



Chandpur Science and Technology University

Department of Computer Science and Engineering

LAB ASSIGNMENT #3



C Lab Assignment Submitted By:

Name: Iftekhar Hossain

Student ID: B220101024

Submitted To:

**Prince Mahmud
Lecturer**

Department of Computer Science and Engineering, CSTU

Lab Date:

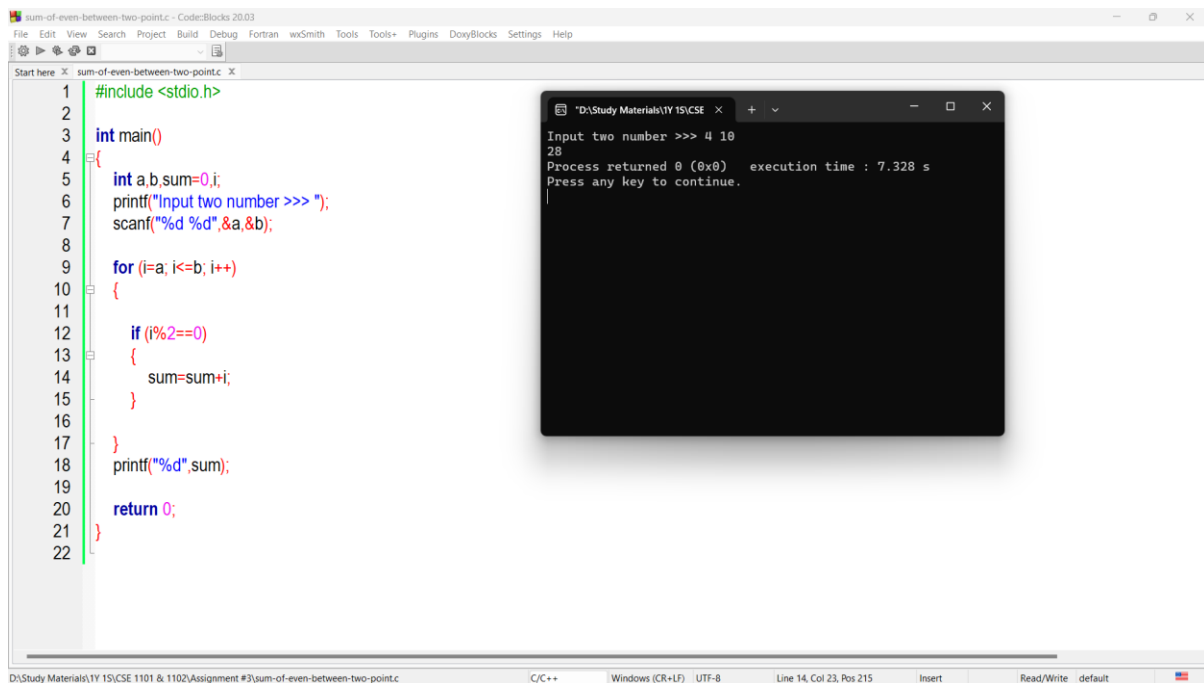
Submission Date: 10/12/2023

Marks & Signature

--

Problem No – 1

Write a program to input two integer numbers and display the sum of even numbers between these two input numbers.



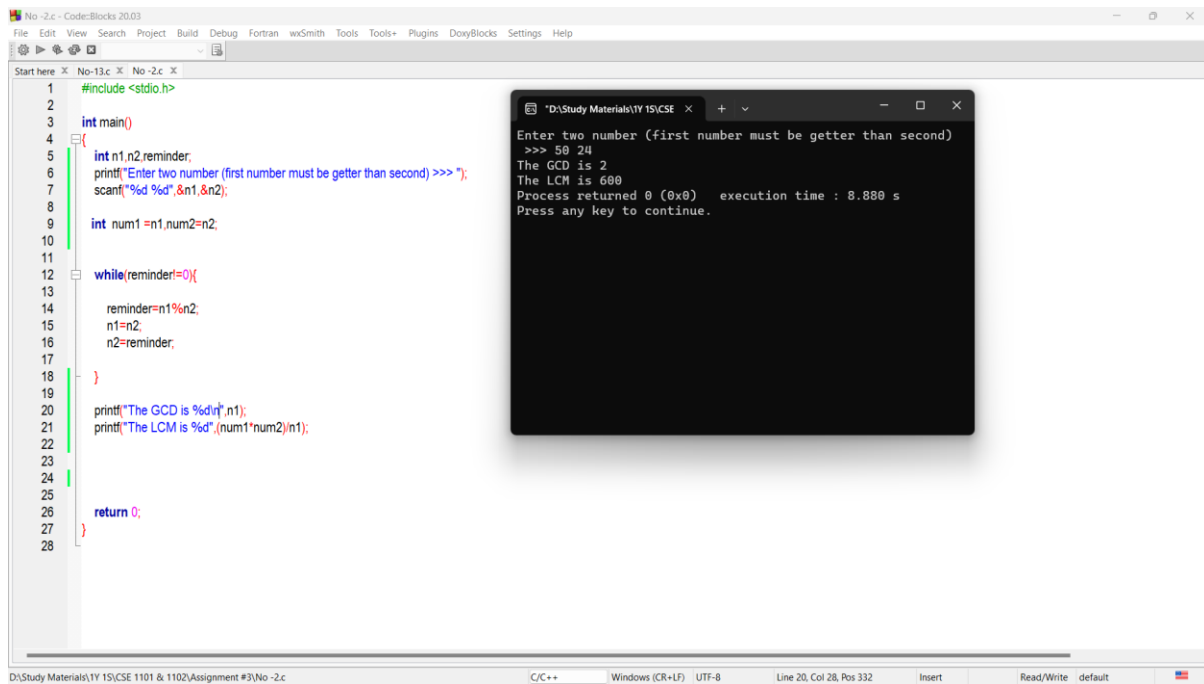
```
1 #include <stdio.h>
2
3 int main()
4 {
5     int a,b,sum=0,i;
6     printf("Input two number >>> ");
7     scanf("%d %d",&a,&b);
8
9     for (i=a; i<=b; i++)
10    {
11
12        if (i%2==0)
13        {
14            sum=sum+i;
15        }
16    }
17    printf("%d",sum);
18
19    return 0;
20 }
21
22
```

Terminal Output:

```
Input two number >>> 4 10
28
Process returned 0 (0x0)   execution time : 7.328 s
Press any key to continue.
```

Problem No – 2

Write a program to find GCD (greatest common divisor or HCF) and LCM (least common multiple) of two numbers.



