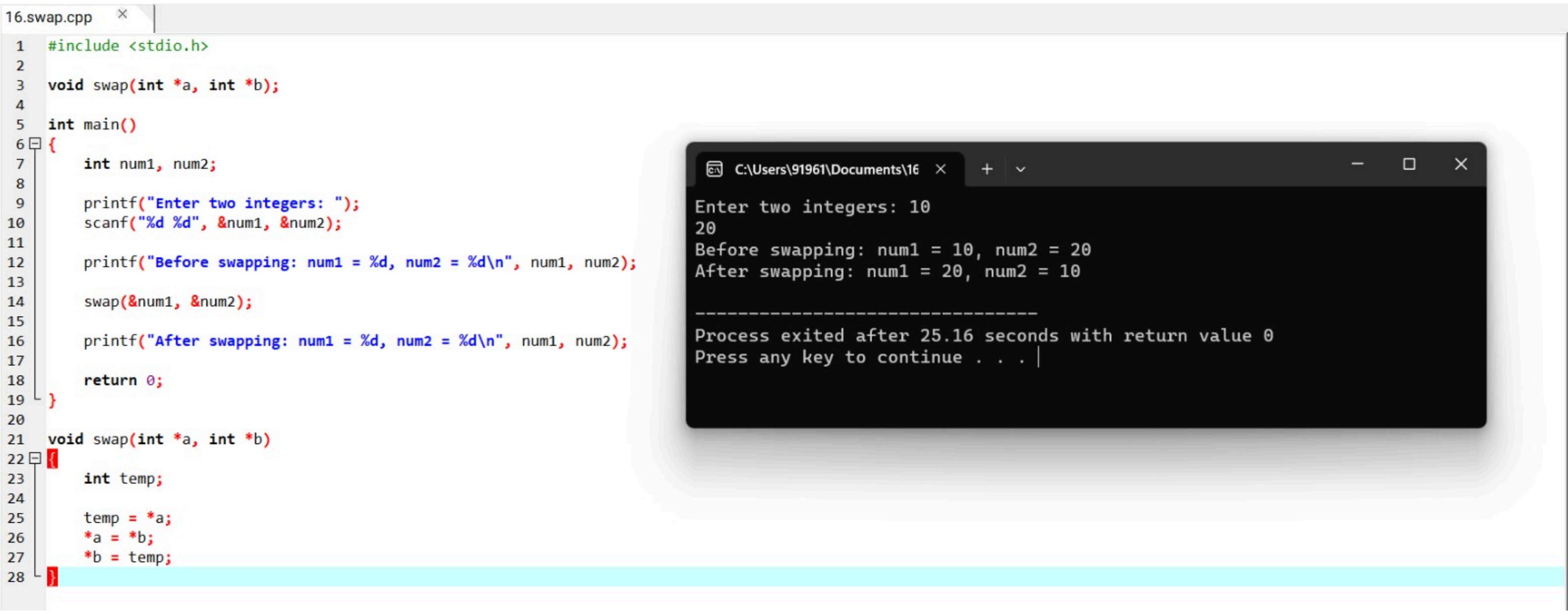
```c
#include <stdio.h>

int isPalindrome(int num, int temp) {
    if (num < 10) {
        return (temp == num);
    }
    if (num % 10 == temp % 10) {
        return isPalindrome(num / 10, temp / 10);
    }
    return 0;
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    if (isPalindrome(num, num)) {
        printf("%d is a palindrome number\n", num);
    } else {
        printf("%d is not a palindrome number\n", num);
    }
    return 0;
}
```

```
Enter a number: 121
121 is a palindrome number

--------------------------------
Process exited after 23.17 seconds with return value 0
Press any key to continue . . .
