Loop 1

1. Write the following programs by using loop

   I. Print “Southeast University” 10 times.

   II. Print all numbers from 1 to 10.

   III. Print all odd numbers from 1 to 10.

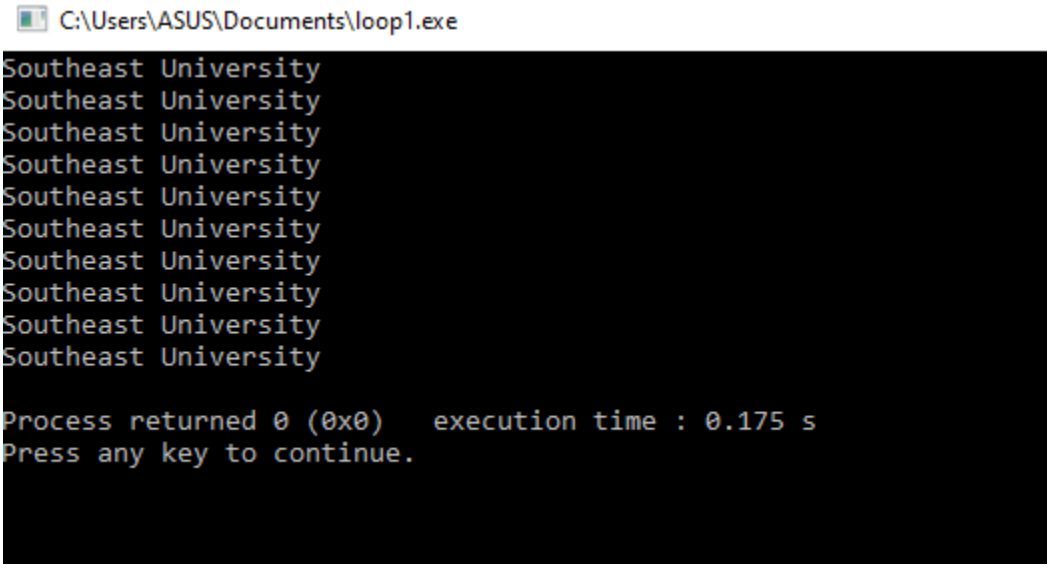
   IV. Print all even numbers from 1 to 10.

Input:

i.

```
#include <stdio.h>


int main() {
   for (int i = 0; i < 10; i++) {
      printf("Southeast University\n");
   }
   return 0;
}
```

Output:

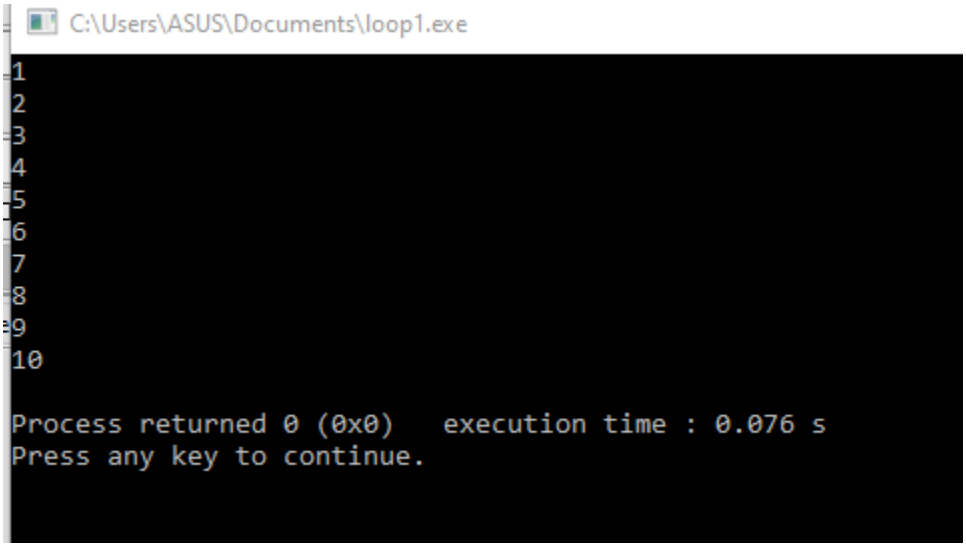


ii.

input:

```
#include <stdio.h>


int main() {
    for (int i = 1; i <= 10; i++) {
        printf("%d\n", i);
    }
    return 0;
}
```

Output:

