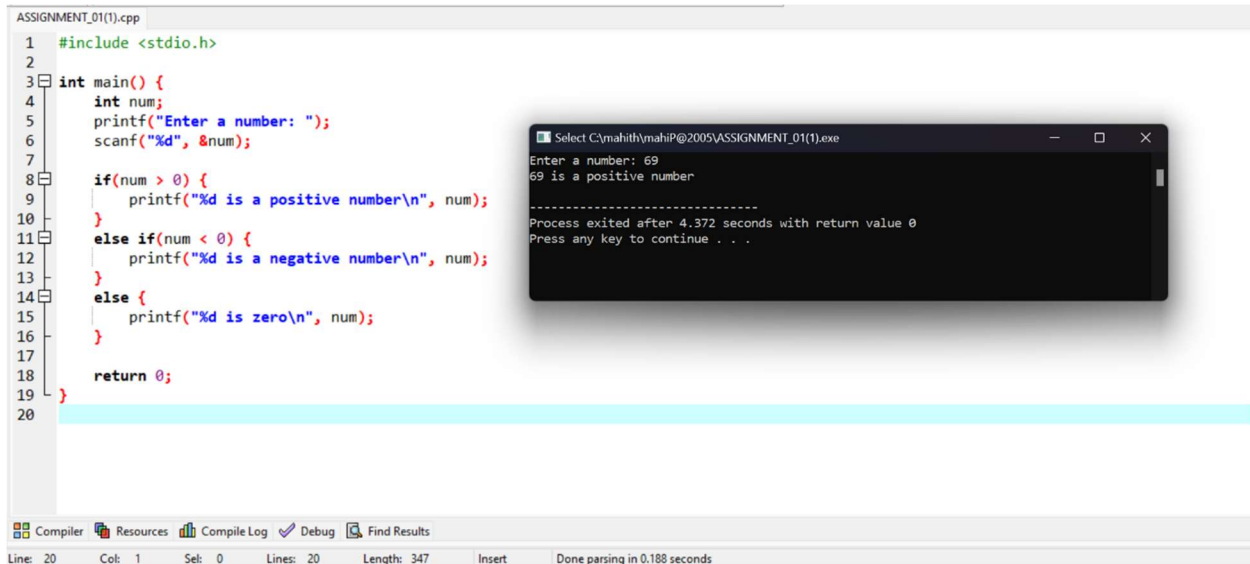


Assignment – 1 (Control Statements and Loops)

1. Write a C program to check positive, negative or zero using simple if or if else. C program to input any number from user and check whether the given number is positive, negative or zero. Logic to check negative, positive or zero in C programming.

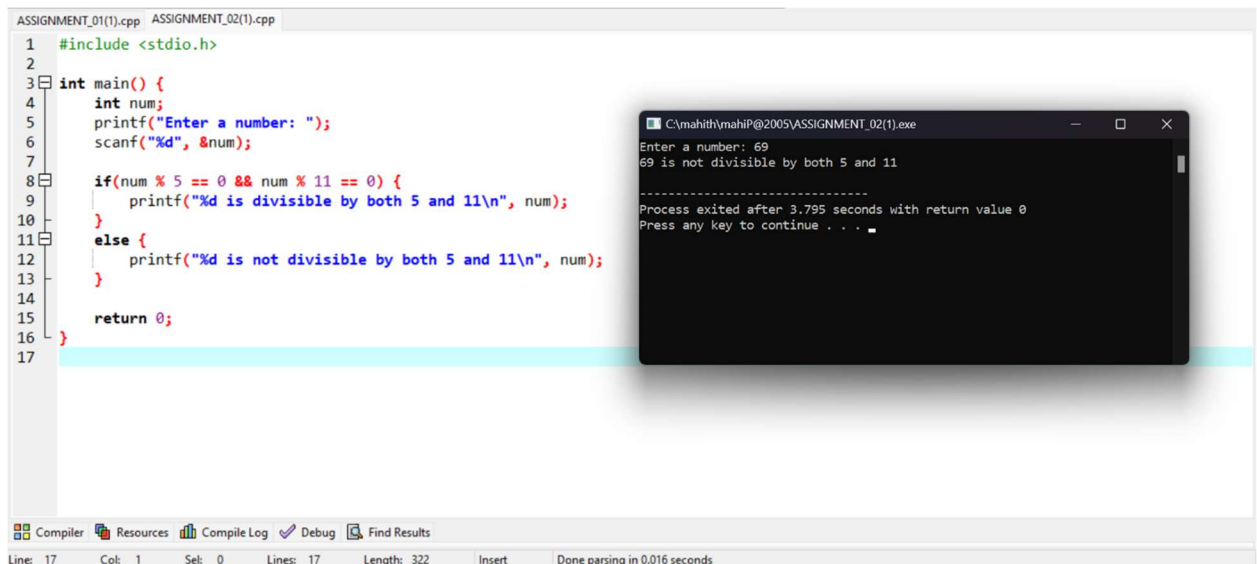


The screenshot shows a C program in a code editor and its execution output in a terminal window. The code is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int num;
5     printf("Enter a number: ");
6     scanf("%d", &num);
7
8     if(num > 0) {
9         printf("%d is a positive number\n", num);
10    }
11    else if(num < 0) {
12        printf("%d is a negative number\n", num);
13    }
14    else {
15        printf("%d is zero\n", num);
16    }
17
18    return 0;
19 }
20
```

The terminal window shows the execution of the program. It prompts the user to enter a number, and the user enters 69. The program outputs "69 is a positive number". The process exits after 4.372 seconds with a return value of 0.

2. Write a C program to check whether a number is divisible by 5 and 11 or not using if else. How to check divisibility of any number in C programming. C program to enter any number and check whether it is divisible by 5 and 11 or not. Logic to check divisibility of a number in C program.

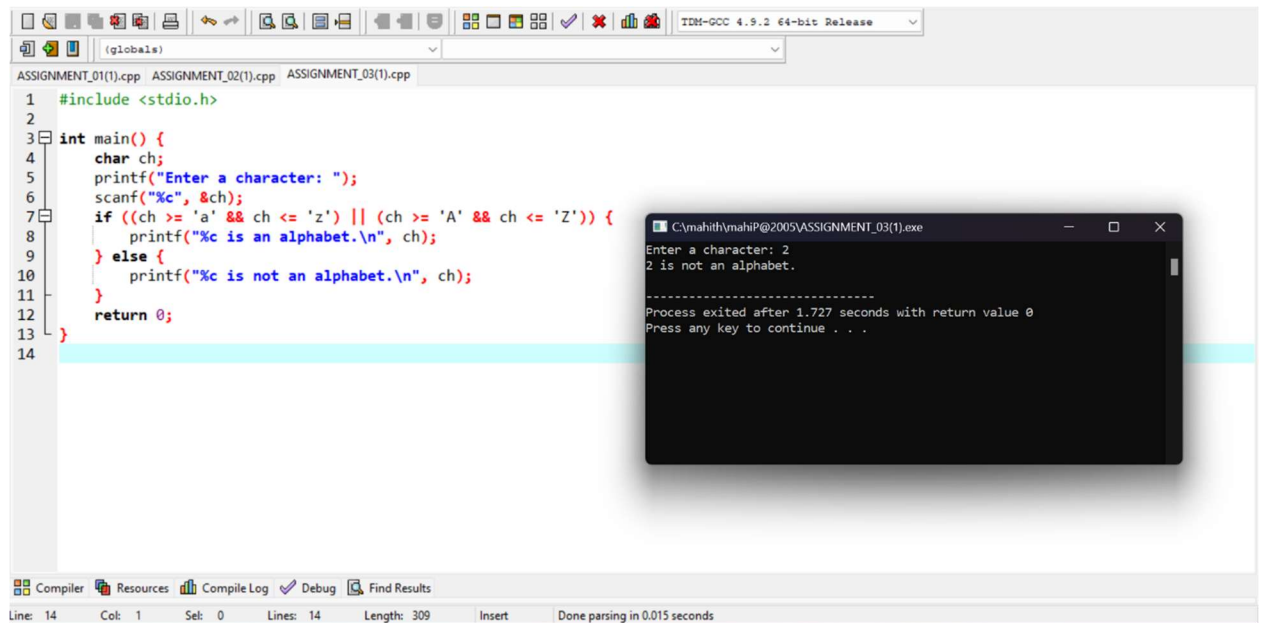


The screenshot shows a C program in a code editor and its execution output in a terminal window. The code is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int num;
5     printf("Enter a number: ");
6     scanf("%d", &num);
7
8     if(num % 5 == 0 && num % 11 == 0) {
9         printf("%d is divisible by both 5 and 11\n", num);
10    }
11    else {
12        printf("%d is not divisible by both 5 and 11\n", num);
13    }
14
15    return 0;
16 }
17
```

The terminal window shows the execution of the program. It prompts the user to enter a number, and the user enters 69. The program outputs "69 is not divisible by both 5 and 11". The process exits after 3.795 seconds with a return value of 0.

3. Write a C program to input a character from user and check whether the given character is alphabet or not using if else. How to check whether a character is alphabet or not in C programming. Logic to check if a character is alphabet or not in C program.

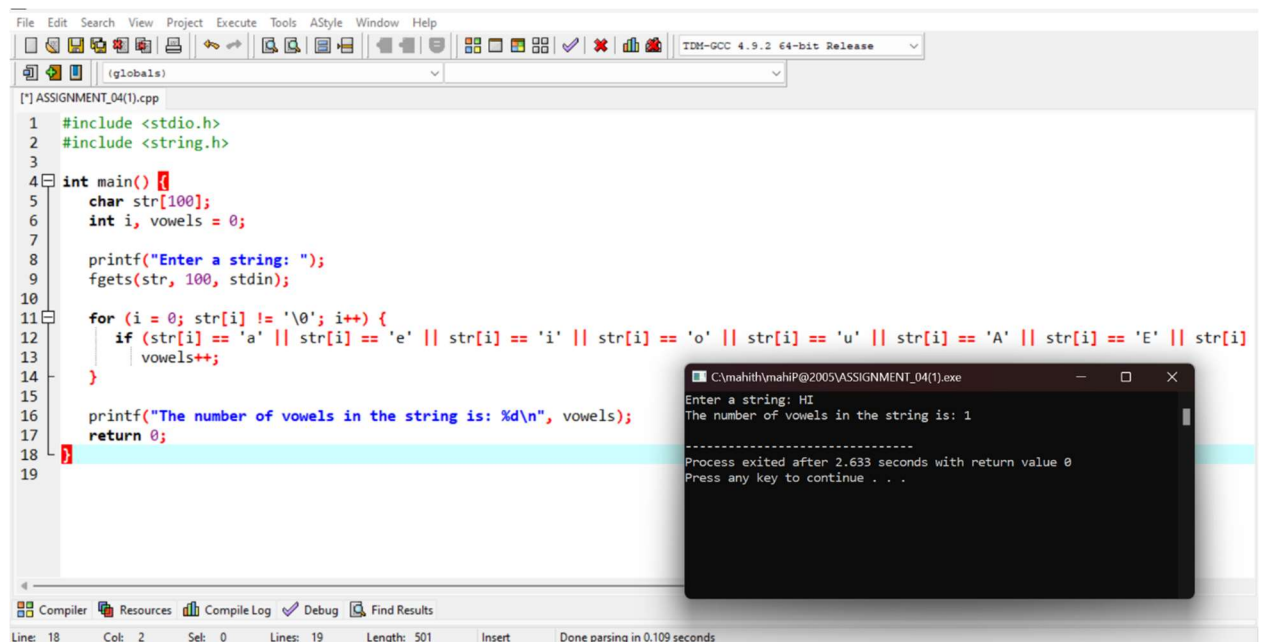


The screenshot shows a C program in a code editor. The program includes `<stdio.h>` and defines a `main` function. It prompts the user to "Enter a character:" and reads a character `ch` using `scanf`. It then uses an `if-else` statement to check if `ch` is an alphabet (between 'a' and 'z' or 'A' and 'Z'). If it is, it prints "`%c is an alphabet.\n`"; otherwise, it prints "`%c is not an alphabet.\n`". The program returns 0.

```
1 #include <stdio.h>
2
3 int main() {
4     char ch;
5     printf("Enter a character: ");
6     scanf("%c", &ch);
7     if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {
8         printf("%c is an alphabet.\n", ch);
9     } else {
10        printf("%c is not an alphabet.\n", ch);
11    }
12    return 0;
13 }
14
```

The output window shows the execution: "Enter a character: 2", "2 is not an alphabet.", and "Process exited after 1.727 seconds with return value 0".

4. Write a C program to count the Vowels in the given string.

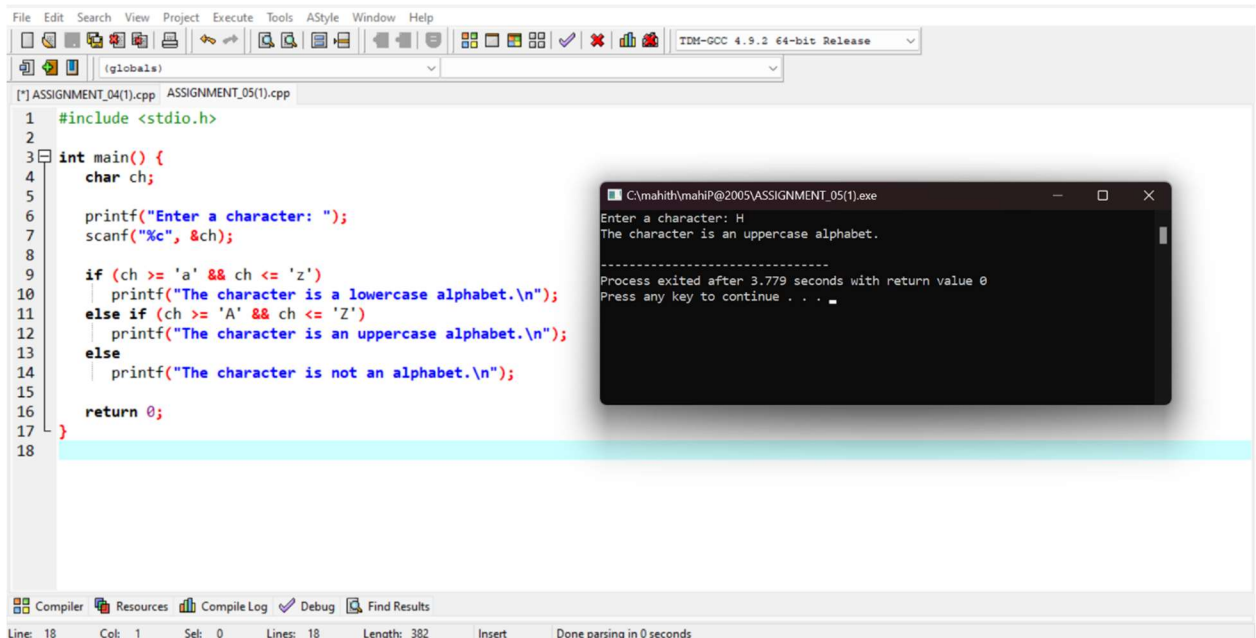


The screenshot shows a C program in a code editor. The program includes `<stdio.h>` and `<string.h>`. It defines a `main` function that declares a character array `str` of size 100 and an integer `vowels` set to 0. It prompts the user to "Enter a string:" and reads the string using `fgets`. It then uses a `for` loop to iterate through the string, checking each character against a set of vowels ('a', 'e', 'i', 'o', 'u' and their uppercase counterparts). If a vowel is found, `vowels` is incremented. Finally, it prints "The number of vowels in the string is: `%d`\n", `vowels`" and returns 0.

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5     char str[100];
6     int i, vowels = 0;
7
8     printf("Enter a string: ");
9     fgets(str, 100, stdin);
10
11     for (i = 0; str[i] != '\0'; i++) {
12         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') {
13             vowels++;
14         }
15     }
16     printf("The number of vowels in the string is: %d\n", vowels);
17     return 0;
18 }
19
```

The output window shows the execution: "Enter a string: HI", "The number of vowels in the string is: 1", and "Process exited after 2.633 seconds with return value 0".

5. Write a C program to input character from user and check whether character is uppercase or lowercase alphabet using if else. How to check uppercase and lowercase using if else in C programming. Logic to check uppercase and lowercase alphabets in C program.



The screenshot shows a C program in a code editor and its execution output in a terminal window. The program prompts the user to enter a character and checks if it is an uppercase or lowercase alphabet. The output shows that the character 'H' is an uppercase alphabet.

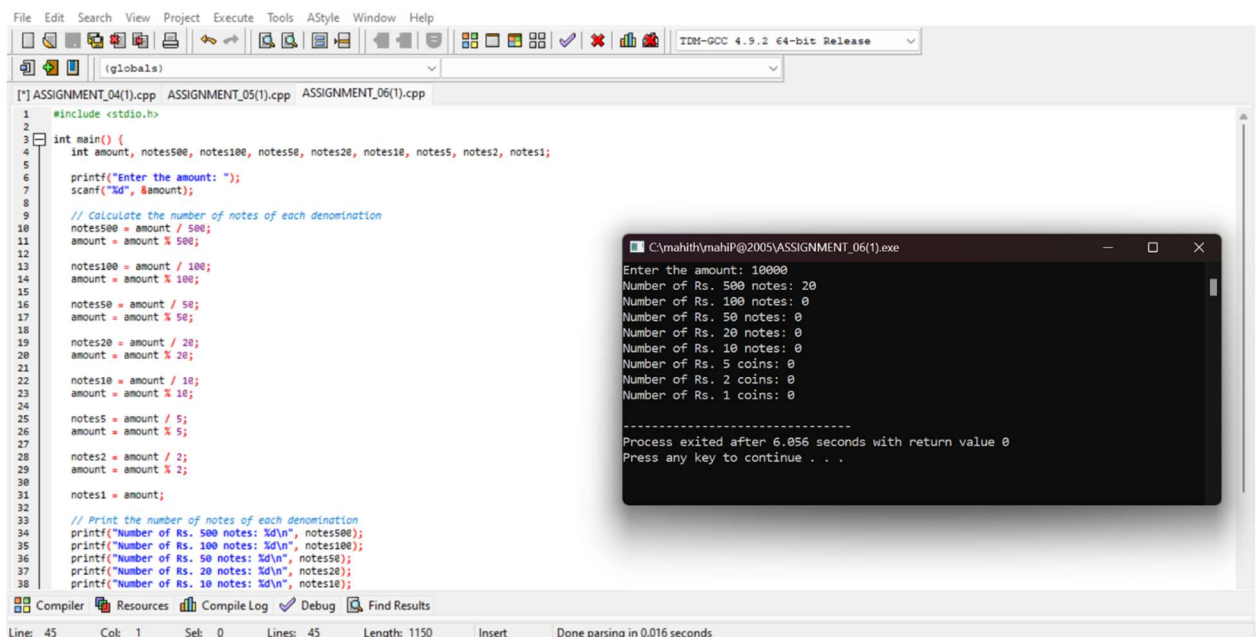
```
1 #include <stdio.h>
2
3 int main() {
4     char ch;
5
6     printf("Enter a character: ");
7     scanf("%c", &ch);
8
9     if (ch >= 'a' && ch <= 'z')
10        printf("The character is a lowercase alphabet.\n");
11    else if (ch >= 'A' && ch <= 'Z')
12        printf("The character is an uppercase alphabet.\n");
13    else
14        printf("The character is not an alphabet.\n");
15
16    return 0;
17 }
18
```

Execution Output:

```
C:\mahith\mahip@2005\ASSIGNMENT_05(1).exe
Enter a character: H
The character is an uppercase alphabet.

Process exited after 3.779 seconds with return value 0
Press any key to continue . . .
```

6. Write a C program to input amount from user and print minimum number of notes (Rs. 500, 100, 50, 20, 10, 5, 2, 1) required for the amount. How to the minimum number of notes required for the given amount in C programming. Program to find minimum number of notes required for the given denomination. Logic to find minimum number of denomination for a given amount in C program.



The screenshot shows a C program in a code editor and its execution output in a terminal window. The program prompts the user to enter an amount and calculates the minimum number of notes required for that amount using denominations of 500, 100, 50, 20, 10, 5, 2, and 1. The output shows the minimum number of notes required for an amount of 10000.

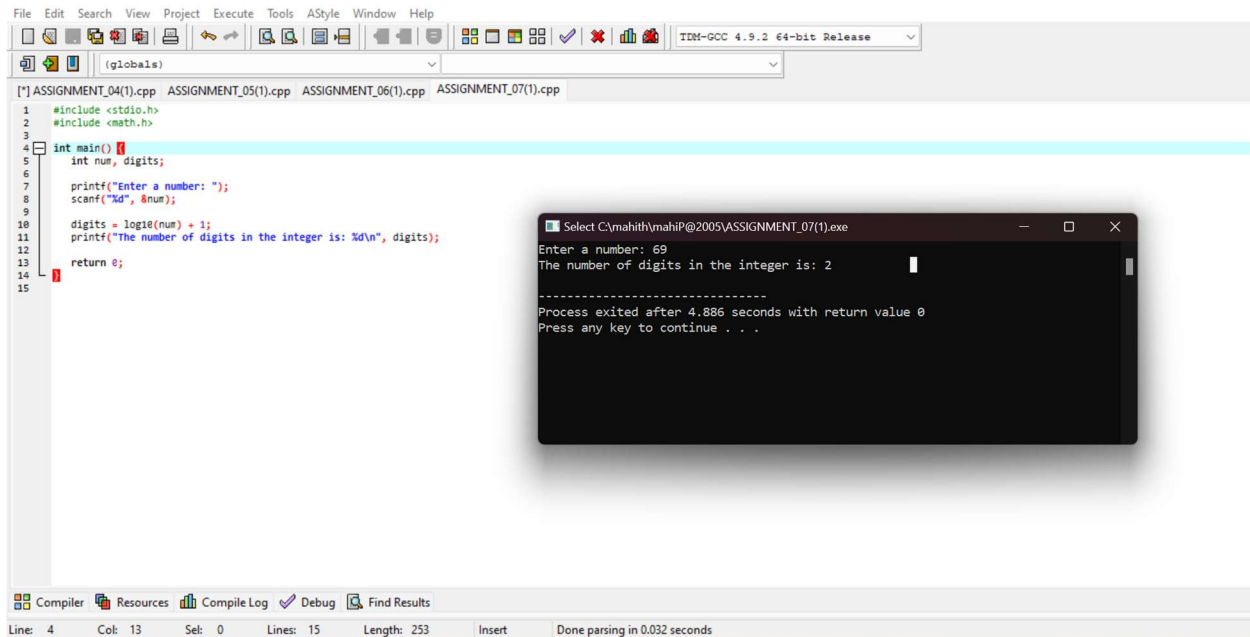
```
1 #include <stdio.h>
2
3 int main() {
4     int amount, notes500, notes100, notes50, notes20, notes10, notes5, notes2, notes1;
5
6     printf("Enter the amount: ");
7     scanf("%d", &amount);
8
9     // Calculate the number of notes of each denomination
10    notes500 = amount / 500;
11    amount = amount % 500;
12
13    notes100 = amount / 100;
14    amount = amount % 100;
15
16    notes50 = amount / 50;
17    amount = amount % 50;
18
19    notes20 = amount / 20;
20    amount = amount % 20;
21
22    notes10 = amount / 10;
23    amount = amount % 10;
24
25    notes5 = amount / 5;
26    amount = amount % 5;
27
28    notes2 = amount / 2;
29    amount = amount % 2;
30
31    notes1 = amount;
32
33    // Print the number of notes of each denomination
34    printf("Number of Rs. 500 notes: %d\n", notes500);
35    printf("Number of Rs. 100 notes: %d\n", notes100);
36    printf("Number of Rs. 50 notes: %d\n", notes50);
37    printf("Number of Rs. 20 notes: %d\n", notes20);
38    printf("Number of Rs. 10 notes: %d\n", notes10);
39
40    return 0;
41 }
```

Execution Output:

```
C:\mahith\mahip@2005\ASSIGNMENT_06(1).exe
Enter the amount: 10000
Number of Rs. 500 notes: 20
Number of Rs. 100 notes: 0
Number of Rs. 50 notes: 0
Number of Rs. 20 notes: 0
Number of Rs. 10 notes: 0
Number of Rs. 5 coins: 0
Number of Rs. 2 coins: 0
Number of Rs. 1 coins: 0

Process exited after 6.056 seconds with return value 0
Press any key to continue . . .
```

7. Write a C program to input a number from user and count number of digits in the given integer using loop. How to find total digits in a given integer using loop in C programming. Logic to count digits in a given integer without using loop in C program.

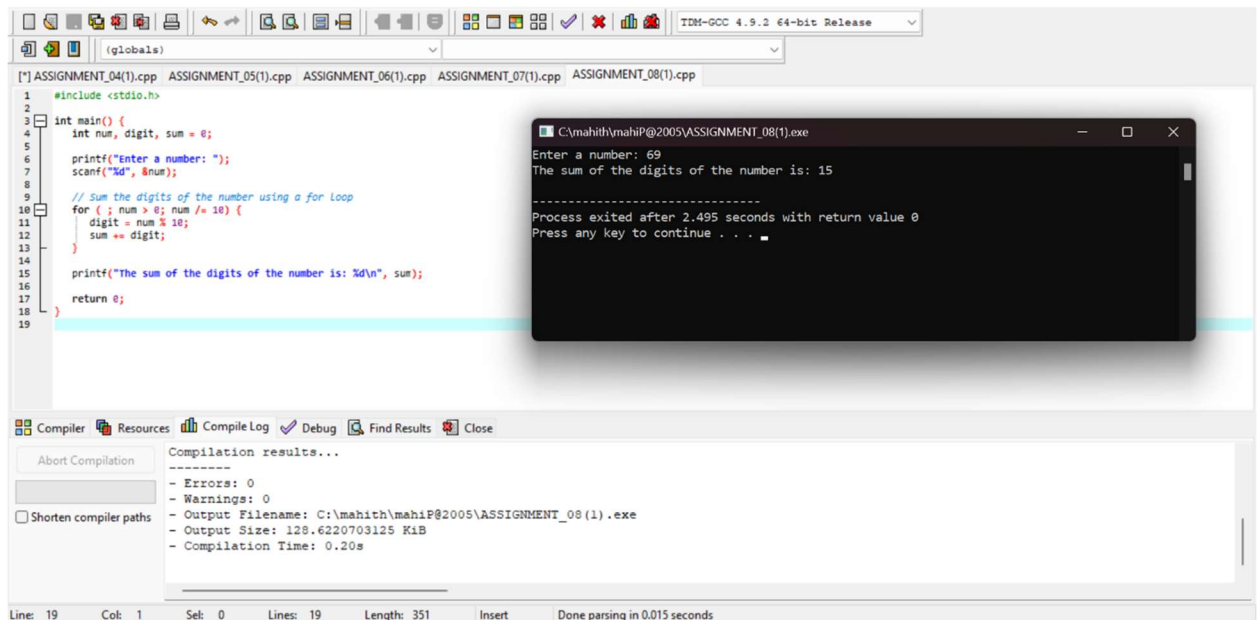


```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     int num, digits;
6
7     printf("Enter a number: ");
8     scanf("%d", &num);
9
10    digits = log10(num) + 1;
11    printf("The number of digits in the integer is: %d\n", digits);
12
13    return 0;
14 }
15
```

Output window: C:\mahith\mahip@2005\ASSIGNMENT_07(1).exe

```
Enter a number: 69
The number of digits in the integer is: 2
-----
Process exited after 4.886 seconds with return value 0
Press any key to continue . . .
```

8. Write a C program to input a number and calculate sum of digits using for loop. How to find sum of digits of a number in C program. Logic to find sum of digits of a given number in C programming.



```
1 #include <stdio.h>
2
3 int main() {
4     int num, digit, sum = 0;
5
6     printf("Enter a number: ");
7     scanf("%d", &num);
8
9     // Sum the digits of the number using a for loop
10    for (; num > 0; num /= 10) {
11        digit = num % 10;
12        sum += digit;
13    }
14
15    printf("The sum of the digits of the number is: %d\n", sum);
16
17    return 0;
18 }
19
```

Output window: C:\mahith\mahip@2005\ASSIGNMENT_08(1).exe

```
Enter a number: 69
The sum of the digits of the number is: 15
-----
Process exited after 2.495 seconds with return value 0
Press any key to continue . . .
```

Compilation results...

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\mahith\mahip@2005\ASSIGNMENT_08(1).exe
- Output Size: 120.6220703125 KiB
- Compilation Time: 0.20s
```

9. Write a C program to input a number from user and find reverse of the given number using for loop. How to find reverse of any number using loop in C program. Logic to find reverse of a number in C programming.

The screenshot shows a C program in a code editor. The program prompts the user to enter a number, reads it, and then uses a for loop to calculate its reverse. The output window shows the input 69 and the reverse 96. The compilation results show no errors or warnings.

```
1 #include <stdio.h>
2
3 int main() {
4     int num, reversed = 0;
5
6     printf("Enter a number: ");
7     scanf("%d", &num);
8
9     // Reverse the number using a for loop
10    for (; num != 0; num /= 10) {
11        reversed = reversed * 10 + num % 10;
12    }
13
14    printf("The reverse of the number is: %d\n", reversed);
15
16    return 0;
17 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\mahith\mahitP@2005\ASSIGNMENT_09(1).exe
- Output Size: 128.1220703125 KiB
- Compilation Time: 0.20s

10. Write a C program to input decimal number from user and convert to binary number system. How to convert from decimal number to binary number system in C program. Logic to convert decimal to binary number system in C programming.

The screenshot shows a C program in a code editor. The program prompts the user to enter a decimal number, reads it, and then uses a while loop to convert it to binary. The output window shows the input 0.69 and the binary equivalent 0. The compilation results show no errors or warnings.

```
1 #include <stdio.h>
2
3 int main() {
4     int decimal, binary = 0, base = 1;
5
6     printf("Enter a decimal number: ");
7     scanf("%d", &decimal);
8
9     // Convert decimal to binary
10    while (decimal != 0) {
11        binary += (decimal % 2) * base;
12        decimal /= 2;
13        base *= 10;
14    }
15
16    printf("The binary equivalent is: %d\n", binary);
17
18    return 0;
19 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\mahith\mahitP@2005\ASSIGNMENT_10(1).exe
- Output Size: 128.1220703125 KiB
- Compilation Time: 0.20s