```



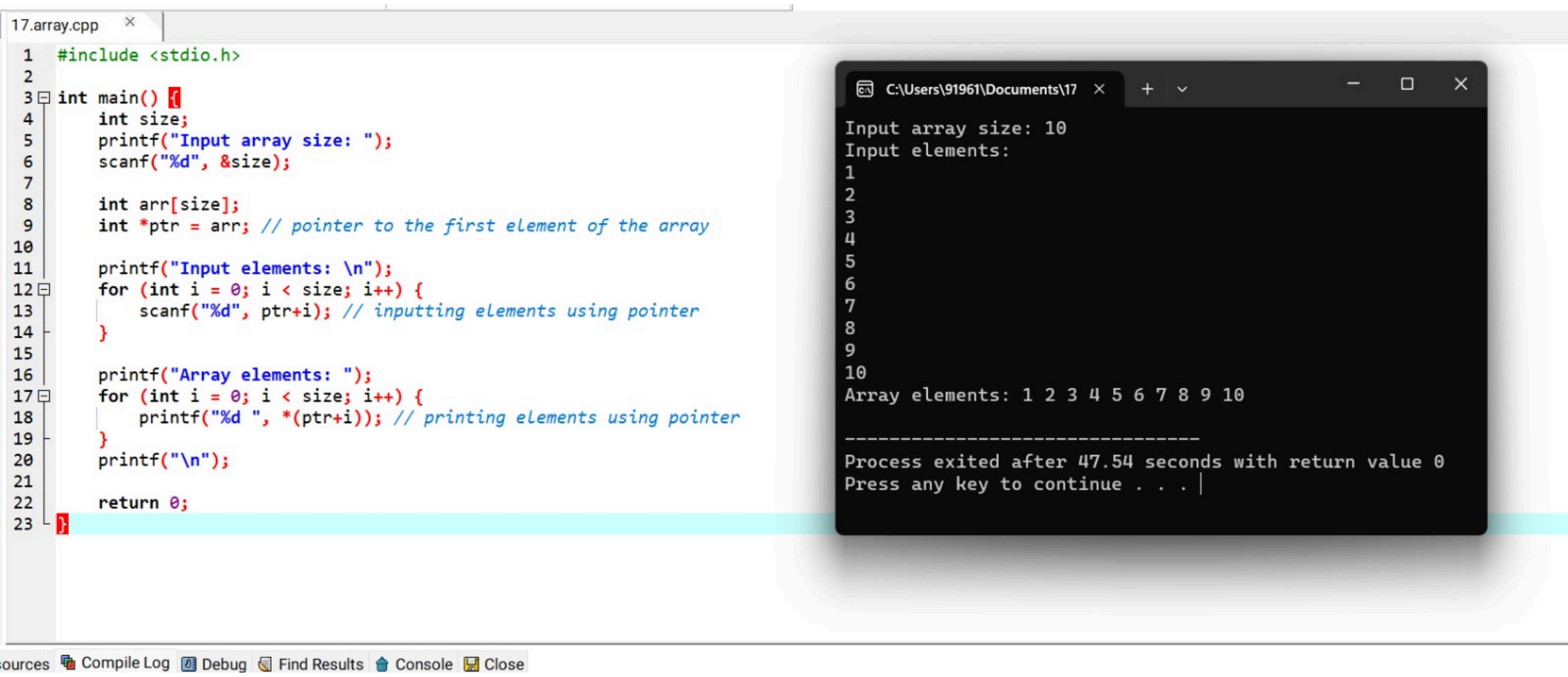
```c
#include <stdio.h>

int sumOfDigits(int num);

int main() {
    int num, sum;
    printf("Enter a number: ");
    scanf("%d", &num);
    sum = sumOfDigits(num);
    printf("The sum of digits of %d is %d.\n", num, sum);
    return 0;
}

int sumOfDigits(int num) {
    if (num == 0) {  // base case
        return 0;
    } else {
        return (num % 10) + sumOfDigits(num / 10);
    }
}
```

```
Enter a number: 1234
The sum of digits of 1234 is 10.

--------------------------------
Process exited after 14.12 seconds with return value 0
Press any key to continue . . .
```



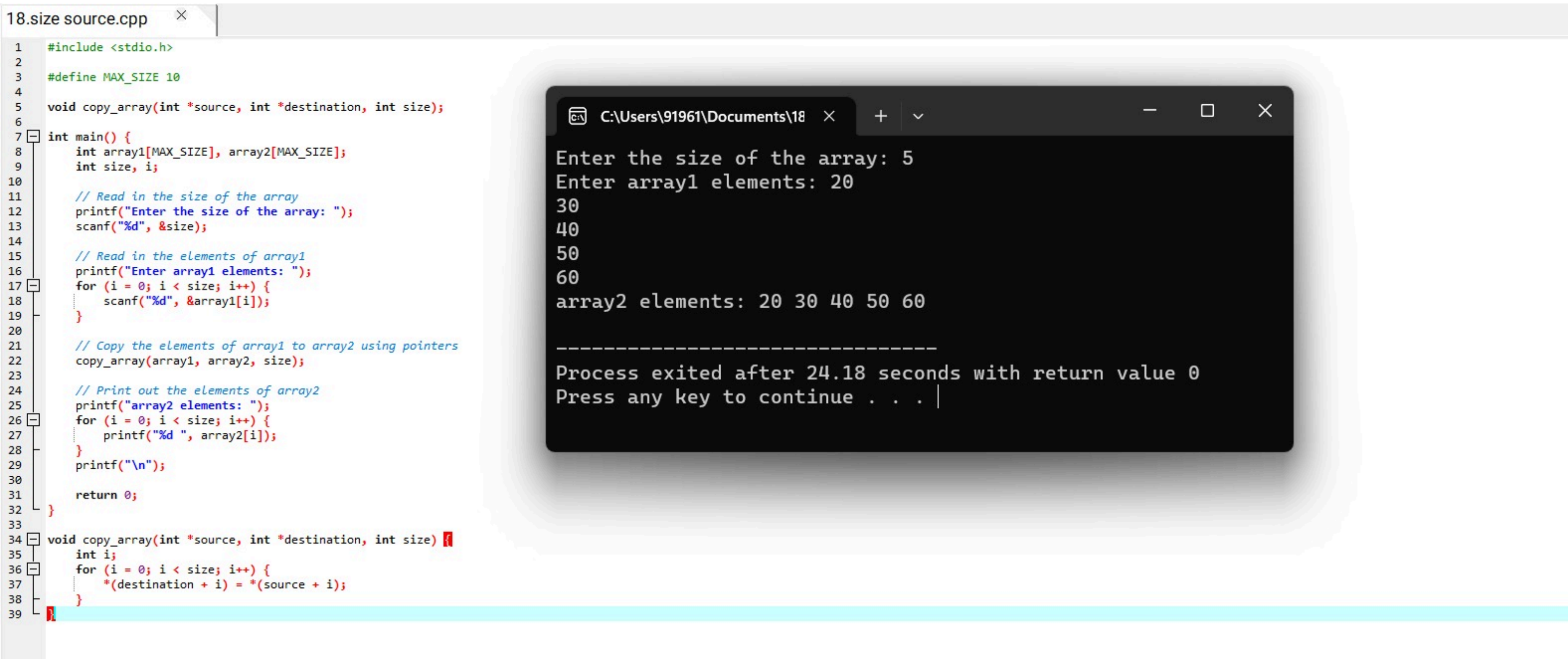
```c
#include <stdio.h>

void swap(int *a, int *b);

int main()
{
    int num1, num2;

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

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

    swap(&num1, &num2);

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

    return 0;
}

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

    temp = *a;
    *a = *b;
    *b = temp;
}
```

```
Enter two integers: 10
20
Before swapping: num1 = 10, num2 = 20
After swapping: num1 = 20, num2 = 10

--------------------------------
Process exited after 25.16 seconds with return value 0
Press any key to continue . . .
```



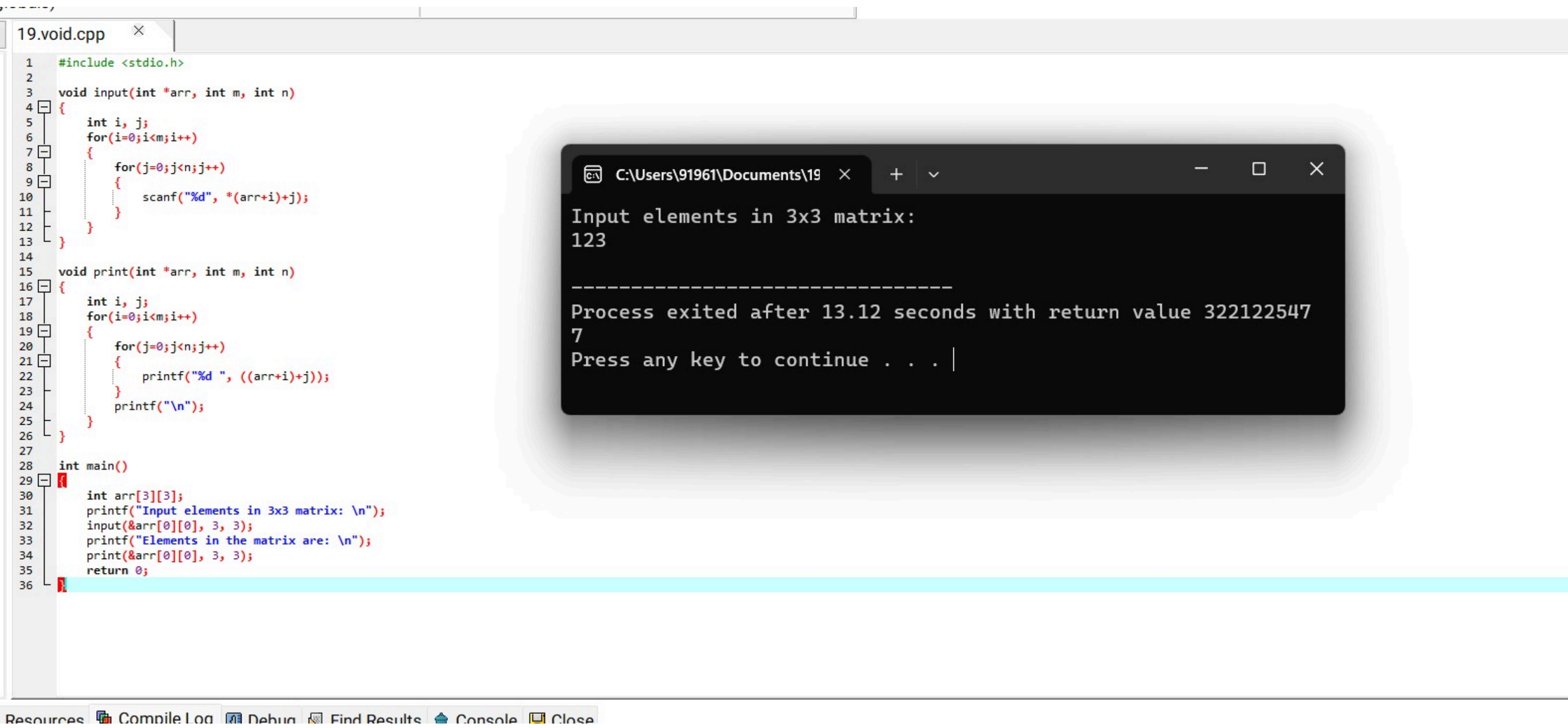
```c
#include <stdio.h>

int main() {
    int size;
    printf("Input array size: ");
    scanf("%d", &size);

    int arr[size];
    int *ptr = arr; // pointer to the first element of the array

    printf("Input elements: \n");
    for (int i = 0; i < size; i++) {
        scanf("%d", ptr+i); // inputting elements using pointer
    }

    printf("Array elements: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", *(ptr+i)); // printing elements using pointer
    }
    printf("\n");

    return 0;
}
```

```
Input array size: 10
Input elements:
1
2
3
4
5
6
7
8
9
10
Array elements: 1 2 3 4 5 6 7 8 9 10

--------------------------------
Process exited after 47.54 seconds with return value 0
Press any key to continue . . .
```



```c
#include <stdio.h>

#define MAX_SIZE 10

void copy_array(int *source, int *destination, int size);

int main() {
    int array1[MAX_SIZE], array2[MAX_SIZE];