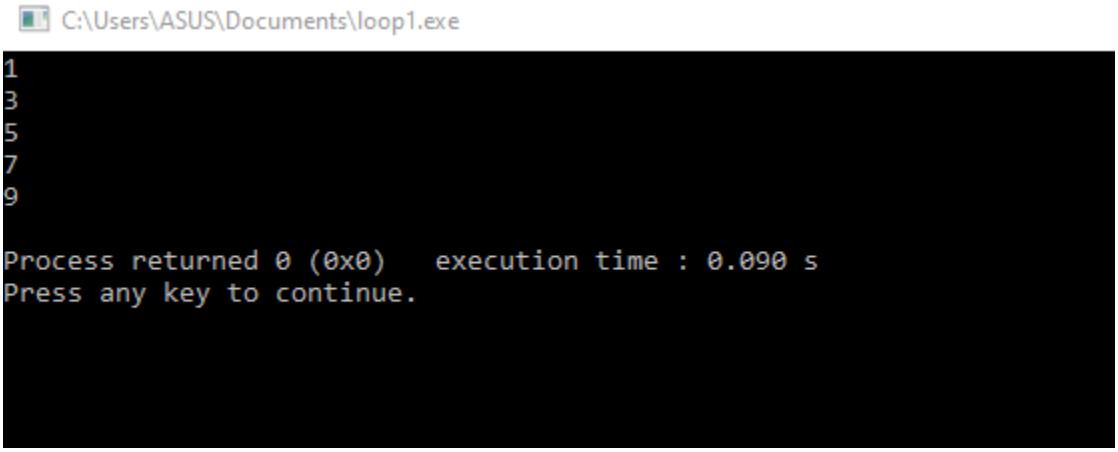


iii.

input:

```
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i += 2) {
        printf("%d\n", i);
    }
    return 0;
}
```

Output:

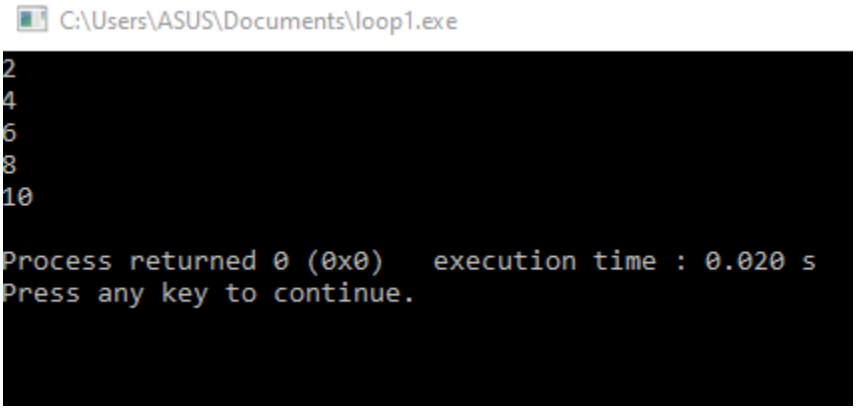


iv.

input:

```
#include <stdio.h>


int main() {
   for (int i = 2; i <= 10; i += 2) {
      printf("%d\n", i);
   }
   return 0;
}
```

Output:



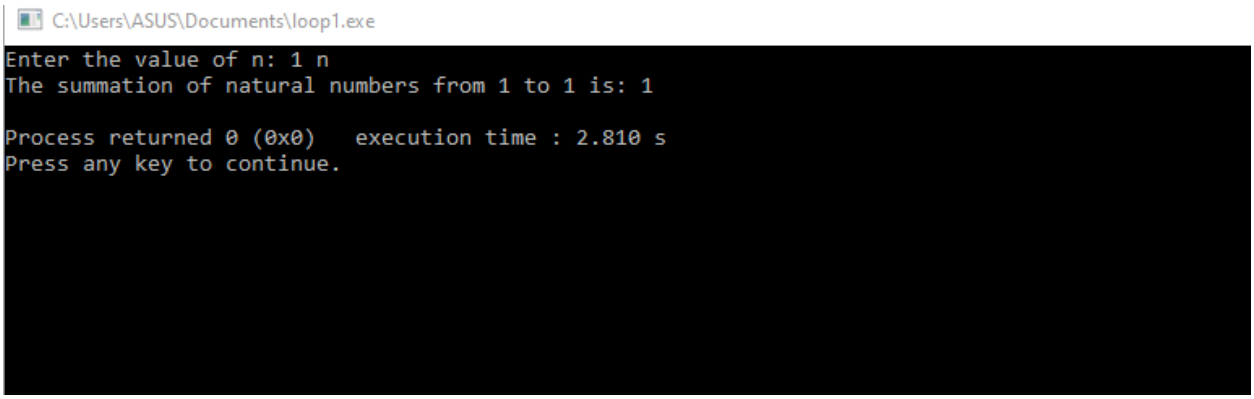
2. Write a C program to print the summation for all natural numbers from 1 to n.