The screenshot shows the Code::Blocks IDE with a C++ program open. The program calculates the GCD and LCM of two numbers using the Euclidean algorithm. The output window shows the results for the input numbers 50 and 24.

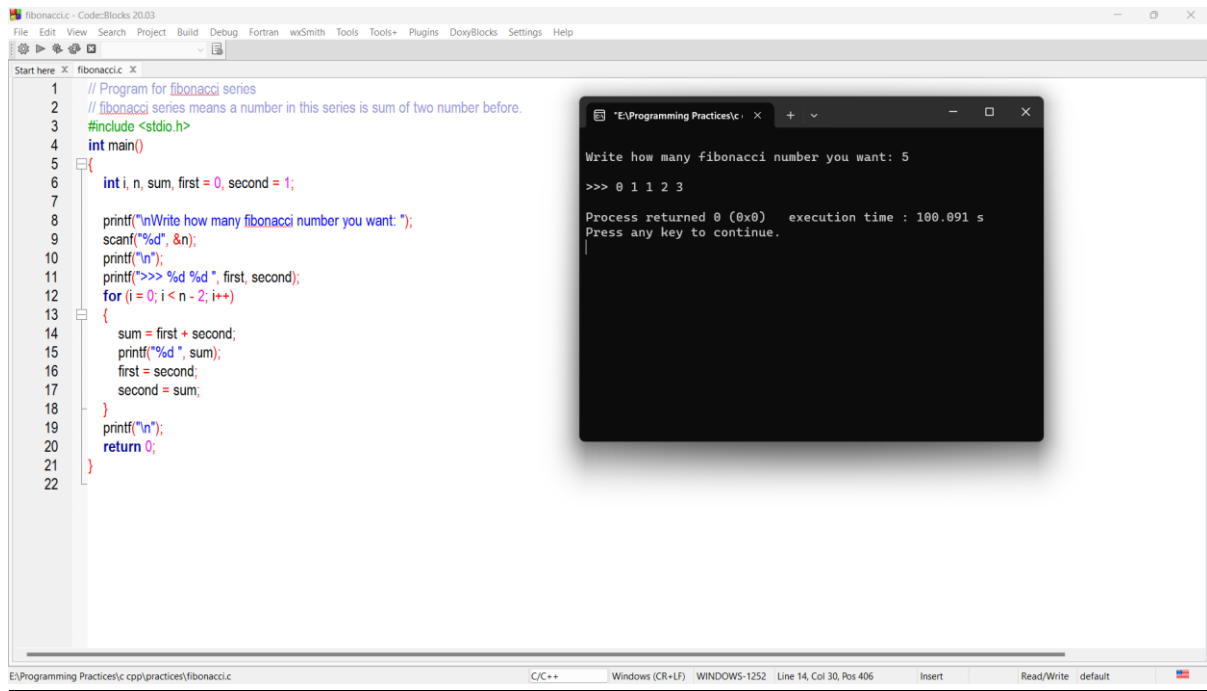
```
1 #include <stdio.h>
2
3 int main()
4 {
5     int n1,n2,remainder;
6     printf("Enter two number (first number must be greater than second) >>> ");
7     scanf("%d %d",&n1,&n2);
8
9     int num1=n1,num2=n2;
10
11
12     while(remainder!=0){
13
14         remainder=n1%n2;
15         n1=n2;
16         n2=remainder;
17     }
18
19     printf("The GCD is %d\n",n1);
20     printf("The LCM is %d", (num1*num2)/n1);
21
22
23
24
25
26     return 0;
27 }
28
```

Output:

```
Enter two number (first number must be greater than second)
>>> 50 24
The GCD is 2
The LCM is 600
Process returned 0 (0x0)   execution time : 8.880 s
Press any key to continue.
```

Problem No – 3

Write a program to print the Fibonacci series up to n terms where n is user input.



The screenshot displays a C++ IDE with a file named 'fibonacci.c'. The code is as follows:

```
1 // Program for fibonacci series
2 // fibonacci series means a number in this series is sum of two number before.
3 #include <stdio.h>
4 int main()
5 {
6     int i, n, sum, first = 0, second = 1;
7
8     printf("\nWrite how many fibonacci number you want: ");
9     scanf("%d", &n);
10    printf("\n");
11    printf(">>> %d %d ", first, second);
12    for (i = 0; i < n - 2; i++)
13    {
14        sum = first + second;
15        printf("%d ", sum);
16        first = second;
17        second = sum;
18    }
19    printf("\n");
20    return 0;
21 }
22
```

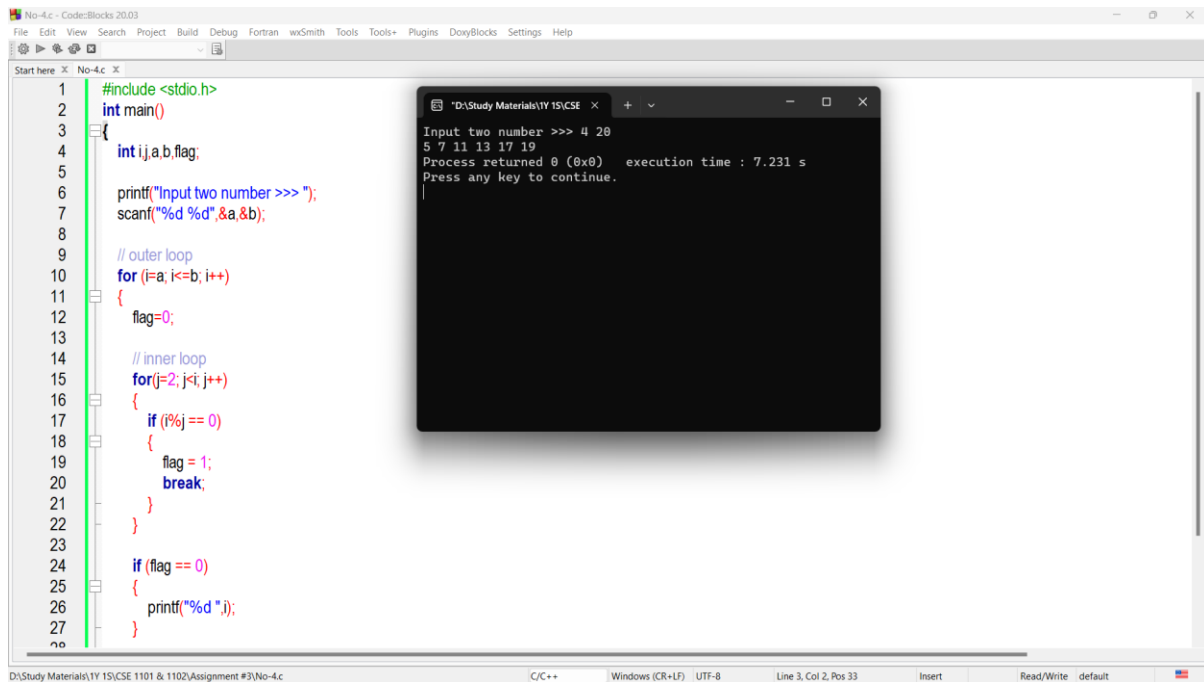
Overlaid on the IDE is a terminal window titled 'E:\Programming Practices\c...' showing the program's execution:

```
Write how many fibonacci number you want: 5
>>> 0 1 1 2 3
Process returned 0 (0x0)   execution time : 100.091 s
Press any key to continue.
```

The IDE's status bar at the bottom indicates the file path 'E:\Programming Practices\c.cpp\practices\fibonacci.c', the language 'C/C++', and the window settings 'Windows (CR+LF)', 'WINDOWS-1252', 'Line 14, Col 30, Pos 406'.

Problem No – 4

Write a program to determine all prime numbers within the range [a ...b] where a & b are user input.



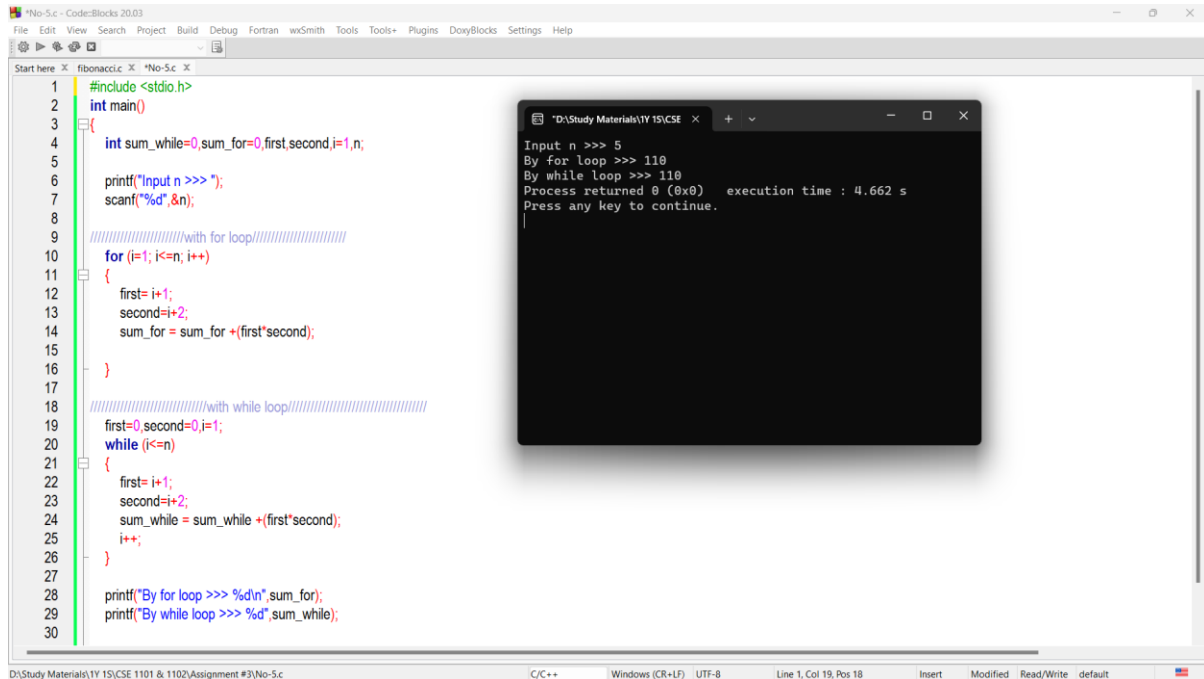
The screenshot shows the Code::Blocks IDE with a C++ program to find prime numbers. The code is as follows:

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,j,a,b,flag;
5
6     printf("Input two number >>> ");
7     scanf("%d %d",&a,&b);
8
9     // outer loop
10    for (i=a; i<=b; i++)
11    {
12        flag=0;
13
14        // inner loop
15        for (j=2; j<i; j++)
16        {
17            if (i%j == 0)
18            {
19                flag = 1;
20                break;
21            }
22        }
23
24        if (flag == 0)
25        {
26            printf("%d ",i);
27        }
28    }
```

The terminal window shows the execution of the program with input 4 and 20, resulting in the output: 5 7 11 13 17 19. The process returned 0 (0x0) and the execution time was 7.231 s.

Problem No – 5

Write a program to find, first using a 'while' loop and then a 'for' loop, the sum of first n terms ($n \geq 1$) of the series $2 \times 3, 3 \times 4, 4 \times 5, \dots, (n+1) \times (n+2)$. You need to verify that you get the same result in both the cases.

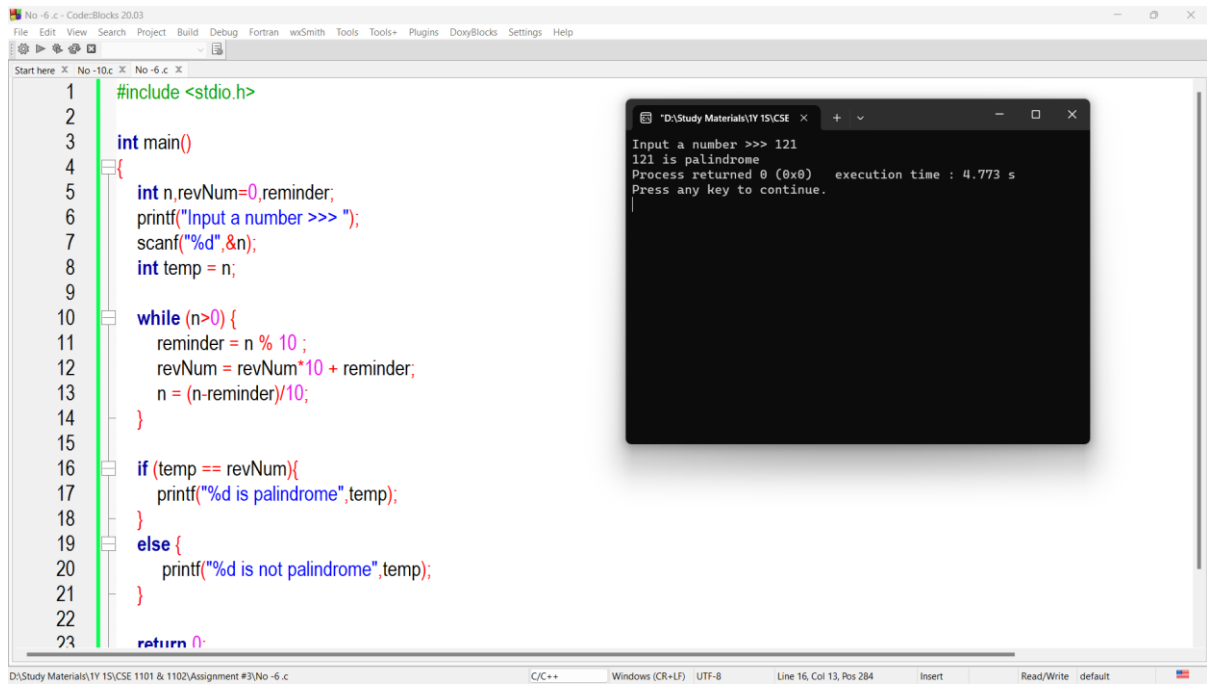


```
1 #include <stdio.h>
2 int main()
3 {
4     int sum_while=0, sum_for=0, first, second, i=1, n;
5
6     printf("Input n >>> ");
7     scanf("%d", &n);
8
9     //with for loop//
10    for (i=1; i<=n; i++)
11    {
12        first= i+1;
13        second=i+2;
14        sum_for = sum_for + (first*second);
15    }
16
17    //with while loop//
18    first=0, second=0, i=1;
19    while (i<=n)
20    {
21        first= i+1;
22        second=i+2;
23        sum_while = sum_while + (first*second);
24        i++;
25    }
26
27    printf("By for loop >>> %d\n", sum_for);
28    printf("By while loop >>> %d", sum_while);
29
30 }
```

Input n >>> 5
By for loop >>> 110
By while loop >>> 110
Process returned 0 (0x0) execution time : 4.662 s
Press any key to continue.

Problem No – 6

Write a program to check whether a given integer is palindrome or not. [121 is palindrome but 123 is not]

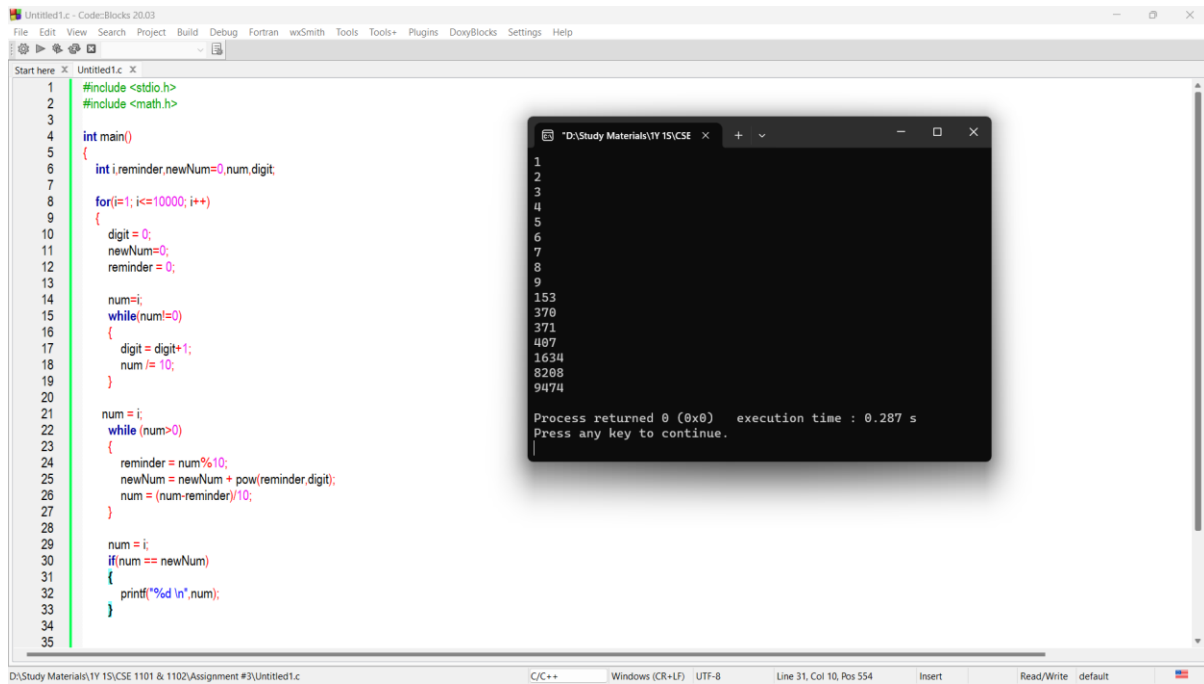


```
1 #include <stdio.h>
2
3 int main()
4 {
5     int n, revNum=0, remainder;
6     printf("Input a number >>> ");
7     scanf("%d", &n);
8     int temp = n;
9
10    while (n>0) {
11        remainder = n % 10;
12        revNum = revNum*10 + remainder;
13        n = (n-remainder)/10;
14    }
15
16    if (temp == revNum){
17        printf("%d is palindrome", temp);
18    }
19    else {
20        printf("%d is not palindrome", temp);
21    }
22
23    return 0;
}
```

```
Input a number >>> 121
121 is palindrome
Process returned 0 (0x0)   execution time : 4.773 s
Press any key to continue.
```

Problem No – 7

Write a program to print out all Armstrong numbers between 1 and 10000.
[Example, $153 = (1*1*1) + (5*5*5) + (3*3*3)$].



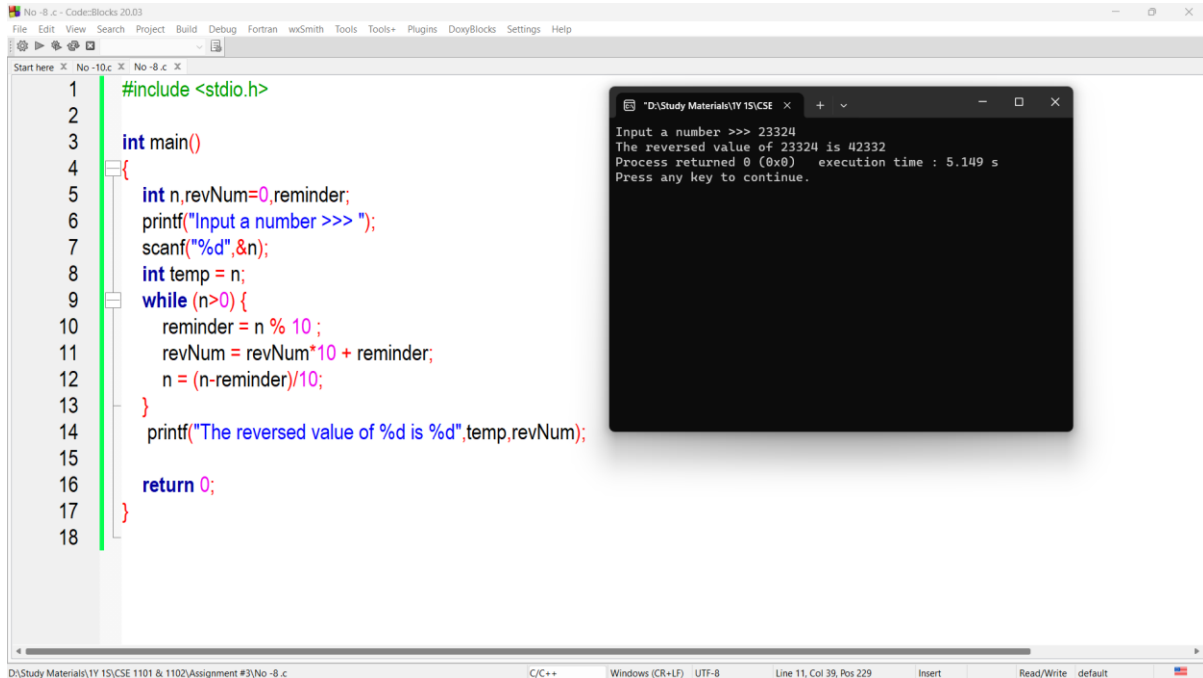
The screenshot shows the Code::Blocks IDE with a C++ program to find Armstrong numbers. The program iterates from 1 to 10000, calculates the sum of the cubes of its digits, and prints the number if the sum equals the original number. The output window shows the following Armstrong numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, and 9474. The program execution time is 0.287 s.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     int i, remainder, newNum=0, num, digit;
7
8     for(i=1; i<=10000; i++)
9     {
10         digit = 0;
11         newNum=0;
12         remainder = 0;
13
14         num=i;
15         while(num!=0)
16         {
17             digit = digit+1;
18             num /= 10;
19         }
20
21         num = i;
22         while (num>0)
23         {
24             remainder = num%10;
25             newNum = newNum + pow(remainder,digit);
26             num = (num-remainder)/10;
27         }
28
29         num = i;
30         if(num == newNum)
31         {
32             printf("%d\n",num);
33         }
34     }
35 }
```

Process returned 0 (0x0) execution time : 0.287 s
Press any key to continue.

Problem No – 8

Write a program to reverse a given number using while loop.



The screenshot shows the Code::Blocks IDE with a C++ project named 'No -8.c'. The code in the editor is as follows:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int n, revNum=0, remainder;
6     printf("Input a number >>> ");
7     scanf("%d", &n);
8     int temp = n;
9     while (n>0) {
10         remainder = n % 10;
11         revNum = revNum*10 + remainder;
12         n = (n-remainder)/10;
13     }
14     printf("The reversed value of %d is %d", temp, revNum);
15
16     return 0;
17 }
18
```

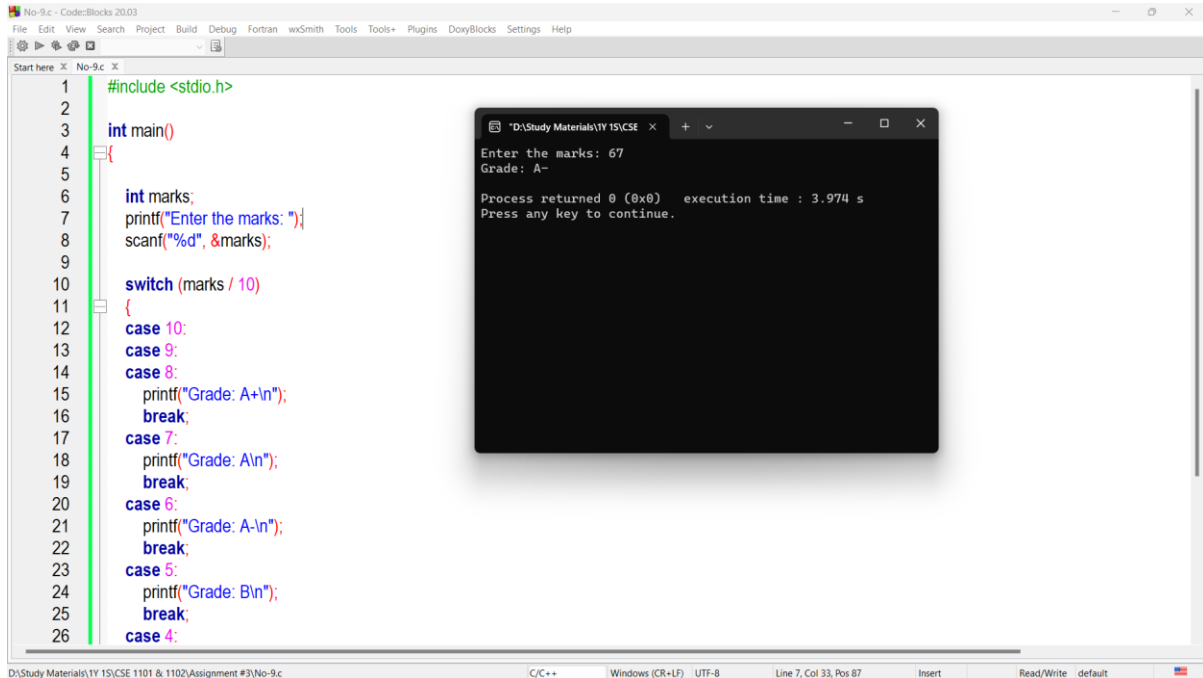
Overlaid on the IDE is a terminal window titled 'D:\Study Materials\TY 15\CSE'. It displays the execution output:

```
Input a number >>> 23324
The reversed value of 23324 is 42332
Process returned 0 (0x0)   execution time : 5.149 s
Press any key to continue.
```

The status bar at the bottom of the IDE indicates the file path 'D:\Study Materials\TY 15\CSE 1101 & 1102\Assignment #3\No -8.c', the compiler 'C/C++', the window title 'Windows (CR+LF)', the encoding 'UTF-8', and the current cursor position 'Line 11, Col 39, Pos 229'.

Problem No – 9

Write a program to find a grade of given marks using switch case statement.



The screenshot displays the CodeBlocks 20.03 IDE with a C++ program and its execution output. The program uses a switch case statement to determine the grade based on marks. The execution window shows the input '67' and the resulting grade 'A-'. The program's execution time is 3.974 s.

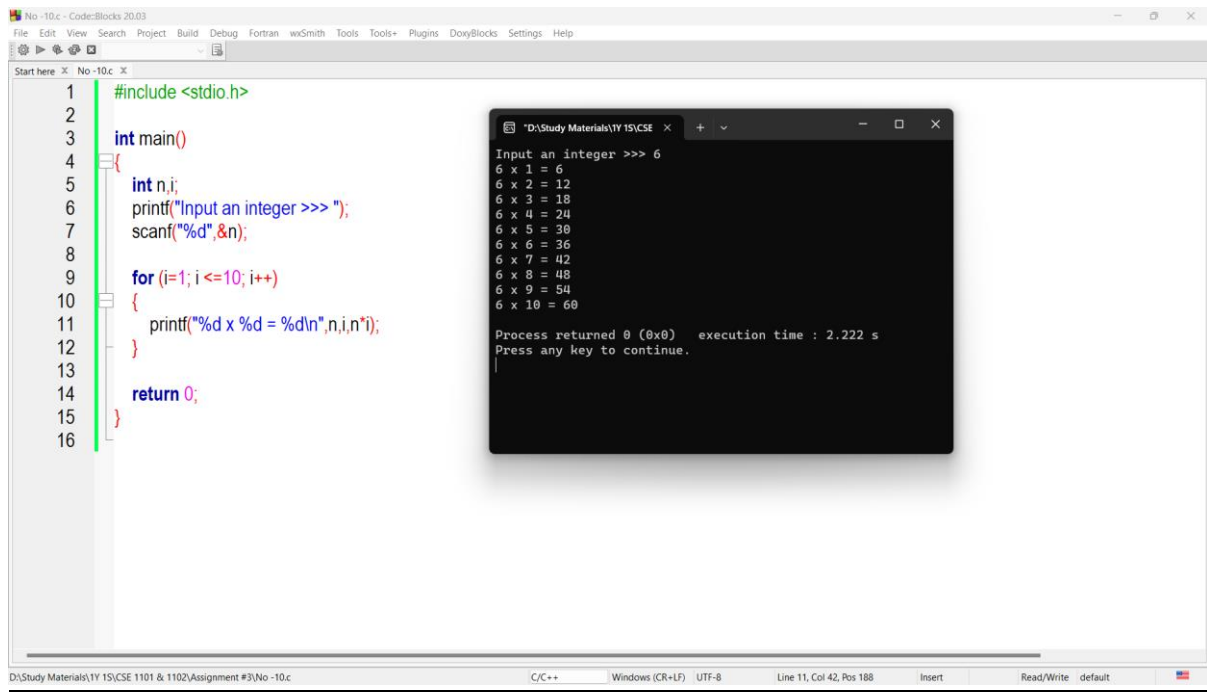
```
1 #include <stdio.h>
2
3 int main()
4 {
5
6     int marks;
7     printf("Enter the marks: ");
8     scanf("%d", &marks);
9
10    switch (marks / 10)
11    {
12        case 10:
13        case 9:
14        case 8:
15            printf("Grade: A+\n");
16            break;
17        case 7:
18            printf("Grade: A\n");
19            break;
20        case 6:
21            printf("Grade: A-\n");
22            break;
23        case 5:
24            printf("Grade: B\n");
25            break;
26        case 4:
```

Execution Output:

```
Enter the marks: 67
Grade: A-
Process returned 0 (0x0)   execution time : 3.974 s
Press any key to continue.
```

Problem No – 10

Write a program in C to display the multiplication table for a given integer.



The screenshot shows the Code::Blocks IDE with a C program open. The program prompts the user to input an integer, which is 6. It then displays the multiplication table for 6, showing products from 6 x 1 to 6 x 10. The program also shows the execution time and a prompt to press any key to continue.

```
#include <stdio.h>

int main()
{
    int n,i;
    printf("Input an integer >>> ");
    scanf("%d",&n);

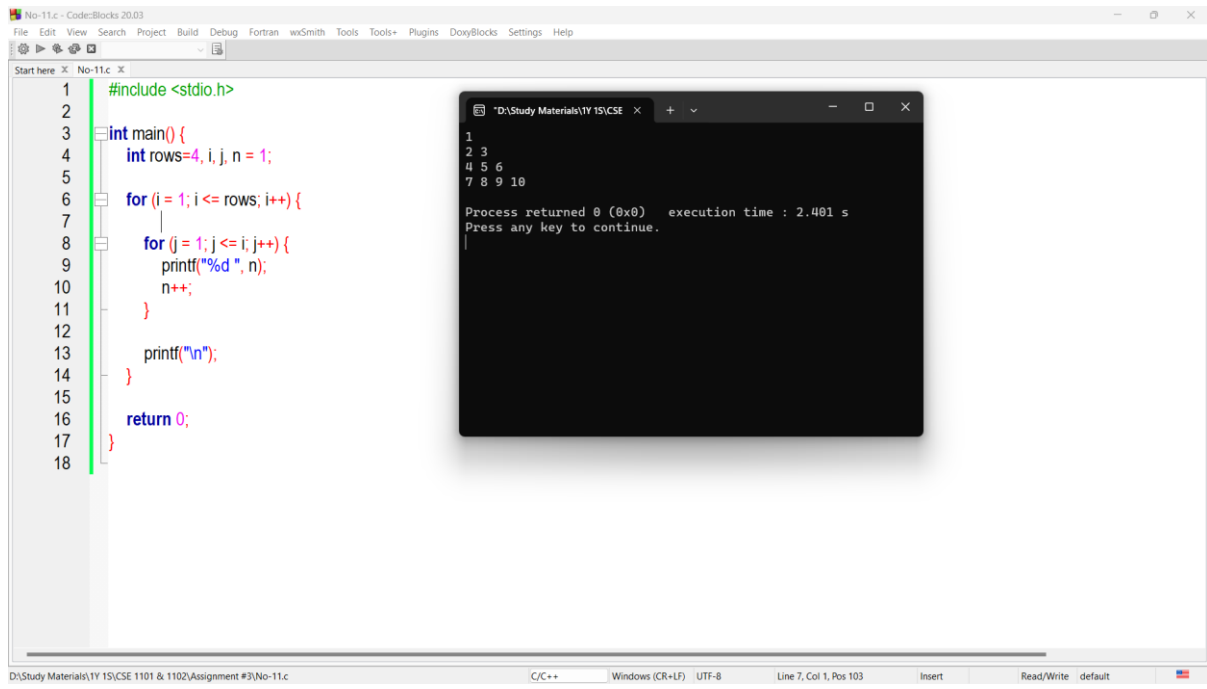
    for (i=1; i <=10; i++)
    {
        printf("%d x %d = %d\n",n,i,n*i);
    }

    return 0;
}
```

Input an integer >>> 6
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
Process returned 0 (0x0) execution time : 2.222 s
Press any key to continue.

Problem No – 11

Write a program in C to make such a pattern like a right angle triangle with the number increased by 1.



The screenshot shows the Code::Blocks IDE with a C program that generates a right angle triangle pattern. The program is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int rows=4, i, j, n = 1;
5
6     for (i = 1; i <= rows; i++) {
7
8         for (j = 1; j <= i; j++) {
9             printf("%d ", n);
10            n++;
11        }
12
13        printf("\n");
14    }
15
16    return 0;
17 }
18
```

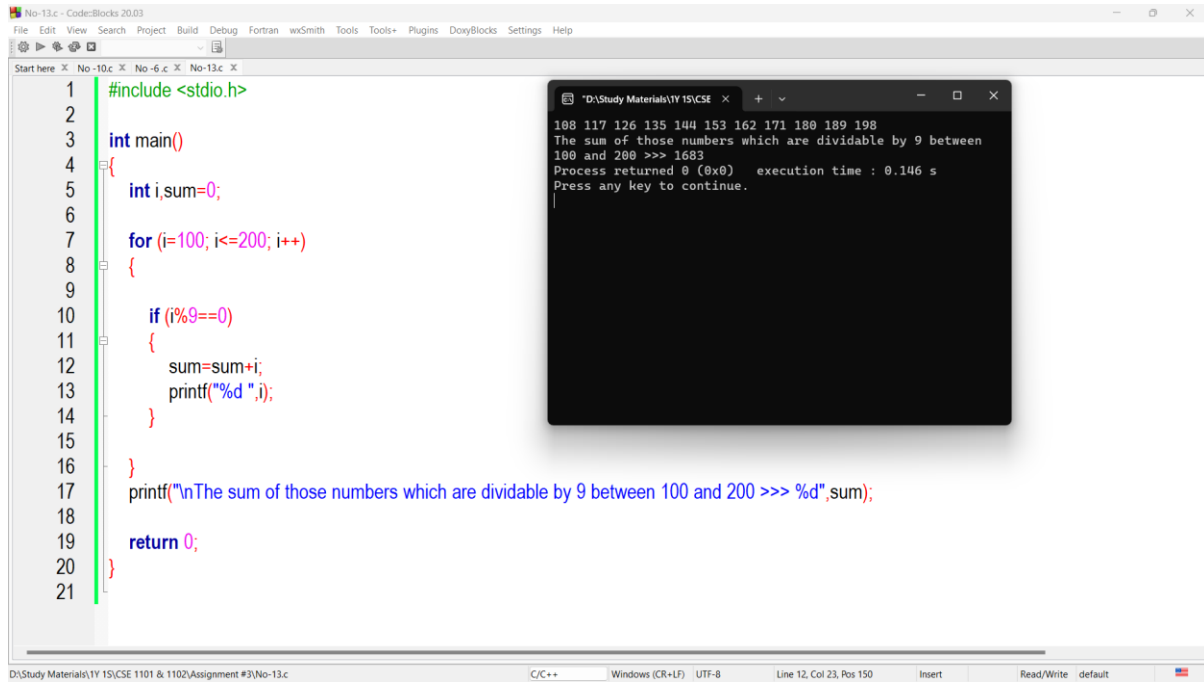
The output of the program, displayed in a terminal window, is a right angle triangle pattern where the numbers increase by 1 in each row:

```
1
2 3
4 5 6
7 8 9 10
```

Below the pattern, the terminal shows the execution status: "Process returned 0 (0x0) execution time : 2.401 s" and "Press any key to continue.".

Problem No – 13

Write a program in C to find the number and sum of all integers between 100 and 200, which are divisible by 9.



The screenshot shows the Code::Blocks IDE with a C program and its execution output. The program is designed to find and sum integers between 100 and 200 that are divisible by 9.

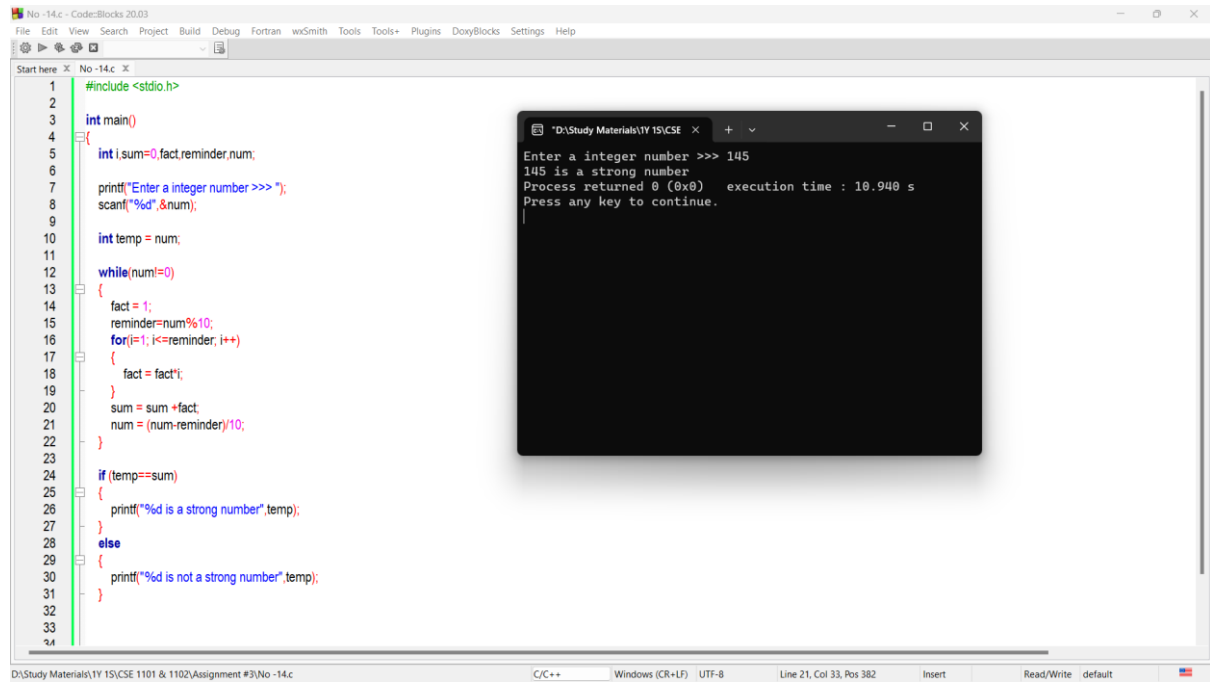
```
1  #include <stdio.h>
2
3  int main()
4  {
5      int i,sum=0;
6
7      for (i=100; i<=200; i++)
8      {
9
10         if (i%9==0)
11         {
12             sum=sum+i;
13             printf("%d ",i);
14         }
15     }
16
17     printf("\nThe sum of those numbers which are dividable by 9 between 100 and 200 >>> %d",sum);
18
19     return 0;
20 }
21
```

The execution output in the terminal window shows the numbers 108 through 198, which are divisible by 9, followed by the sum 1683. It also displays the process return value and execution time.

```
"D:\Study Materials\TY 15\CSE" x + -
108 117 126 135 144 153 162 171 180 189 198
The sum of those numbers which are dividable by 9 between
100 and 200 >>> 1683
Process returned 0 (0x0)   execution time : 0.146 s
Press any key to continue.
```

Problem No – 14

Write a C program to check whether a number is a Strong Number or not (A strong number is one in which the factorial of each digit of a number equals the original number's sum).



The screenshot shows the Code::Blocks IDE with a C program to check if a number is a strong number. The program is named 'No -14.c' and is located at 'D:\Study Materials\1Y 1S\CSE 1101 & 1102\Assignment #3\No -14.c'. The code is as follows:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int i, sum=0, fact, remainder, num;
6
7     printf("Enter a integer number >>> ");
8     scanf("%d", &num);
9
10    int temp = num;
11
12    while(num!=0)
13    {
14        fact = 1;
15        remainder = num%10;
16        for(i=1; i<=remainder; i++)
17        {
18            fact = fact*i;
19        }
20        sum = sum + fact;
21        num = (num-remainder)/10;
22    }
23
24    if (temp==sum)
25    {
26        printf("%d is a strong number", temp);
27    }
28    else
29    {
30        printf("%d is not a strong number", temp);
31    }
32
33 }
```

The program is executed, and the output is shown in a terminal window:

```
Enter a integer number >>> 145
145 is a strong number
Process returned 0 (0x0)   execution time : 10.940 s
Press any key to continue.
```

The status bar at the bottom of the IDE shows 'C/C++', 'Windows (CR+LF)', 'UTF-8', 'Line 21, Col 33, Pos 382', 'Insert', 'Read/Write', 'default', and a small icon.