    int size, i;

    // Read in the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    // Read in the elements of array1
    printf("Enter array1 elements: ");
    for (i = 0; i < size; i++) {
        scanf("%d", &array1[i]);
    }

    // Copy the elements of array1 to array2 using pointers
    copy_array(array1, array2, size);

    // Print out the elements of array2
    printf("array2 elements: ");
    for (i = 0; i < size; i++) {
        printf("%d ", array2[i]);
    }
    printf("\n");

    return 0;
}

void copy_array(int *source, int *destination, int size) {
    int i;
    for (i = 0; i < size; i++) {
        *(destination + i) = *(source + i);
    }
}
```

```
Enter the size of the array: 5
Enter array1 elements: 20
30
40
50
60
array2 elements: 20 30 40 50 60

--------------------------------
Process exited after 24.18 seconds with return value 0
Press any key to continue . . .
```



```c
#include <stdio.h>

void input(int *arr, int m, int n)
{
    int i, j;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d", *(arr+i)+j);
        }
    }
}

void print(int *arr, int m, int n)
{
    int i, j;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("%d ", ((arr+i)+j));
        }
        printf("\n");
    }
}

int main()
{
    int arr[3][3];
    printf("Input elements in 3x3 matrix: \n");
    input(&arr[0][0], 3, 3);
    printf("Elements in the matrix are: \n");
    print(&arr[0][0], 3, 3);
    return 0;
}
```

```
Input elements in 3x3 matrix:
123

--------------------------------
Process exited after 13.12 seconds with return value 322122547
7
Press any key to continue . . .
```