Input:

```
#include <stdio.h>


int main() {
    int n, sum = 0;

    // Input the value of n
    printf("Enter the value of n: ");
    scanf("%d", &n);

    // Calculate the summation
    for (int i = 1; i <= n; i++) {
        sum += i;
```

```
    }

    // Print the result
    printf("The summation of natural numbers from 1 to %d is: %d\n", n, sum);

    return 0;
}
```

Output:



3. Write a C program to print the factorial of n.

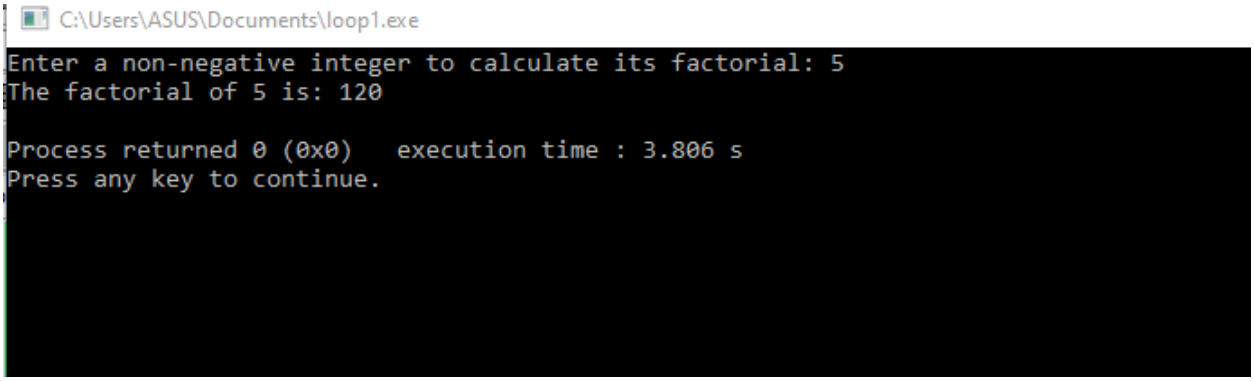
Input:

```
#include <stdio.h>

// Function to calculate factorial
int calculateFactorial(int n) {
```

```
    if (n == 0 || n == 1) {
        return 1;
    } else {
        return n * calculateFactorial(n - 1);
    }
}

int main() {
    int n;

    // Input the value of n
    printf("Enter a non-negative integer to calculate its factorial: ");
    scanf("%d", &n);

    // Check for negative input
    if (n < 0) {
        printf("Factorial is not defined for negative numbers.\n");
    } else {
        // Calculate and print the factorial
        printf("The factorial of %d is: %d\n", n, calculateFactorial(n));
    }
```

```
    return 0;
}
```

Output:



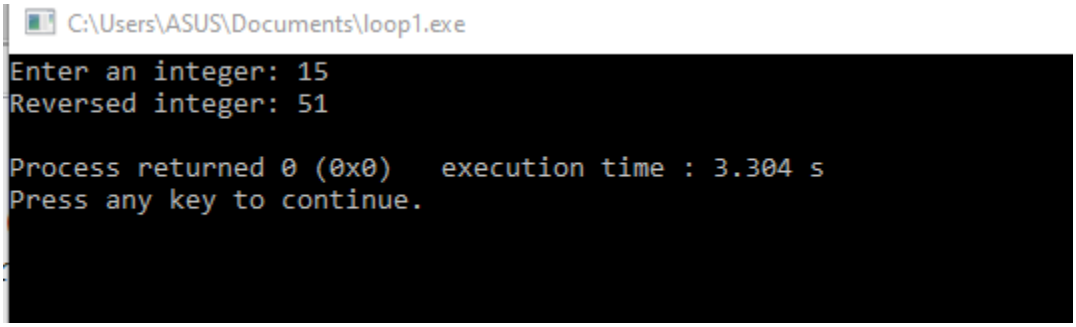
4. Write a C program to reverse an integer number.

Input:

```
#include <stdio.h>

int main() {
    int num, reversedNum = 0, remainder;

    // Input the integer number
    printf("Enter an integer: ");
    scanf("%d", &num);

    // Reverse the integer
    while (num != 0) {
```

```
        remainder = num % 10;
        reversedNum = reversedNum * 10 + remainder;
        num /= 10;
    }

    // Print the reversed integer
    printf("Reversed integer: %d\n", reversedNum);

    return 0;
}
```

Output:



5. Write a C program to print the GCD and LCM of 2 integer numbers.

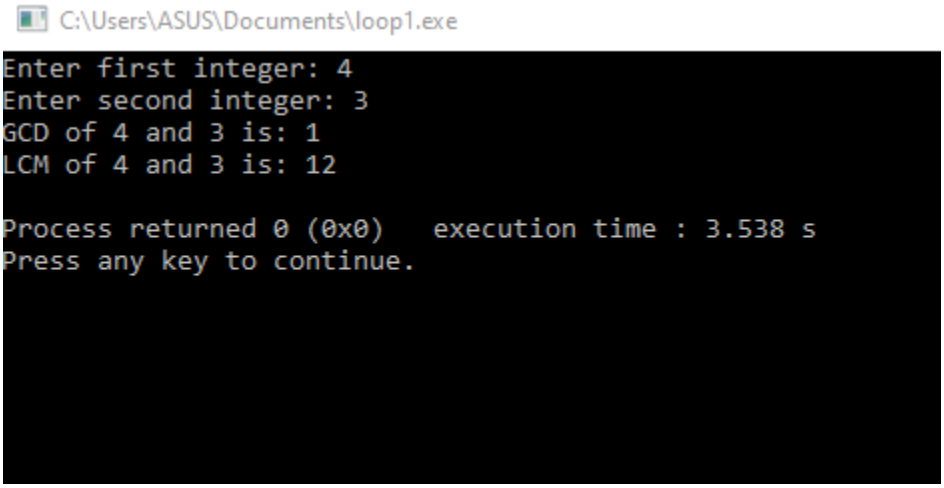
Input:

```
#include <stdio.h>

// Function to calculate GCD using Euclidean Algorithm
```

```
int calculateGCD(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

// Function to calculate LCM using GCD
int calculateLCM(int a, int b) {
    int gcd = calculateGCD(a, b);
    int lcm = (a * b) / gcd;
    return lcm;
}

int main() {
    int num1, num2;

    // Input the two integer numbers
    printf("Enter first integer: ");
    scanf("%d", &num1);
```

```
    printf("Enter second integer: ");
    scanf("%d", &num2);

    // Calculate and print the GCD
    printf("GCD of %d and %d is: %d\n", num1, num2,
calculateGCD(num1, num2));

    // Calculate and print the LCM
    printf("LCM of %d and %d is: %d\n", num1, num2,
calculateLCM(num1, num2));

    return 0;
}
```

Output:



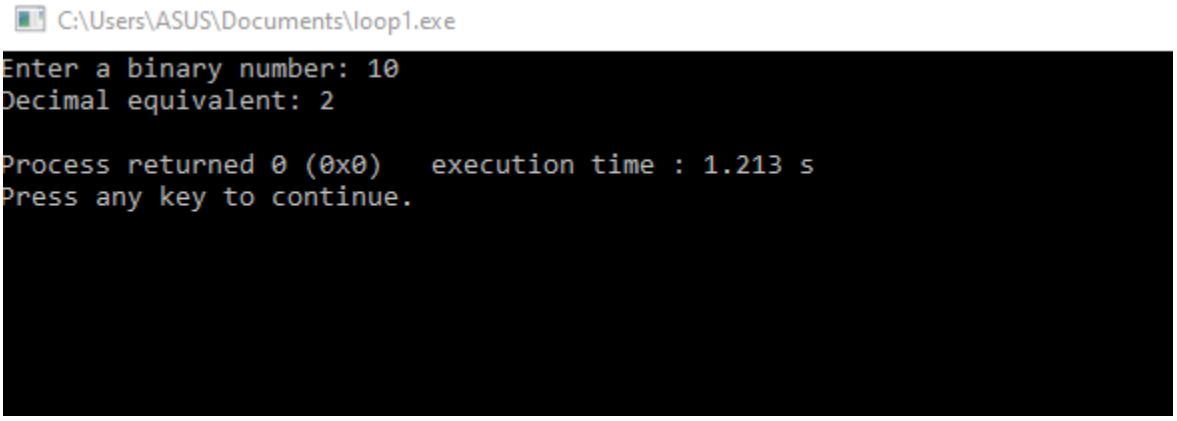
6. Write a C program to convert a binary value into a decimal value.

Input:

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

// Function to convert binary to decimal
int binaryToDecimal(long long binary) {
    int decimal = 0, i = 0, remainder;

    // Iterate through each binary digit
    while (binary != 0) {
        remainder = binary % 10;
        binary /= 10;
        decimal += remainder * pow(2, i);
        ++i;
    }

    return decimal;
}

int main() {
    long long binary;
```

```
    // Input the binary number
    printf("Enter a binary number: ");
    scanf("%lld", &binary);

    // Check if the input is a binary number
    long long temp = binary;
    while (temp != 0) {
        if (temp % 10 != 0 && temp % 10 != 1) {
            printf("Invalid binary number. Please enter a valid binary number.\n");
            return 1; // Exit the program with an error code
        }
        temp /= 10;
    }

    // Convert and print the decimal value
    printf("Decimal equivalent: %d\n", binaryToDecimal(binary));

