

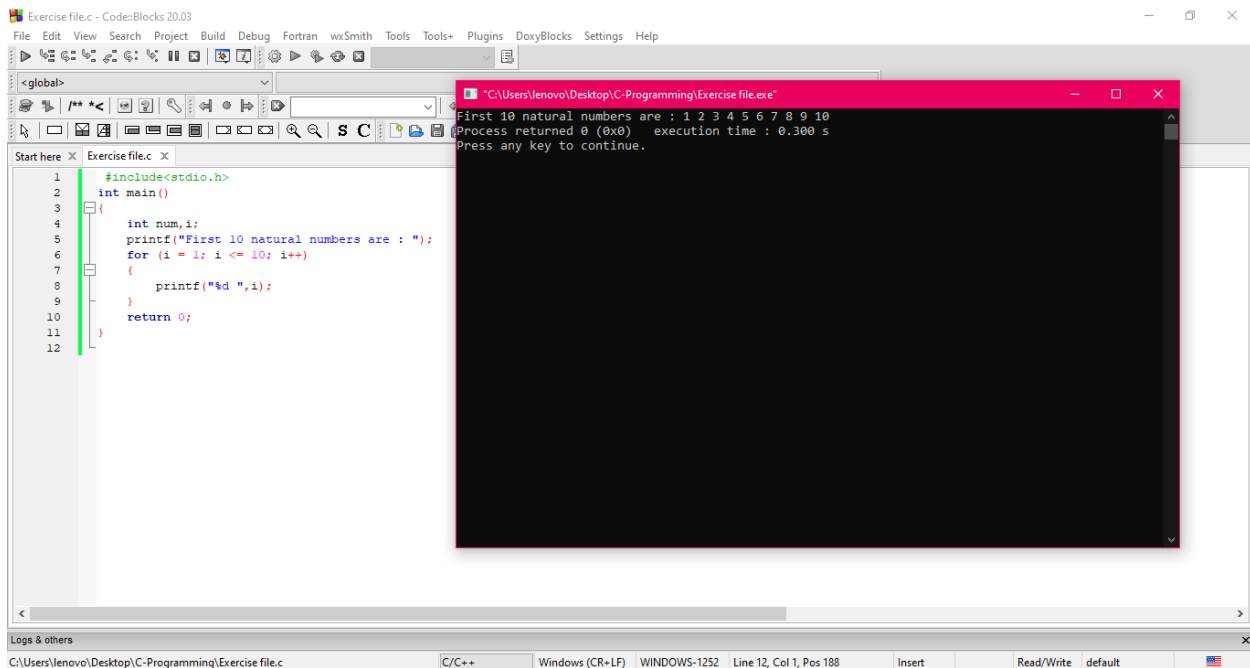
ASSIGNMENT 08 & 09

NAME: Om Vivek Gharge

PRN: 2020BTEIT00041

BRANCH: INFORMATION TECHNOLOGY

Q1. Write a program in C to display the first 10 natural numbers.



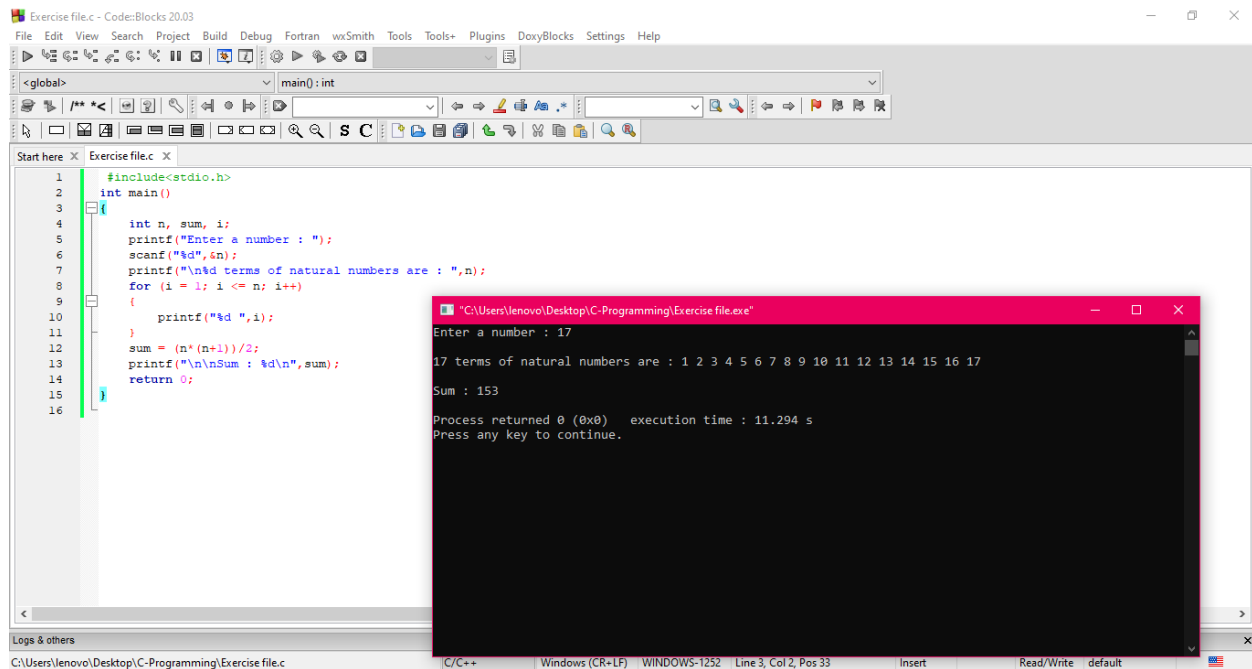
The screenshot displays the Code::Blocks 20.03 IDE. The main editor window shows a C program named 'Exercise file.c' with the following code:

```
1 #include<stdio.h>
2 int main()
3 {
4     int num,i;
5     printf("First 10 natural numbers are : ");
6     for (i = 1; i <= 10; i++)
7     {
8         printf("%d ",i);
9     }
10    return 0;
11 }
12
```

Overlaid on the IDE is a terminal window titled "C:\Users\lenovo\Desktop\C-Programming\Exercise file.exe". It shows the output of the program: "First 10 natural numbers are : 1 2 3 4 5 6 7 8 9 10". Below the output, it states "Process returned 0 (0x0) execution time : 0.300 s" and "Press any key to continue.".

The status bar at the bottom of the IDE indicates the file path "C:\Users\lenovo\Desktop\C-Programming\Exercise file.c", the language "C/C++", the window title "Windows (CR+LF)", the window ID "WINDOWS-1252", the cursor position "Line 12, Col 1, Pos 188", and the editor mode "Insert".

Q2. Write a program in C to display n terms of natural number and their sum.



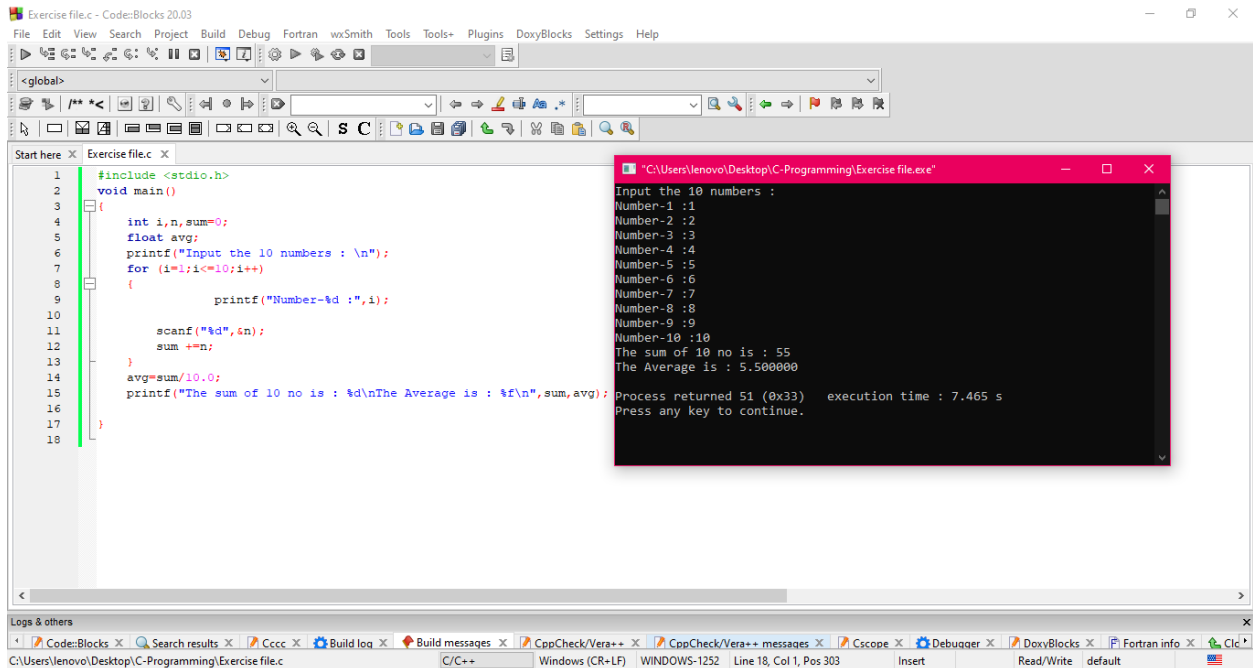
The screenshot shows the Code::Blocks IDE with a C program open. The program prompts the user to enter a number, displays the first n terms of natural numbers, and calculates their sum. The execution window shows the program running with the input 17, displaying the sequence 1 to 17 and a sum of 153.

```
1 #include<stdio.h>
2 int main()
3 {
4     int n, sum, i;
5     printf("Enter a number : ");
6     scanf("%d",&n);
7     printf("\n%d terms of natural numbers are : ",n);
8     for (i = 1; i <= n; i++)
9     {
10         printf("%d ",i);
11     }
12     sum = (n*(n+1))/2;
13     printf("\n\nSum : %d\n",sum);
14     return 0;
15 }
16
```

Execution Output:

```
Enter a number : 17
17 terms of natural numbers are : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Sum : 153
Process returned 0 (0x0)   execution time : 11.294 s
Press any key to continue.
```

Q3. Write a program in C to read 10 numbers from keyboard and find their sum and average.



The screenshot displays the Code::Blocks IDE with a C program open in the editor and its execution output in a separate window.

Source Code (Exercise file.c):

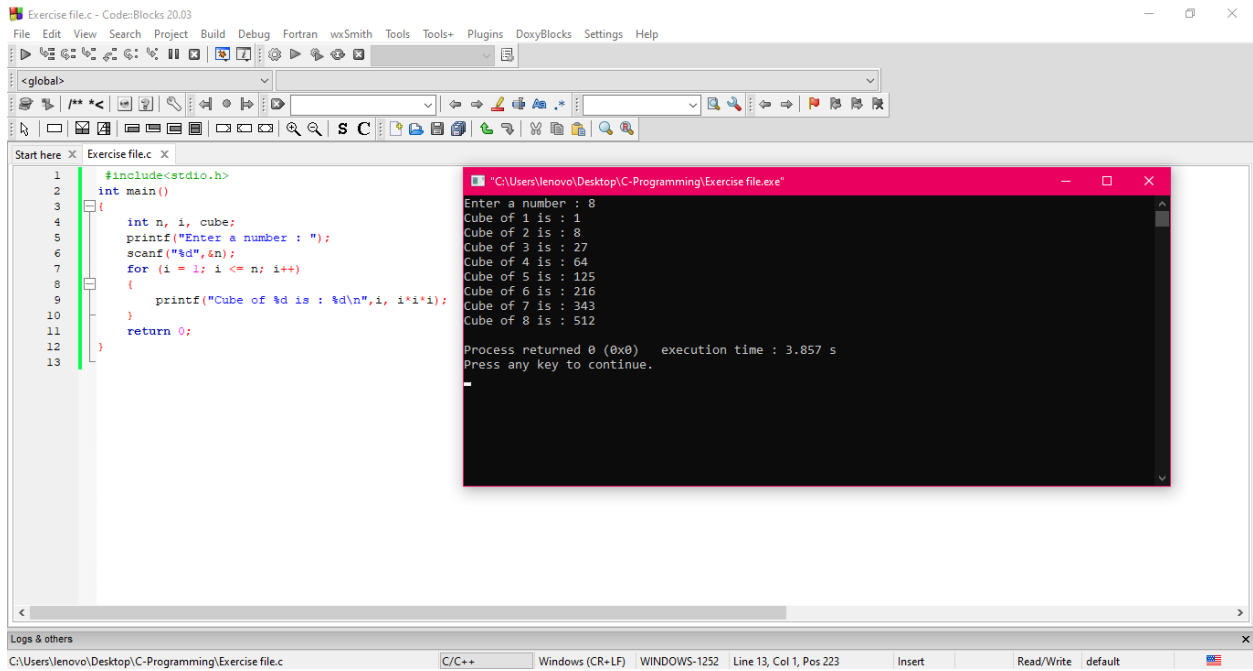
```
1  #include <stdio.h>
2  void main()
3  {
4      int i,n,sum=0;
5      float avg;
6      printf("Input the 10 numbers : \n");
7      for (i=1;i<=10;i++)
8      {
9          printf("Number-%d :",i);
10
11          scanf("%d",&n);
12          sum +=n;
13      }
14      avg=sum/10.0;
15      printf("The sum of 10 no is : %d\nThe Average is : %f\n",sum,avg);
16
17  }
```

Execution Output (C:\Users\lenovo\Desktop\C-Programming\Exercise file.exe):

```
Input the 10 numbers :
Number-1 :1
Number-2 :2
Number-3 :3
Number-4 :4
Number-5 :5
Number-6 :6
Number-7 :7
Number-8 :8
Number-9 :9
Number-10 :10
The sum of 10 no is : 55
The Average is : 5.500000
Process returned 51 (0x33)   execution time : 7.465 s
Press any key to continue.
```

The IDE interface includes a menu bar (File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help), a toolbar, a project explorer on the left, and a status bar at the bottom showing the current file path, language (C/C++), and window state (Windows (CR-LF), WINDOWS-1252, Line 18, Col 1, Pos 303).

Q4. Write a program in C to display the cube of the number up to given an integer.



The screenshot shows a C code editor window titled "Exercise file.c - Code::Blocks 20.03". The code is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int n, i, cube;
5     printf("Enter a number : ");
6     scanf("%d",&n);
7     for (i = 1; i <= n; i++)
8     {
9         printf("Cube of %d is : %d\n",i, i*i*i);
10    }
11    return 0;
12 }
13
```

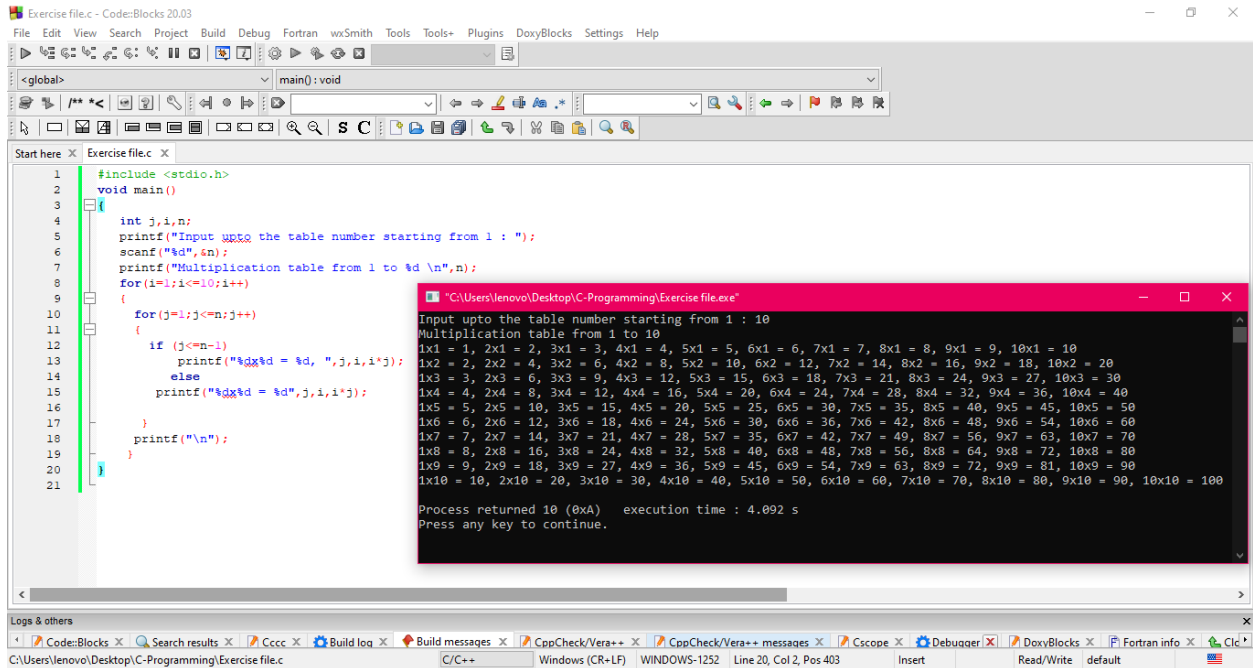
The output window, titled "C:\Users\lenovo\Desktop\C-Programming\Exercise file.exe", shows the following output:

```
Enter a number : 8
Cube of 1 is : 1
Cube of 2 is : 8
Cube of 3 is : 27
Cube of 4 is : 64
Cube of 5 is : 125
Cube of 6 is : 216
Cube of 7 is : 343
Cube of 8 is : 512

Process returned 0 (0x0)   execution time : 3.857 s
Press any key to continue.
```

The status bar at the bottom indicates the file path "C:\Users\lenovo\Desktop\C-Programming\Exercise file.c", the editor is "C/C++", the window is "Windows (CR+LF)", the file encoding is "WINDOWS-1252", the cursor is at "Line 13, Col 1, Pos 223", and the keyboard layout is "Insert".

Q5. Write a program in C to display multiplication table from 1 to n.



The screenshot shows the Code::Blocks IDE with a C program to generate a multiplication table. The source code is as follows:

```
1 #include <stdio.h>
2 void main()
3 {
4     int j,i,n;
5     printf("Input upto the table number starting from 1 : ");
6     scanf("%d",&n);
7     printf("Multiplication table from 1 to %d \n",n);
8     for(i=1;i<=10;i++)
9     {
10        for(j=1;j<=n;j++)
11        {
12            if (j<=n-1)
13                printf("%dx%d = %d, ",j,i,i*j);
14            else
15                printf("%dx%d = %d",j,i,i*j);
16        }
17        printf("\n");
18    }
19 }
20
21
```

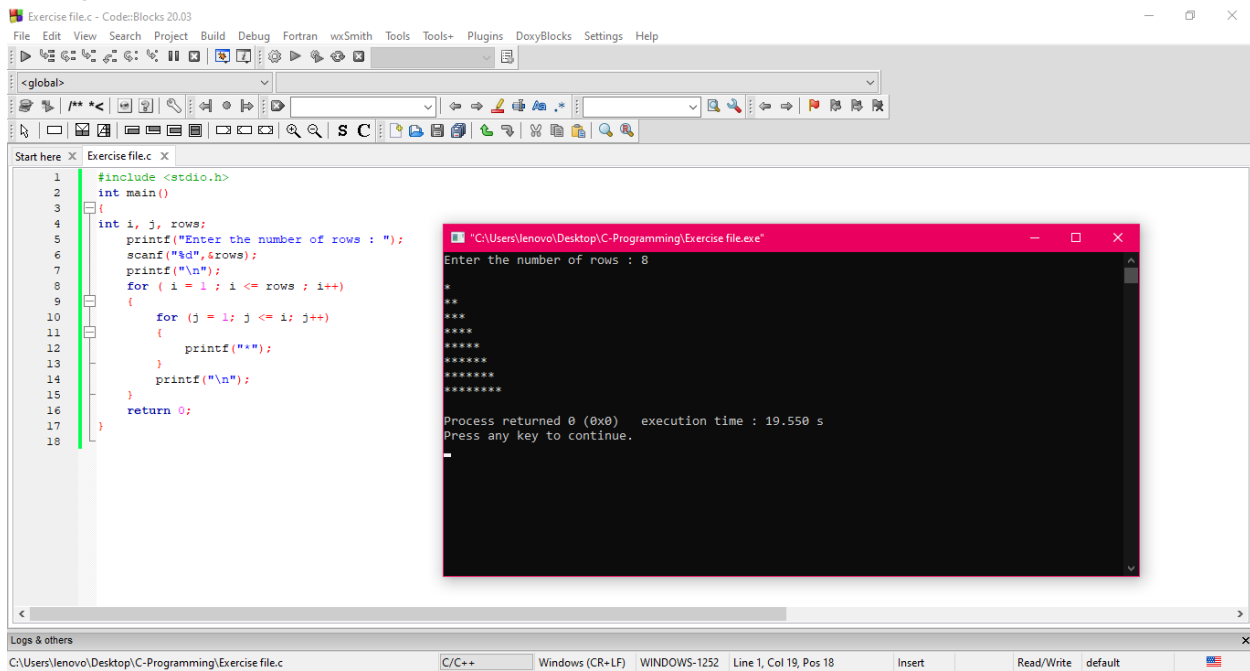
The execution output window shows the program running with input 10, displaying a 10x10 multiplication table. The output is as follows:

```
Input upto the table number starting from 1 : 10
Multiplication table from 1 to 10
1x1 = 1, 2x1 = 2, 3x1 = 3, 4x1 = 4, 5x1 = 5, 6x1 = 6, 7x1 = 7, 8x1 = 8, 9x1 = 9, 10x1 = 10
1x2 = 2, 2x2 = 4, 3x2 = 6, 4x2 = 8, 5x2 = 10, 6x2 = 12, 7x2 = 14, 8x2 = 16, 9x2 = 18, 10x2 = 20
1x3 = 3, 2x3 = 6, 3x3 = 9, 4x3 = 12, 5x3 = 15, 6x3 = 18, 7x3 = 21, 8x3 = 24, 9x3 = 27, 10x3 = 30
1x4 = 4, 2x4 = 8, 3x4 = 12, 4x4 = 16, 5x4 = 20, 6x4 = 24, 7x4 = 28, 8x4 = 32, 9x4 = 36, 10x4 = 40
1x5 = 5, 2x5 = 10, 3x5 = 15, 4x5 = 20, 5x5 = 25, 6x5 = 30, 7x5 = 35, 8x5 = 40, 9x5 = 45, 10x5 = 50
1x6 = 6, 2x6 = 12, 3x6 = 18, 4x6 = 24, 5x6 = 30, 6x6 = 36, 7x6 = 42, 8x6 = 48, 9x6 = 54, 10x6 = 60
1x7 = 7, 2x7 = 14, 3x7 = 21, 4x7 = 28, 5x7 = 35, 6x7 = 42, 7x7 = 49, 8x7 = 56, 9x7 = 63, 10x7 = 70
1x8 = 8, 2x8 = 16, 3x8 = 24, 4x8 = 32, 5x8 = 40, 6x8 = 48, 7x8 = 56, 8x8 = 64, 9x8 = 72, 10x8 = 80
1x9 = 9, 2x9 = 18, 3x9 = 27, 4x9 = 36, 5x9 = 45, 6x9 = 54, 7x9 = 63, 8x9 = 72, 9x9 = 81, 10x9 = 90
1x10 = 10, 2x10 = 20, 3x10 = 30, 4x10 = 40, 5x10 = 50, 6x10 = 60, 7x10 = 70, 8x10 = 80, 9x10 = 90, 10x10 = 100

Process returned 10 (0xA)   execution time : 4.092 s
Press any key to continue.
```

The bottom status bar indicates the file path is C:\Users\lenovo\Desktop\C-Programming\Exercise file.c, the language is C/C++, and the window title is Windows (CR+LF).

Q6. Write a program in C to make such a pattern like right angle triangle using an asterisk.



The screenshot displays the Code::Blocks IDE with a C program designed to print a right-angled triangle pattern of asterisks. The source code is as follows:

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

The program's execution is shown in a separate window titled "C:\Users\lenovo\Desktop\C-Programming\Exercise file.exe". It prompts the user to "Enter the number of rows : 8". The output displays a right-angled triangle pattern consisting of 8 rows of asterisks. The first row has 1 asterisk, the second has 2, and so on, up to 8 asterisks in the eighth row. Below the pattern, the program reports "Process returned 0 (0x0) execution time : 19.550 s" and "Press any key to continue."

At the bottom of the IDE, the status bar indicates the file path "C:\Users\lenovo\Desktop\C-Programming\Exercise file.c", the language "C/C++", the window title "Windows (CR+LF)", the window ID "WINDOWS-1252", the current cursor position "Line 1, Col 19, Pos 18", and the keyboard layout "Insert".

Q7. Write a program in C to make such a pattern like right angle triangle with number increased by 1.

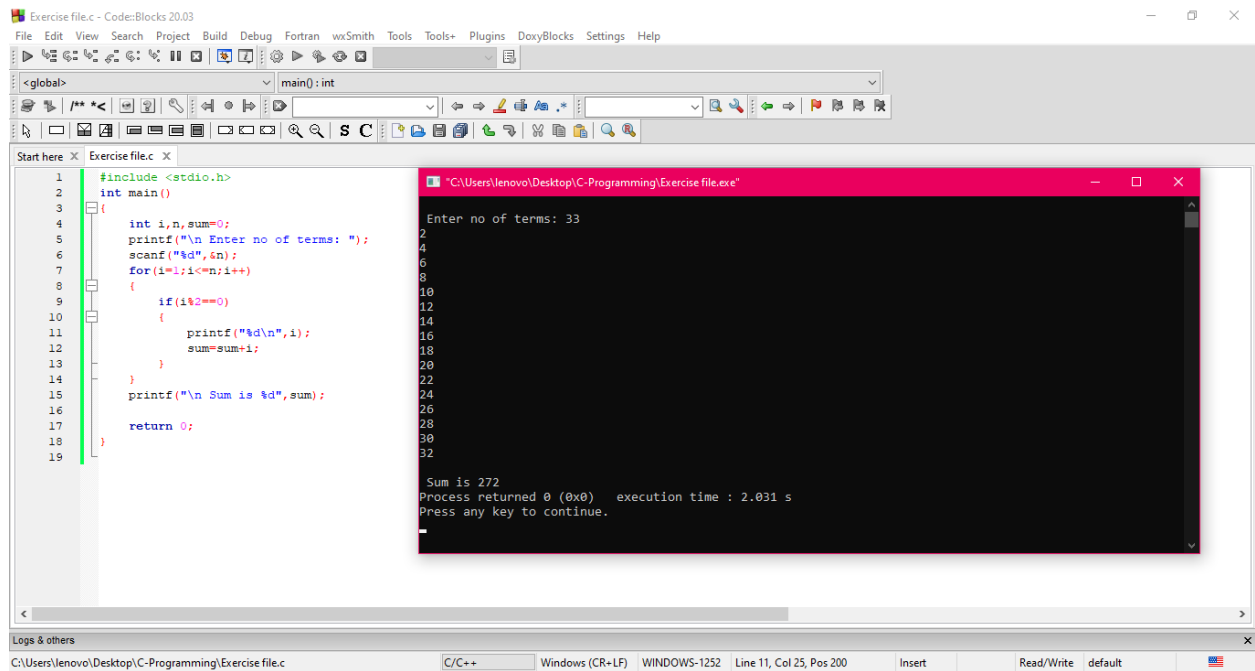
```
#include<stdio.h>
int main()
{
    int i, j, rows, a = 1;
    printf("Enter the number of rows : ");
    scanf("%d",&rows);
    printf("\n");
    for ( i = 1; i <= rows; i++)
    {
        for ( j = 1; j <= i; j++)
        {
            printf("%d ",a);
            a++;
        }
        printf("\n");
    }
    return 0;
}
```

Enter the number of rows : 17

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45
46 47 48 49 50 51 52 53 54 55
56 57 58 59 60 61 62 63 64 65 66
67 68 69 70 71 72 73 74 75 76 77 78
79 80 81 82 83 84 85 86 87 88 89 90 91
92 93 94 95 96 97 98 99 100 101 102 103 104 105
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136
137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
```

Process returned 0 (0x0) execution time : 2.024 s
Press any key to continue.

Q8. Write a program in C to display n even natural numbers and their sum .



The screenshot shows the CodeBlocks IDE with a C program open in the editor and its execution output in a separate window.

Source Code (Exercise file.c):

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,n,sum=0;
5     printf("\n Enter no of terms: ");
6     scanf("%d",&n);
7     for(i=1;i<=n;i++)
8     {
9         if(i%2==0)
10        {
11            printf("%d\n",i);
12            sum=sum+i;
13        }
14    }
15    printf("\n Sum is %d",sum);
16
17    return 0;
18 }
19
```

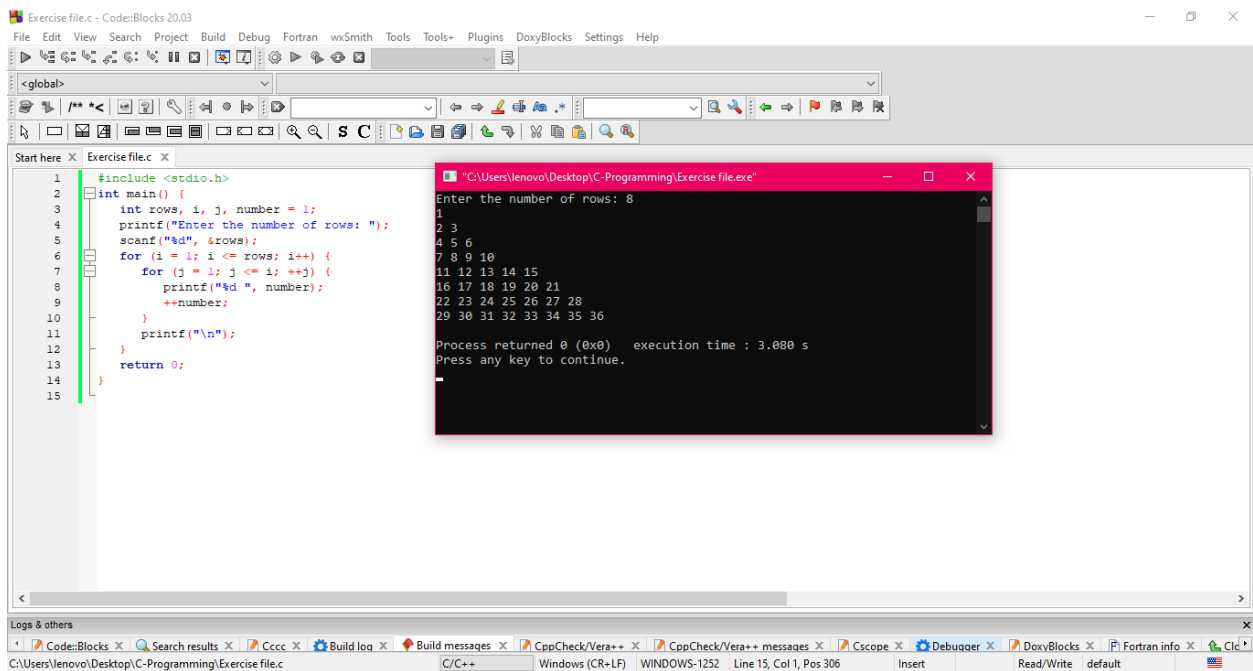
Execution Output (C:\Users\lenovo\Desktop\C-Programming\Exercise file.exe):

```
Enter no of terms: 33
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32

Sum is 272
Process returned 0 (0x0)   execution time : 2.031 s
Press any key to continue.
```

The status bar at the bottom indicates the file path: C:\Users\lenovo\Desktop\C-Programming\Exercise file.c, the language: C/C++, and the current position: Line 11, Col 25, Pos 200.

Q9. Write a program in C to print the Floyd's Triangle.



The screenshot shows the Code::Blocks IDE with a C program that prints Floyd's Triangle. The program prompts the user to enter the number of rows (8) and then prints the triangle. The output is displayed in a separate window titled "C:\Users\lenovo\Desktop\C-Programming\Exercise file.exe".

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

Enter the number of rows: 8

```
1
2 3
3 4 5 6
4 7 8 9 10
5 11 12 13 14 15
6 16 17 18 19 20 21
7 22 23 24 25 26 27 28
8 29 30 31 32 33 34 35 36
```

Process returned 0 (0x0) execution time : 3.080 s
Press any key to continue.

Q10. Write a program in C to find the sum of the series $1 + 11 + 111 + 1111 + \dots n$ terms.

The screenshot shows the Code::Blocks IDE with a C program open in the editor and its execution output in a separate window.

Code::Blocks IDE - Exercise file.c

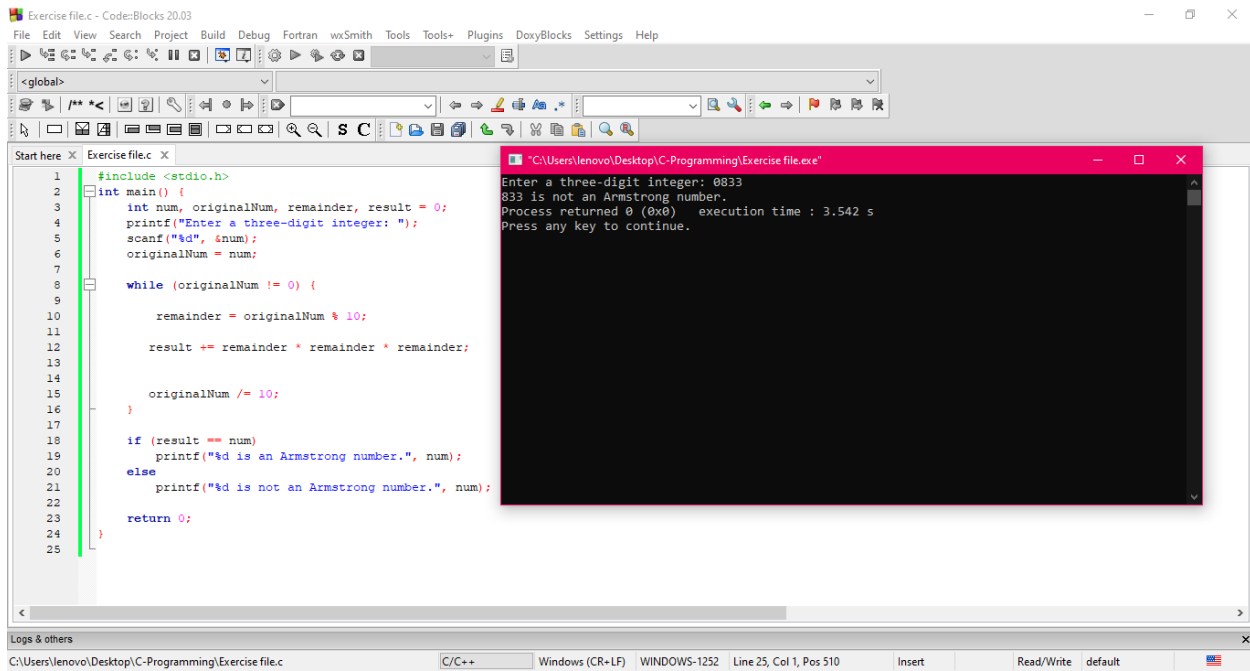
```
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     int sum=0,i;
6     printf("Enter the range of number:");
7     scanf("%d",&n);
8     int p=1;
9     for(i=1;i<=n;i++)
10     {
11
12         sum+=p;
13         p=(p*10)+1;
14     }
15     printf("The sum of the series = %d",sum);
16 }
17
18
```

Execution Output (C:\Users\lenovo\Desktop\C-Programming\Exercise file.exe)

```
Enter the range of number:8
The sum of the series = 12345678
Process returned 0 (0x0)   execution time : 1.552 s
Press any key to continue.
```

IDE Status Bar: C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 10, Col 6, Pos 170 | Insert | Read/Write | default

Q11. Write a C program to check whether a given number is an Armstrong number or not.



The screenshot shows the Code::Blocks IDE with a C program open in the editor and its execution output in a separate window.

Code::Blocks IDE - Exercise file.c

```
1 #include <stdio.h>
2 int main() {
3     int num, originalNum, remainder, result = 0;
4     printf("Enter a three-digit integer: ");
5     scanf("%d", &num);
6     originalNum = num;
7
8     while (originalNum != 0) {
9
10        remainder = originalNum % 10;
11
12        result += remainder * remainder * remainder;
13
14
15        originalNum /= 10;
16    }
17
18    if (result == num)
19        printf("%d is an Armstrong number.", num);
20    else
21        printf("%d is not an Armstrong number.", num);
22
23    return 0;
24 }
25
```

Execution Output (C:\Users\lenovo\Desktop\C-Programming\Exercise file.exe)

```
Enter a three-digit integer: 0833
833 is not an Armstrong number.
Process returned 0 (0x0)   execution time : 3.542 s
Press any key to continue.
```

Log status bar: C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 25, Col 1, Pos 510 | Insert | Read/Write | default

Q12. Write a program in C to find the prime numbers within a range of numbers

The screenshot shows the Code::Blocks IDE with a C program open in the editor and its execution output in a separate window.

Code::Blocks IDE - Exercise file.c

```
1 #include <stdio.h>
2 void main() {
3     int num,i,ctr,stno,enno;
4     printf("Input starting number of range: ");
5     scanf("%d",&stno);
6
7     printf("Input ending number of range: ");
8     scanf("%d",&enno);
9     printf("The prime numbers between %d and %d are : \n",stno,enno);
10
11     for(num = stno;num<=enno;num++)
12     {
13         ctr = 0;
14
15         for(i=2;i<=num/2;i++)
16         {
17             if(num%i==0) {
18                 ctr++;
19                 break;
20             }
21         }
22
23         if(ctr==0 && num!= 1)
24             printf("%d ",num);
25     }
26     printf("\n");
27 }
```

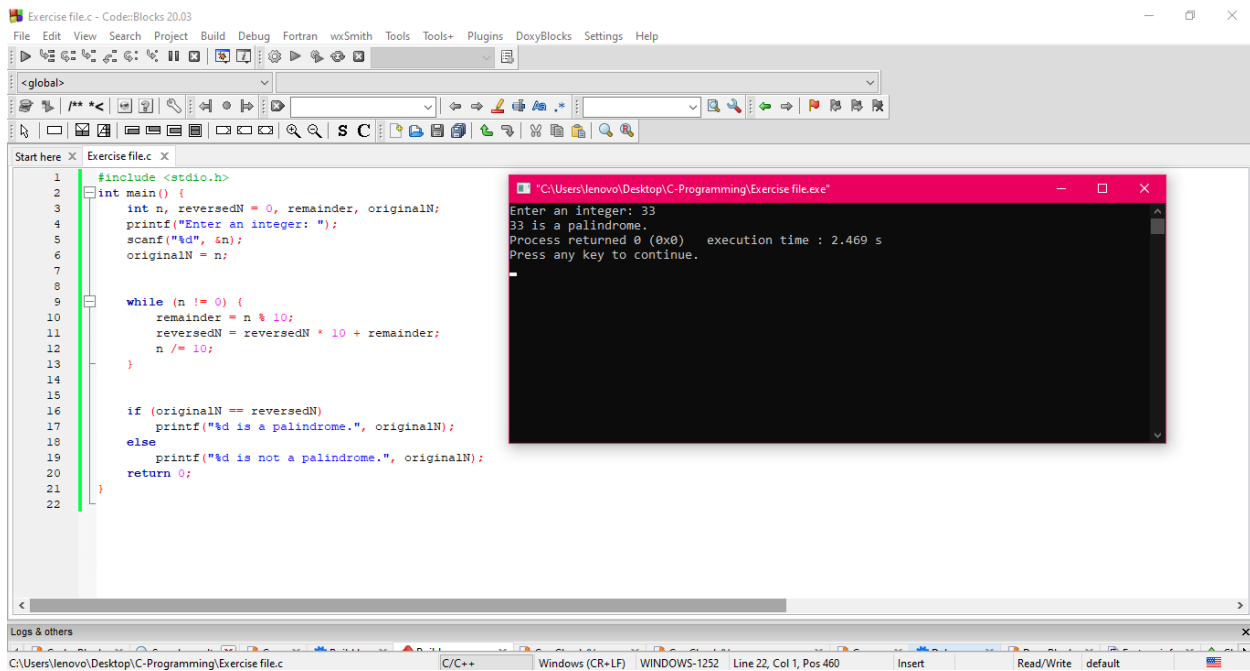
Execution Output (C:\Users\lenovo\Desktop\C-Programming\Exercise file.exe)

```
Input starting number of range: 3
Input ending number of range : 8
The prime numbers between 3 and 8 are :
3 5 7

Process returned 10 (0xA)   execution time : 4.439 s
Press any key to continue.
```

The program successfully identifies the prime numbers 3, 5, and 7 within the specified range of 3 to 8.

Q13. Write a program in C to check whether a number is a palindrome or not.



The screenshot displays the Code::Blocks IDE with a C program designed to check if a number is a palindrome. The program uses a while loop to reverse the digits of the input number and then compares the reversed number with the original. A separate console window shows the program's execution with the input '33' and the output '33 is a palindrome'.

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

Execution Output:

```
Enter an integer: 33
33 is a palindrome.
Process returned 0 (0x0)   execution time : 2.469 s
Press any key to continue.
```

The image displays the Code-Blocks IDE interface, showing the source code of a C program and its execution output.

Source Code (Exercise file.c):

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  char *decimal_to_binary(int);
4  char *decimal_to_binary(int dn)
5  {
6      int i, j, temp;
7      char *ptr;
8      temp = 0;
9      ptr = (char*)malloc(32+1);
10     for (i = 31; i >= 0; i--)
11     {
12         j = dn >> i;
13         if (j < 1)
14             *(ptr+temp) = 1 + '0';
15         else
16             *(ptr+temp) = 0 + '0';
17         temp++;
18     }
19     *(ptr+temp) = '\0';
20     return ptr;
21 }
22
23 int main()
24 {
25     int dn;
26     char *ptr;
27     printf("Input a decimal number: ");
28     scanf("%d", &dn);
29     ptr = decimal_to_binary(dn);
30     printf("Binary number equivalent to said decimal number is: %s", ptr);
31     free(ptr);
32     return 0;
33 }
```

Execution Output (Exercise file.exe):

```
Binary number equivalent to said decimal number is: 00000000000000000000000000000000
001001
Process returned 0 (0x0)   execution time : 5.321 s
Press any key to continue.
```

Q15. Write a program in c to find the Sum of GP series.

The screenshot shows the Code::Blocks IDE with a C program open. The program calculates the sum of a Geometric Progression (GP) series. It prompts the user for the first number, the number of terms, and the common ratio. It then displays the individual terms of the series and the total sum.

```
1  #include <stdio.h>
2  #include <math.h>
3
4  void main()
5  {
6      float gl,cr,i,n,j;
7      float nterm,gpn;
8      float sum=0;
9      printf("\n\n Find the Sum of GP series:\n ");
10     printf("Input the first number of the G.P. series: ");
11     scanf("%f",&gl);
12     printf("Input the number or terms in the G.P. series: ");
13     scanf("%f",&nterm);
14     printf("Input the common ratio of G.P. series: ");
15     scanf("%f",&cr);
16     printf("\nThe numbers for the G.P. series:\n ");
17     printf("%f ",gl);
18     sum=gl;
19     for(j=1;j<nterm;j++)
20     {
21         gpn=gl*pow(cr,j);
22         sum=sum+gpn;
23         printf("%f ",gpn);
24     }
25
26     printf("\nThe Sum of the G.P. series : %f\n\n",sum);
27 }
28
```

The output window shows the following text:

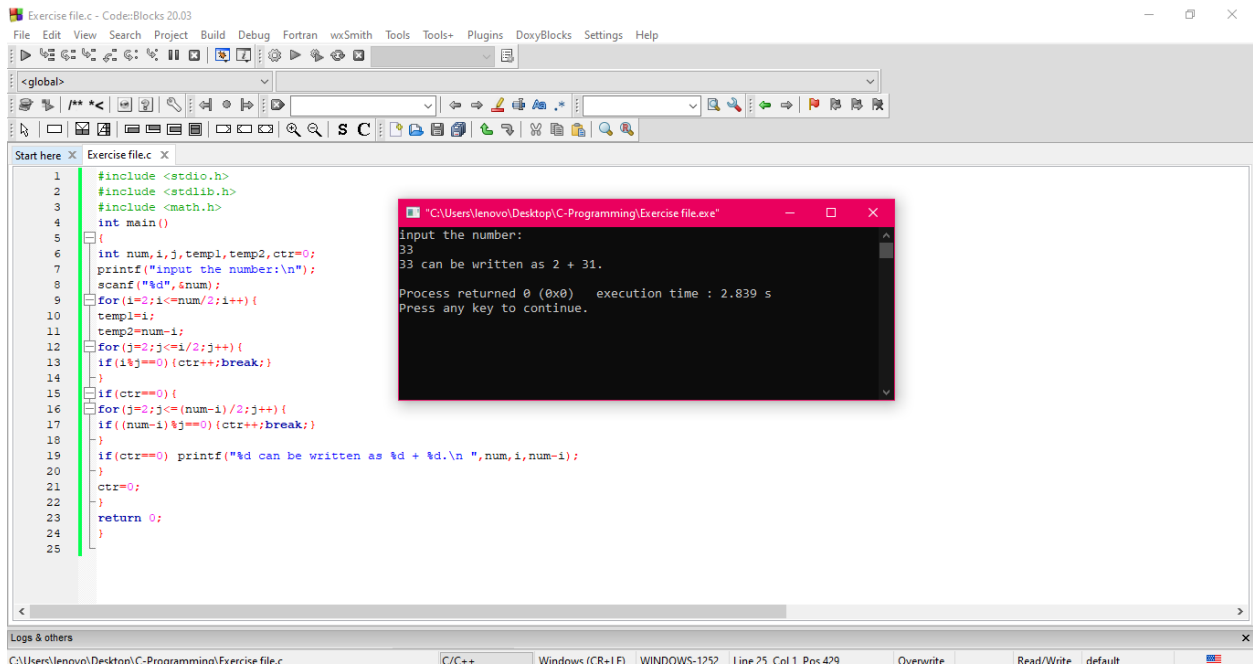
```
Find the Sum of GP series.:
Input the first number of the G.P. series: 3
Input the number or terms in the G.P. series: 8
Input the common ratio of G.P. series: 2

The numbers for the G.P. series:
3.000000 6.000000 12.000000 24.000000 48.000000 96.000000 192.000000 384.000000

The Sum of the G.P. series : 765.000000

Process returned 42 (0x2A)   execution time : 90.096 s
Press any key to continue.
```

Q16. Write a program in C to Check Whether a Number can be Express as Sum of Two Prime Numbers.



```
Exercise file.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

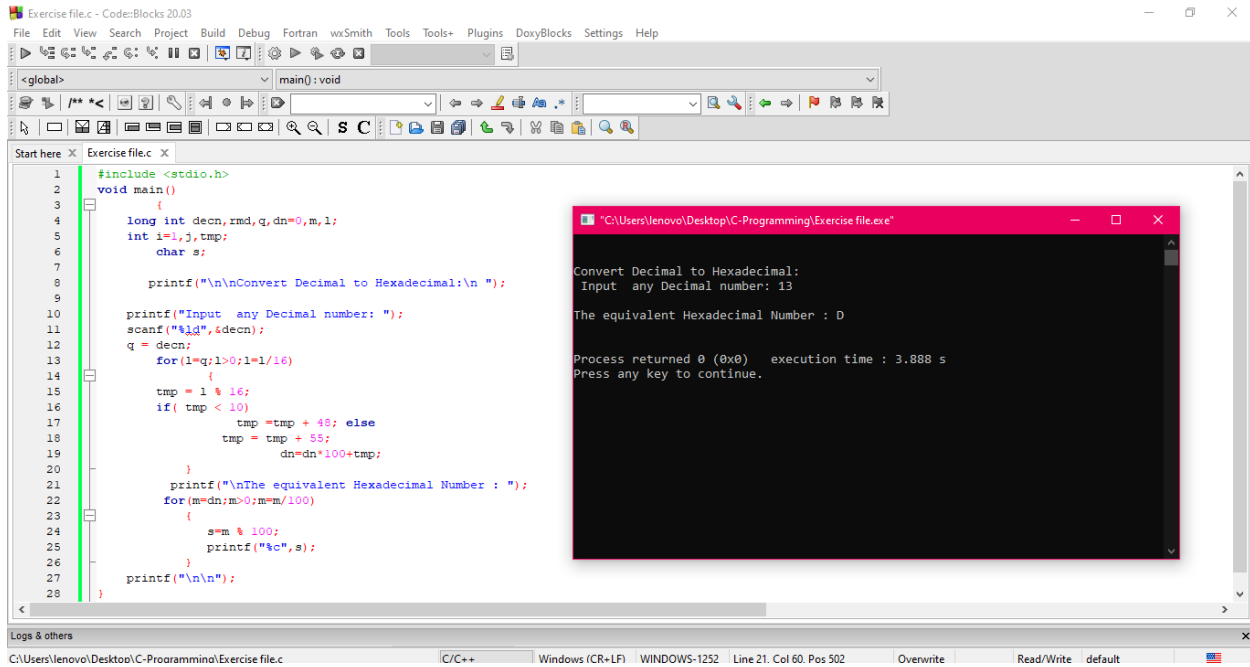
Start here X Exercise file.c X
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  int main()
5  {
6  int num,i,j,temp1,temp2,ctr=0;
7  printf("input the number:\n");
8  scanf("%d",&num);
9  for(i=2;i<=num/2;i++){
10 temp1=i;
11 temp2=num-i;
12 for(j=2;j<=1/2;j++){
13 if(i%j==0){ctr++;break;}
14 }
15 if(ctr==0){
16 for(j=2;j<=(num-1)/2;j++){
17 if((num-1)%j==0){ctr++;break;}
18 }
19 if(ctr==0) printf("%d can be written as %d + %d.\n",num,i,num-i);
20 }
21 ctr=0;
22 }
23 return 0;
24 }
25 }

"C:\Users\lenovo\Desktop\C-Programming\Exercise file.exe"
input the number:
33
33 can be written as 2 + 31.
Process returned 0 (0x0) execution time : 2.839 s
Press any key to continue.
```

Logs & others

C:\Users\lenovo\Desktop\C-Programming\Exercise file.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 25 Col 1 Pos 429 Overwrite Read/Write default

Q17. Write a program in C to convert a decimal number to hexadecimal
Test Data.



The screenshot displays a C code editor window titled "Exercise file.c - Code::Blocks 20.03". The code is as follows:

```
1  #include <stdio.h>
2  void main()
3  {
4      long int decn,rmd,q,dn=0,m,1;
5      int i=1,j,tmp;
6      char s;
7
8      printf("\n\nConvert Decimal to Hexadecimal:\n ");
9
10     printf("Input any Decimal number: ");
11     scanf("%ld",&decn);
12     q = decn;
13     for(i=q;1>0;i=1/16)
14     {
15         tmp = i % 16;
16         if( tmp < 10)
17             tmp =tmp + 48; else
18             tmp = tmp + 55;
19         dn=dn*100+tmp;
20     }
21     printf("\nThe equivalent Hexadecimal Number : ");
22     for(m=dn;m>0;m=m/100)
23     {
24         s=m % 100;
25         printf("%c",s);
26     }
27     printf("\n\n");
28 }
```

Overlaid on the editor is a terminal window titled "C:\Users\lenovo\Desktop\C-Programming\Exercise file.exe". It shows the program's execution:

```
Convert Decimal to Hexadecimal:
Input any Decimal number: 13

The equivalent Hexadecimal Number : D

Process returned 0 (0x0)   execution time : 3.888 s
Press any key to continue.
```

The status bar at the bottom indicates the file path "C:\Users\lenovo\Desktop\C-Programming\Exercise file.c", the editor is "C/C++", the window is "Windows (CR+LF)", the file encoding is "WINDOWS-1252", the cursor is at "Line 21. Col 60. Pos 502", and the mode is "Overwrite".