    return 0;
}
```

Output:



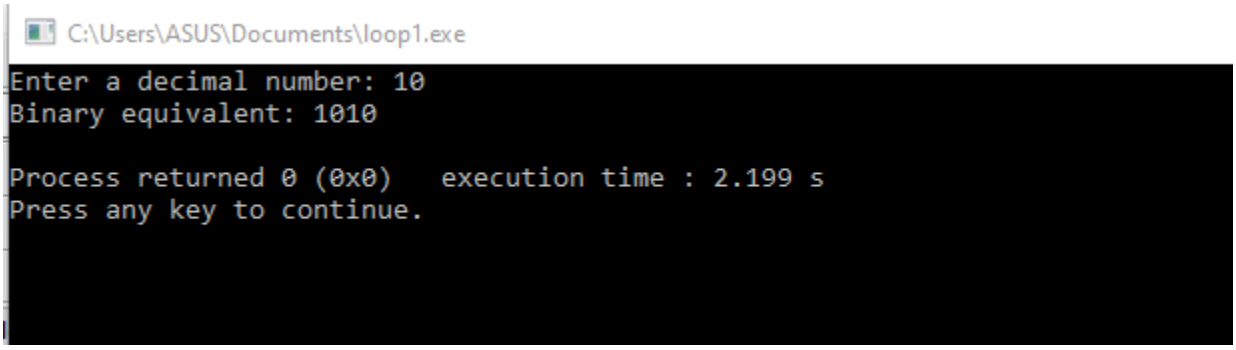
7. Write a C program to convert a decimal value into a binary value.

Input:

```
#include <stdio.h>

// Function to convert decimal to binary
long long decimalToBinary(int decimal) {
    long long binary = 0;
    int remainder, place = 1;

    // Iterate until the decimal becomes 0
    while (decimal > 0) {
        remainder = decimal % 2;
        binary += remainder * place;
        decimal /= 2;
```

```
        place *= 10;
    }

    return binary;
}

int main() {
    int decimal;

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

    // Convert and print the binary value
    printf("Binary equivalent: %lld\n", decimalToBinary(decimal));

    return 0;
}
```

Output:



8. Find out the sum of each of the following series. n is the input from the user for series (i) to (viii)

i. 1+2+3+ … +100

ii. 3 + 11 + 19 + … + 1691

iii. 7 + 20 + 33 + … (up to 100th term)

iv. 5 – 11 + 17 - … (up to 75th term)

v. 1 + (1 + 2) + (1 + 2 + 3) + … + (1 + 2 + 3 + … + n)

vi. 1 + 2 ^2/2! + 3^2/3! + … + n^ n/2!

vii. 1 * 2 * 3 * … * 100

viii. 2 * 7 * 12 * … * 37

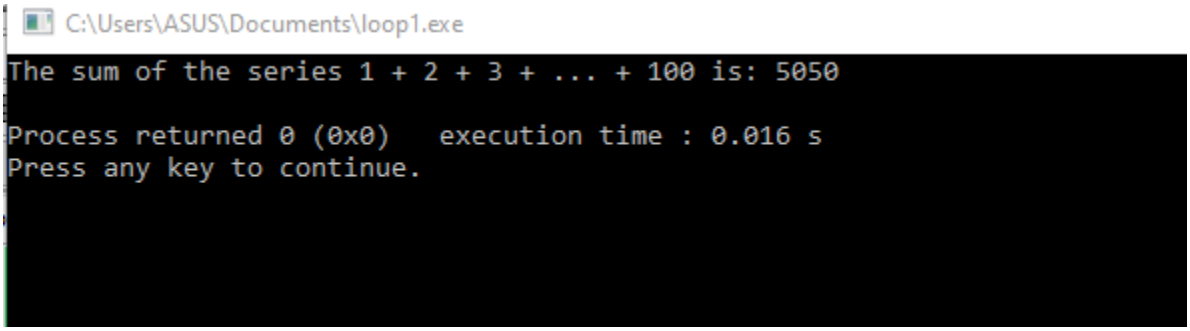
Solution:

i.

Input:

```
#include <stdio.h>

int main() {
    int n = 100; // The upper limit of the series
    int sum = 0;

    // Calculate the sum of the series
    for (int i = 1; i <= n; i++) {
        sum += i;
    }

    // Print the result
    printf("The sum of the series 1 + 2 + 3 + ... + 100 is: %d\n", sum);

    return 0;
}
```

Output:



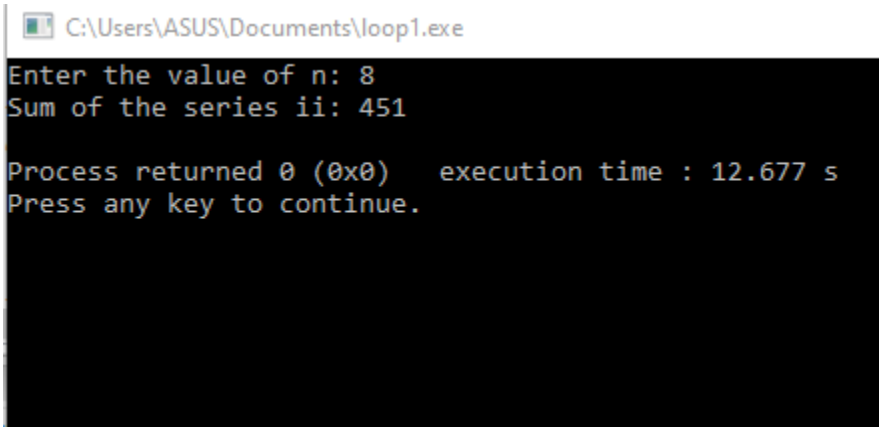
ii.

Input:

```
#include <stdio.h>

// Function to calculate the sum of the series ii. 3 + 11 + 19 + ... + 1691
int series2(int n) {
    return 3 + 8 * (n - 1) * n;
}

int main() {
    int n;

    // Input the value of n
    printf("Enter the value of n: ");
    scanf("%d", &n);

    // Calculate and print the sum for the series ii
```

```
    printf("Sum of the series ii: %d\n", series2(n));

    return 0;
}
```

Output:



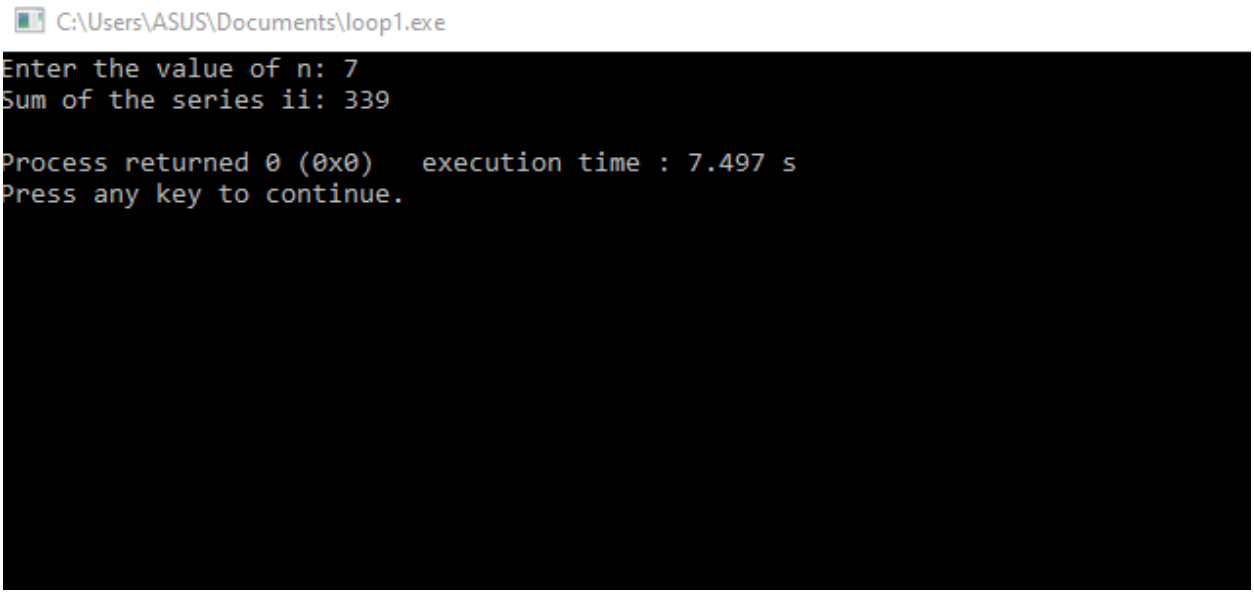
iii.

Input:

```
#include <stdio.h>

// Function to calculate the sum of the series ii. 3 + 11 + 19 + ... + 1691
int series2(int n) {
    return 3 + 8 * (n - 1) * n;
}
```

```
int main() {
    int n;

    // Input the value of n
    printf("Enter the value of n: ");
    scanf("%d", &n);

    // Calculate and print the sum for the series ii
    printf("Sum of the series ii: %d\n", series2(n));

    return 0;
}
```

Output:

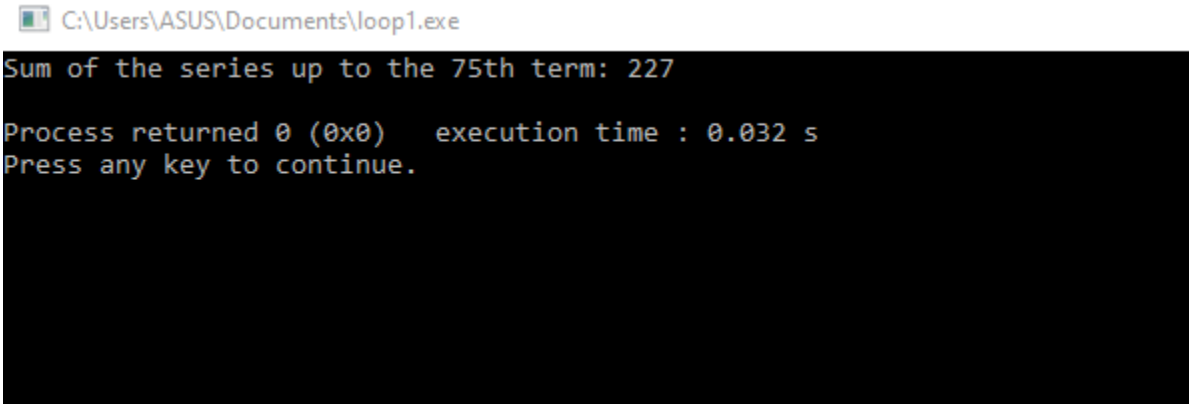


iv.

Input:

```
#include <stdio.h>

int main() {
   int terms = 75;  // Number of terms in the series
   int sum = 0;     // Variable to store the sum
   int sign = 1;    // Variable to alternate the sign (+1 or -1)
   int term = 5;    // Starting term of the series

   for (int i = 1; i <= terms; ++i) {
      sum += sign * term;  // Add the term to the sum with the
appropriate sign
      sign = -sign;        // Alternate the sign for the next term
      term += 6;           // Increment the term by 6 for the next term in the
series
   }

   // Display the sum
   printf("Sum of the series up to the %dth term: %d\n", terms, sum);

   return 0;
}
```

Output:



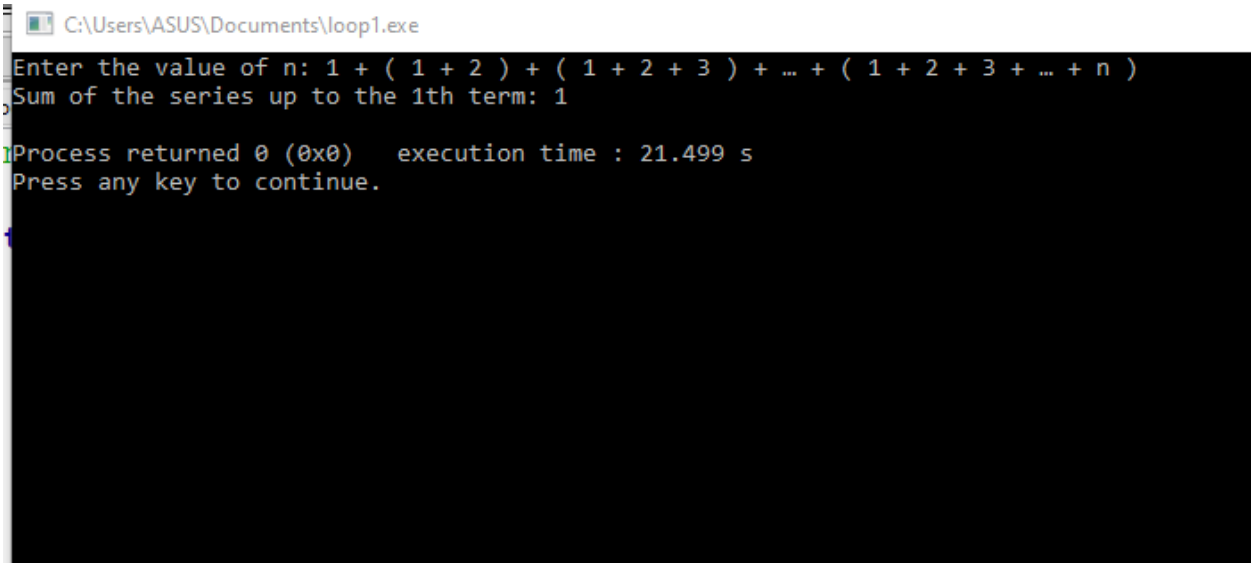
v.

Input:

```
#include <stdio.h>


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


   int sum = 0;  // Variable to store the sum


   for (int i = 1; i <= n; ++i) {
      int innerSum = 0;  // Variable to store the sum of inner series (1 + 2 + 3 + ... + i)
```

```
        for (int j = 1; j <= i; ++j) {
            innerSum += j;
        }

        sum += innerSum;  // Add the inner sum to the overall sum
    }

    // Display the sum
    printf("Sum of the series up to the %dth term: %d\n", n, sum);

    return 0;
}
```

Output:



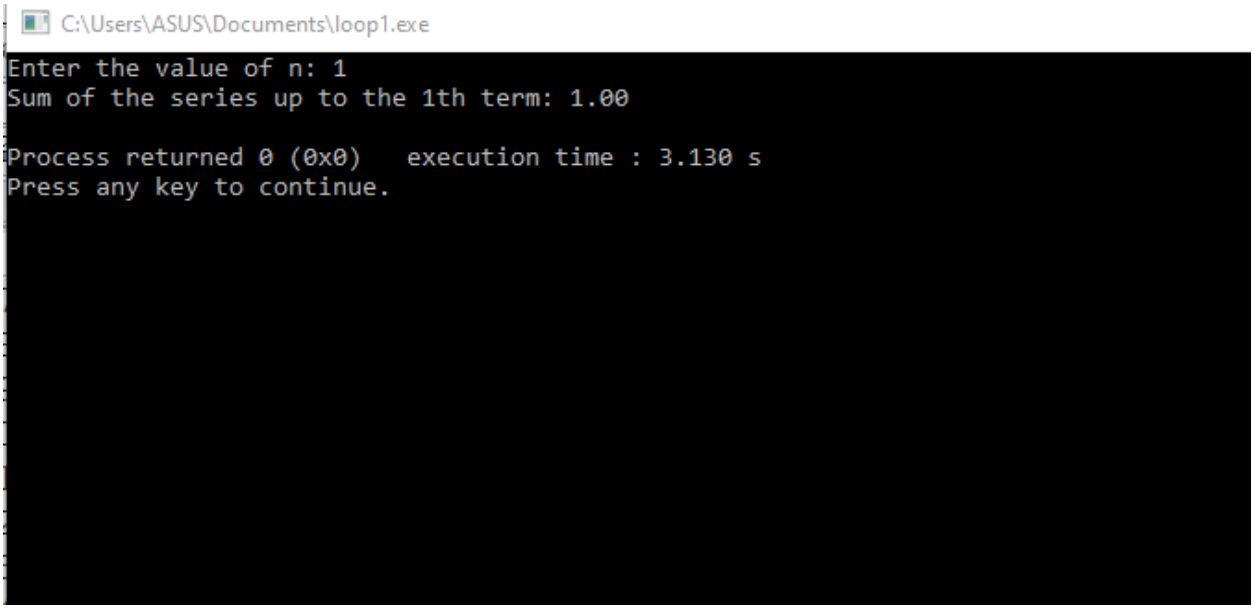
vi.

input:

```
#include <stdio.h>

// Function to calculate the factorial of a number
int factorial(int num) {
    if (num == 0 || num == 1) {
        return 1;
    } else {
        return num * factorial(num - 1);
    }
}

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

    double sum = 0.0;  // Variable to store the sum

    for (int i = 1; i <= n; ++i) {
        double term = (double)factorial(i) / (i * i);  // Calculate each term
```

```
        sum += term;  // Add the term to the sum
    }

    // Display the sum
    printf("Sum of the series up to the %dth term: %.2f\n", n, sum);

    return 0;
}
```

Output:

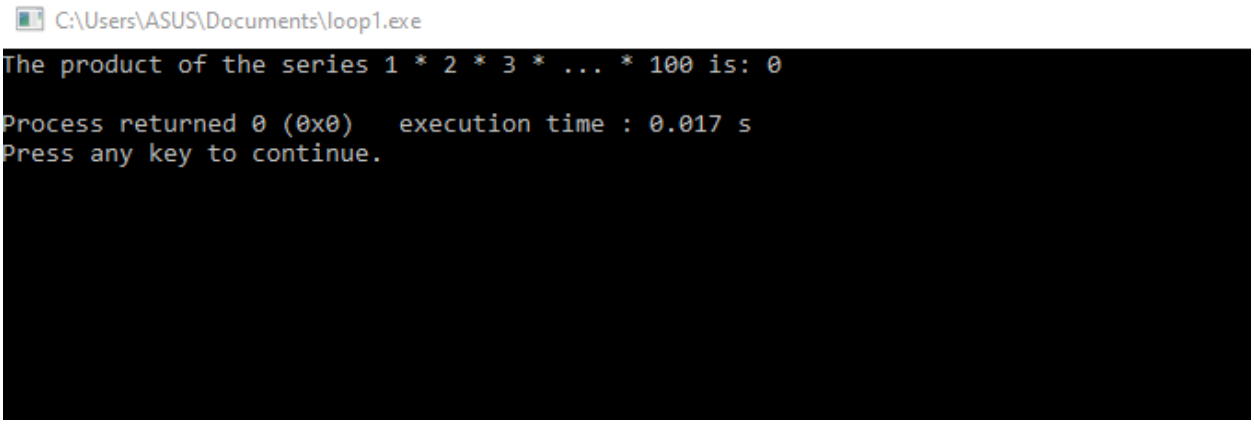


vii.

Input:

```
#include <stdio.h>

int main() {
   int n = 100;  // Number up to which the series is calculated
   long long product = 1;  // Variable to store the product

   for (int i = 1; i <= n; ++i) {
      product *= i;  // Multiply the current number to the product
   }

   // Display the result
   printf("The product of the series 1 * 2 * 3 * ... * %d is: %lld\n", n,
product);

   return 0;
}
```

Output:



viii.

Input:

```
#include <stdio.h>

int main() {
    int start = 2;   // Starting term of the series
    int end = 37;    // Ending term of the series
    int step = 5;    // Step between terms in the series
    long long product = 1;  // Variable to store the product

    for (int i = start; i <= end; i += step) {
        product *= i;  // Multiply the current term to the product
    }
```

```
    // Display the result
    printf("The product of the series 2 * 7 * 12 * ... * 37 is: %lld\n", product);


    return 0;
}
```

Output: