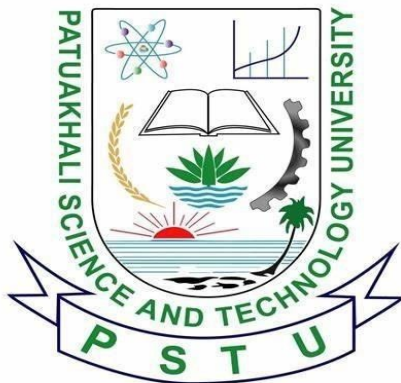


PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY



Course Code: CIT-112

SUBMITTED TO:

MD Mahbubur Rahman Sir

Department of Computer Science And Communication

Engineering

Faculty of Computer Science And Engineering

SUBMITTED BY:

Name: MD Noushad Bhuiyan

ID: 20210238, Registration No: 10165

Faculty of Computer Science and Engineering

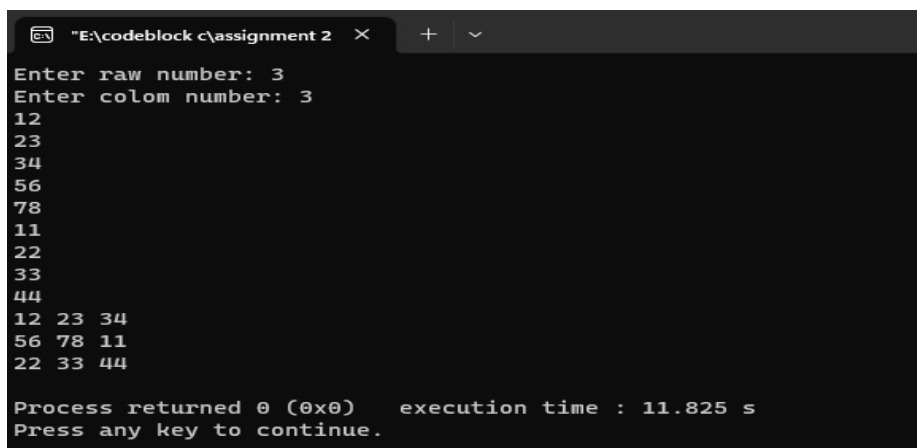
Date of submission: 22-3-2023

1. 2D Array Scanning value

```
#include<stdio.h>
int main()
{
    int n1,n2,a[100][100],i,j;
    printf("Enter raw number: ");
    scanf("%d",&n1);
    printf("Enter colom number: ");
    scanf("%d",&n2);

    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }

    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            printf("%d ",a[i][j]);
        }
        printf("\n");
    }
}
```



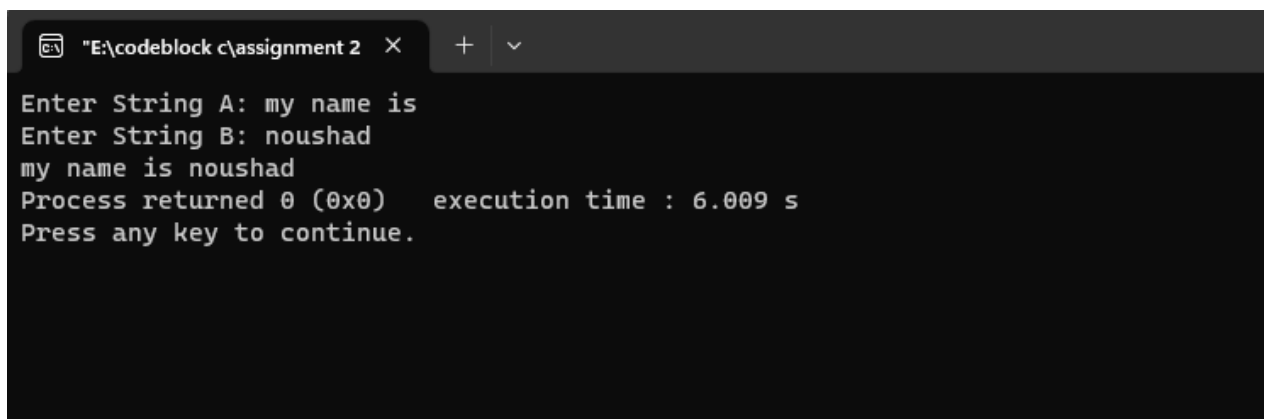
The screenshot shows a code editor window titled "E:\codeblock c\assignment 2". The program has been executed, and the output is displayed in a terminal window. The user has entered '3' for both the row and column numbers. The program then scans the values for a 3x3 array. The output shows the values entered row by row: 12, 23, 34 for the first row; 56, 78, 11 for the second row; and 22, 33, 44 for the third row. At the bottom, it shows "Process returned 0 (0x0) execution time : 11.825 s" and "Press any key to continue."

```
"E:\codeblock c\assignment 2" X + -
Enter raw number: 3
Enter colom number: 3
12
23
34
56
78
11
22
33
44
12 23 34
56 78 11
22 33 44

Process returned 0 (0x0)   execution time : 11.825 s
Press any key to continue.
```

2. Add another string without using strcat function

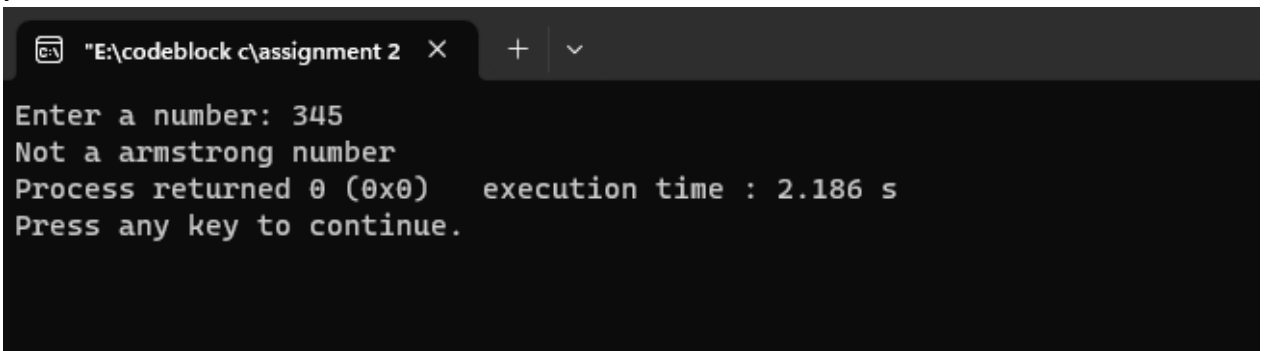
```
#include<stdio.h>
int main()
{
    char len=0,a[1000],b[1000],i,j;
    printf("Enter String A: ");
    gets(a);
    printf("Enter String B: ");
    gets(b);
    i=0;
    j=0;
    while(a[i]!='\0')
    {
        len++;
        i++;
    }
    while(b[j]!=0)
    {
        a[len+j]=b[j];
        j++;
    }
    printf("%s",a);
}
```



```
E:\codeblock c\assignment 2  X  +  v
Enter String A: my name is
Enter String B: noushad
my name is noushad
Process returned 0 (0x0)   execution time : 6.009 s
Press any key to continue.
```

3. Armstrong Number

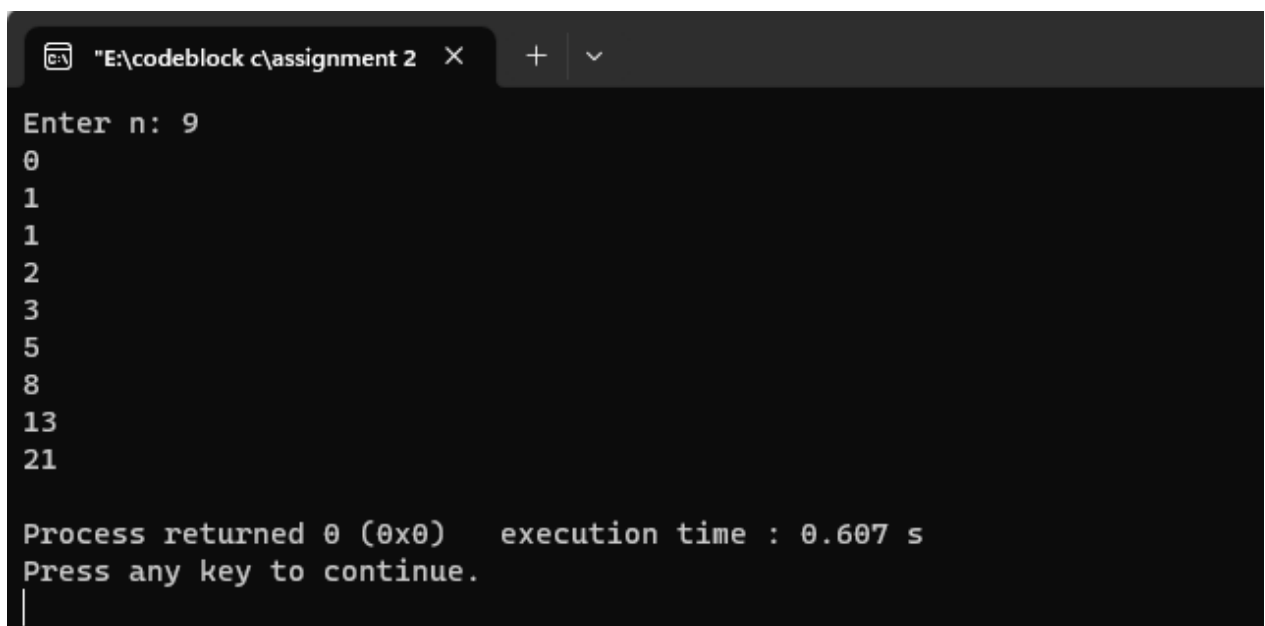
```
#include<stdio.h>
int main()
{
    int n,num,temp=0,r;
    printf("Enter a number: ");
    scanf("%d",&n);
    num=n;
    while(num!=0)
    {
        r=num%10;
        temp=temp+(r*r*r);
        num=num/10;
    }
    if(temp==n)
        printf("The number is a armstrong number");
    else
        printf("Not a armstrong number");
    return 0;
}
```



```
"E:\codeblock c\assignment 2" X + v
Enter a number: 345
Not a armstrong number
Process returned 0 (0x0) execution time : 2.186 s
Press any key to continue.
```

4. Array Fibonacci number

```
#include<stdio.h>
int main()
{
    int sum=0,i,n,a[100];
    printf("Enter n: ");
    scanf("%d",&n);
    a[0]=0;
    a[1]=1;
    printf("%d\n",a[0]);
    printf("%d\n",a[1]);
    for(i=2;i<n;i++)
    {
        a[i]=a[i-1]+a[i-2];
        printf("%d\n",a[i]);
    }
}
```



The screenshot shows a code editor window with the title "E:\codeblock c\assignment 2". The code is the same as the one above. The output of the program is displayed in the console, showing the Fibonacci sequence for n=9: 0, 1, 1, 2, 3, 5, 8, 13, 21. The process returned 0 (0x0) and the execution time was 0.607 s. The prompt "Press any key to continue." is shown at the bottom.

```
Enter n: 9
0
1
1
2
3
5
8
13
21

Process returned 0 (0x0)   execution time : 0.607 s
Press any key to continue.
|
```

5. Array matrix Sub

```
#include<stdio.h>
int main()
{
    int i,j,n1,n2,a[100][100],b[100][100],c[100][100];
    printf("Enter the raw number: ");
    scanf("%d",&n1);
    printf("Enter the columb number: ");
    scanf("%d",&n2);
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Matrix A:\n");
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
            {printf("A[%d][%d] = %d ",i,j,a[i][j]);
        }
        printf("\n");
    }
    //end of matrix a
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
    printf("Matrix B:\n");
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            printf("B[%d][%d] = %d ",i,j,a[i][j]);
        }
    }
```

```

        printf("\n");
    }
    //end of matrix b;
    printf("\n\nMatrix A - Matrix B = \n");
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            c[i][j]= a[i][j]-b[i][j];

            printf("[%d][%d] = %d ",i,j,c[i][j]);

        }
        printf("\n");
    }
}

```

```

E:\codeblock c\assignment 2
Enter the raw number: 2
Enter the columb number: 2
12
23
43
44
Matrix A:
A[0][0] = 12 A[0][1] = 23
A[1][0] = 43 A[1][1] = 44
11
43
22
65
Matrix B:
B[0][0] = 12 B[0][1] = 23
B[1][0] = 43 B[1][1] = 44

Matrix A - Matrix B =
[0][0] = 1 [0][1] = -20
[1][0] = 21 [1][1] = -21

Process returned 0 (0x0)   execution time : 13.725 s
Press any key to continue.

```

6. Array matrix Sum

```
#include<stdio.h>
int main()
{
    int i,j,n1,n2,a[100][100],b[100][100],c[100][100];
    printf("Enter the raw number: ");
    scanf("%d",&n1);
    printf("Enter the columb number: ");
    scanf("%d",&n2);
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Matrix A:\n");
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
            {printf("A[%d][%d] = %d ",i,j,a[i][j]);
        }
        printf("\n");
    }
    //end of matrix a
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
    printf("Matrix B:\n");
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            printf("B[%d][%d] = %d ",i,j,a[i][j]);
        }
    }
```



```

        printf("\n");
    }
    //end of matrix b;
    printf("\n\nMatrix A - Matrix B = \n");
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            c[i][j]= a[i][j]+b[i][j];

            printf("[%d][%d] = %d ",i,j,c[i][j]);

        }
        printf("\n");
    }
}

```

```

E:\codeblock c\assignment 2 x + v
Enter the row number: 2
Enter the columb number: 2
11
22
33
44
Matrix A:
A[0][0] = 11 A[0][1] = 22
A[1][0] = 33 A[1][1] = 44
44
33
22
11
Matrix B:
B[0][0] = 11 B[0][1] = 22
B[1][0] = 33 B[1][1] = 44

Matrix A + Matrix B =
[0][0] = 55 [0][1] = 55
[1][0] = 55 [1][1] = 55

Process returned 0 (0x0)   execution time : 8.732 s
Press any key to continue.

```

7. Array matrix sum of Diagonal digits

```
#include<stdio.h>
int main()
//The sum of diagonal elements
{
    int n1,n2,i,j,a[100][100],sum=0;
    printf("Enter raw number : ");
    scanf("%d",&n1);
    printf("Enter columb number : ");
    scanf("%d",&n2);
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Matrix A:\n");
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            printf("A[%d][%d] = %d ",i,j,a[i][j]);

        }
        printf("\n");
    }
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            if(i==j )
            {
                sum=sum+a[i][j];
            }
        }
    }
    printf("The sum of diagonal elements are: %d",sum);
```

}

```
"E:\codeblock c\assignment 2" × + v
Enter raw number : 3
Enter columb number : 3
12
33
22
34
54
32
12
21
22
Matrix A:
A[0][0] = 12  A[0][1] = 33  A[0][2] = 22
A[1][0] = 34  A[1][1] = 54  A[1][2] = 32
A[2][0] = 12  A[2][1] = 21  A[2][2] = 22
The sum of diagonal elements are: 88
Process returned 0 (0x0)   execution time : 13.422 s
Press any key to continue.
```

8. Array matrix sum of lower triangle digits of matrix

```
#include<stdio.h>
int main()
//The sum of lower Triangle digits
{
    int n1,n2,i,j,a[100][100],sum=0;
    printf("Enter raw number : ");
    scanf("%d",&n1);
    printf("Enter columb number : ");
    scanf("%d",&n2);
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Matrix A:\n");
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            printf("A[%d][%d] = %d ",i,j,a[i][j]);

        }
        printf("\n");
    }
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            if(i==j || i>j)
            {
                sum=sum+a[i][j];
            }
        }
    }
    printf("The sum of lower Triangle elements are: %d",sum);
}
```

}

```
"E:\codeblock c\assignment 2" X + v
Enter raw number : 3
Enter columb number : 3
12
33
22
44
54
21
23
44
22
Matrix A:
A[0][0] = 12 A[0][1] = 33 A[0][2] = 22
A[1][0] = 44 A[1][1] = 54 A[1][2] = 21
A[2][0] = 23 A[2][1] = 44 A[2][2] = 22
The sum of lower Triangle elements are: 199
Process returned 0 (0x0) execution time : 11.367 s
Press any key to continue.
```

9. Array matrix sum of upper Triangle digits

```
#include<stdio.h>
int main()
//The sum of Upper triangle digits
{
    int n1,n2,i,j,a[100][100],sum=0;
    printf("Enter raw number : ");
    scanf("%d",&n1);
    printf("Enter columb number : ");
    scanf("%d",&n2);
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Matrix A:\n");
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            printf("A[%d][%d] = %d ",i,j,a[i][j]);

        }
        printf("\n");
    }
    for(i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            if(i==j || i<j)
            {
                sum=sum+a[i][j];
            }
        }
    }
    printf("The sum of Upper triangle elements are: %d",sum);
}
```

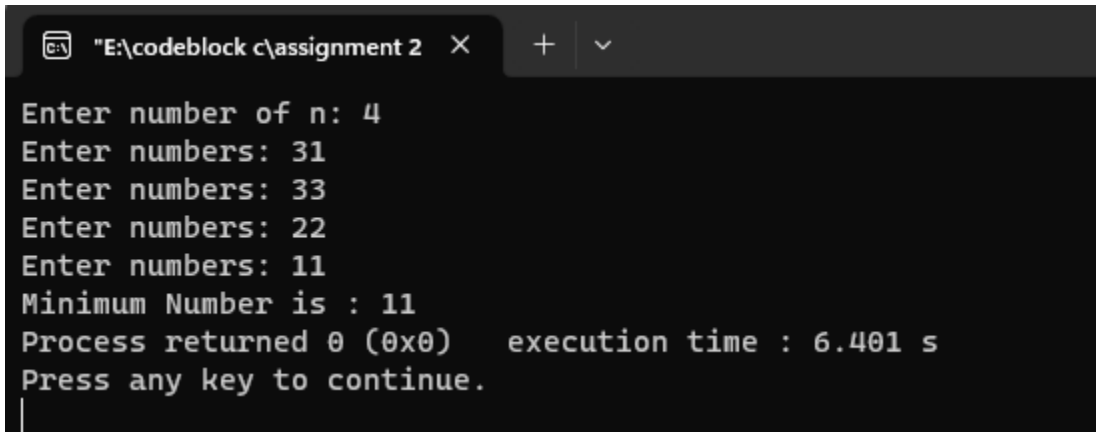
}

```
"E:\codeblock c\assignment 2" X + v
Enter raw number : 3
Enter columb number : 3
12
23
34
45
54
43
32
21
23
Matrix A:
A[0][0] = 12 A[0][1] = 23 A[0][2] = 34
A[1][0] = 45 A[1][1] = 54 A[1][2] = 43
A[2][0] = 32 A[2][1] = 21 A[2][2] = 23
The sum of Upper triangle elements are: 189
Process returned 0 (0x0) execution time : 15.648 s
Press any key to continue.
```

10. Array minimum number

```
#include<stdio.h>
int main()
{

    int n,a[5],i;
    printf("Enter number of n: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter numbers: ");
        scanf("%d",&a[i]);
    }
    int min= a[0];
    for(i=1;i<n;i++)
    {
        if(min>a[i])
        {
            min=a[i];
        }
    }
    printf("Minimum Number is : %d",min);
}
```

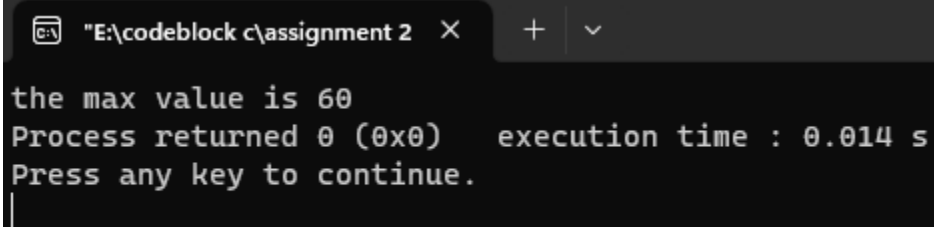


```
"E:\codeblock c\assignment 2" X + v
Enter number of n: 4
Enter numbers: 31
Enter numbers: 33
Enter numbers: 22
Enter numbers: 11
Enter numbers: 11
Minimum Number is : 11
Process returned 0 (0x0)   execution time : 6.401 s
Press any key to continue.
|
```


11. Array max value with using function

```
#include<stdio.h>
int maximum(int x[])
{
    int i,max;
    max=x[0];
    for(i=0;i<5;i++)
    {
        if(max<x[i])
            max=x[i];
    }
    return max;
}

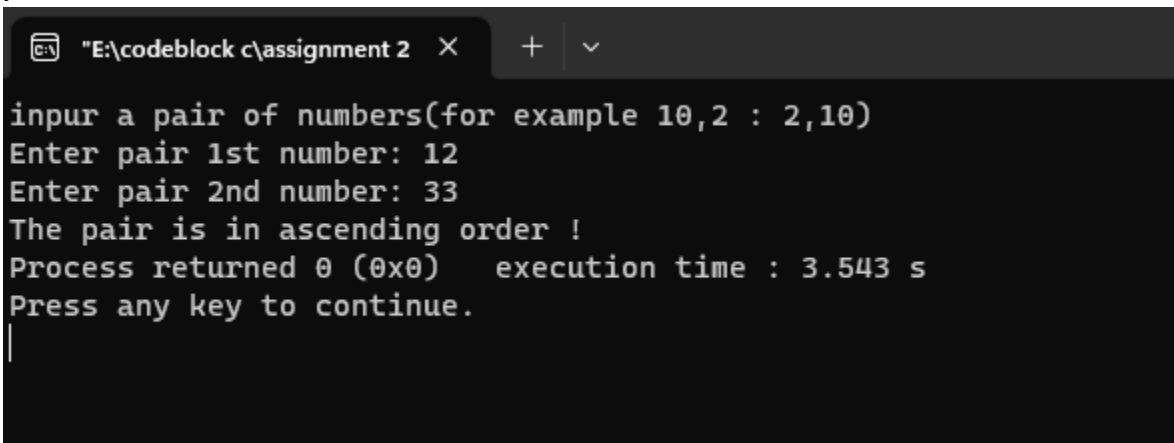
int main()
{
    int a[]={20,30,40,50,60};
    int maxi = maximum(a);
    printf("the max value is %d",maxi);
}
```



```
"E:\codeblock c\assignment 2" X + v
the max value is 60
Process returned 0 (0x0)   execution time : 0.014 s
Press any key to continue.
|
```

12.Ascending to descending Form

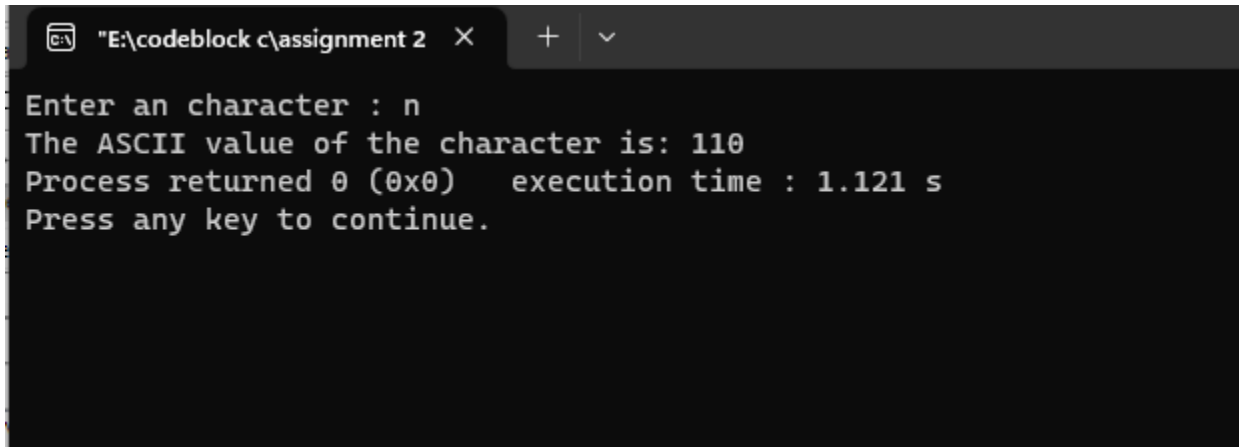
```
#include<stdio.h>
int main()
{ int i,j;
  printf("inpur a pair of numbers(for example 10,2 : 2,10)\n");
  printf("Enter pair 1st number: ");
  scanf("%d",&i);
  printf("Enter pair 2nd number: ");
  scanf("%d",&j);
  if(i>j)
    {printf("The pair is in descending order !");}
  else
    {printf("The pair is in ascending order !");}
}
```



```
"E:\codeblock c\assignment 2" X + v
inpur a pair of numbers(for example 10,2 : 2,10)
Enter pair 1st number: 12
Enter pair 2nd number: 33
The pair is in ascending order !
Process returned 0 (0x0)   execution time : 3.543 s
Press any key to continue.
|
```

13.ASCII Value to Integer number

```
#include<stdio.h>
int main()
{ char n;
  printf("Enter an character : ");
  scanf("%c",&n);
  printf("The ASCII value of the character is: %d",n);
  return 0;
}
```

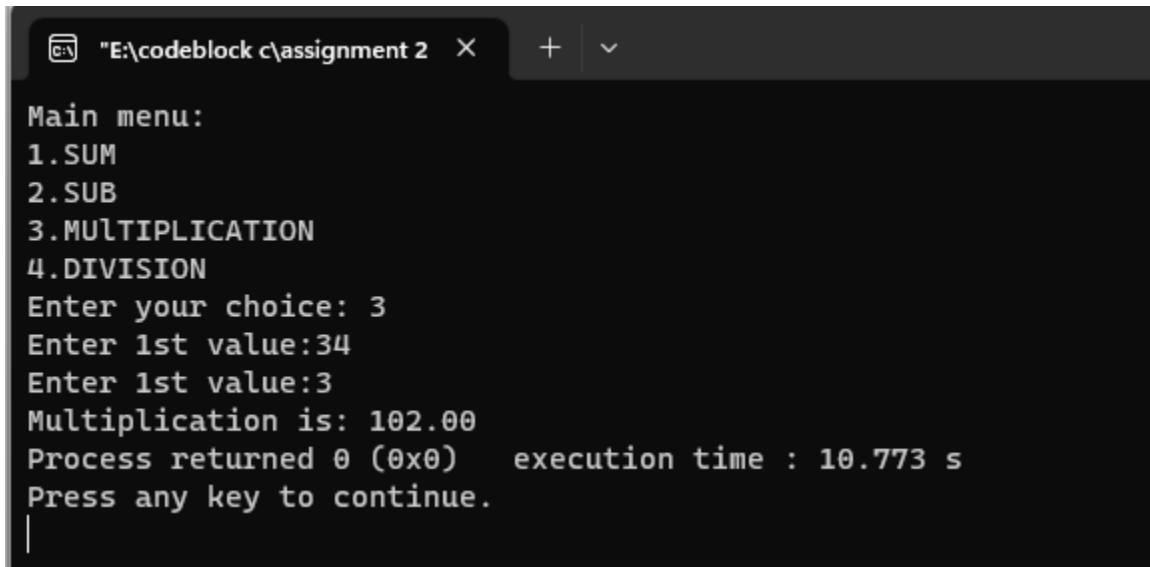


```
"E:\codeblock c\assignment 2" × + v
Enter an character : n
The ASCII value of the character is: 110
Process returned 0 (0x0)   execution time : 1.121 s
Press any key to continue.
```

14. Calculator (Sum, Sub, Multiplication, Division)

```
#include<stdio.h>
int main()
{
    float sum,sub,mul,div;
    int a,b,n;
    printf("Main menu:\n1.SUM\n2.SUB\n3.MUITIPLICATION\n4.DIVISION\nEnter your
choice: ");
    scanf("%d",&n);
    switch(n)
    {
        case 1:
            {printf("Enter 1st value:");
            scanf("%d",&a);
            printf("Enter 1st value:");
            scanf("%d",&b);
            sum=a+b;
            printf("Sum is: %0.2f",sum);}
            break;
        case 2:
            {printf("Enter 1st value:");
            scanf("%d",&a);
            printf("Enter 1st value:");
            scanf("%d",&b);
            sub=a-b;
            printf("Sub is: %0.2f",sub);}
            break;
        case 3:
            {printf("Enter 1st value:");
            scanf("%d",&a);
            printf("Enter 1st value:");
            scanf("%d",&b);
            mul=a*b;
            printf("Multiplication is: %0.2f",mul);}
            break;
        case 4:
            {printf("Enter 1st value:");
            scanf("%d",&a);
            printf("Enter 1st value:");
```

```
scanf("%d",&b);  
div=a/b;  
printf("Division is: %0.2f",div);}  
break;  
  
}  
}
```

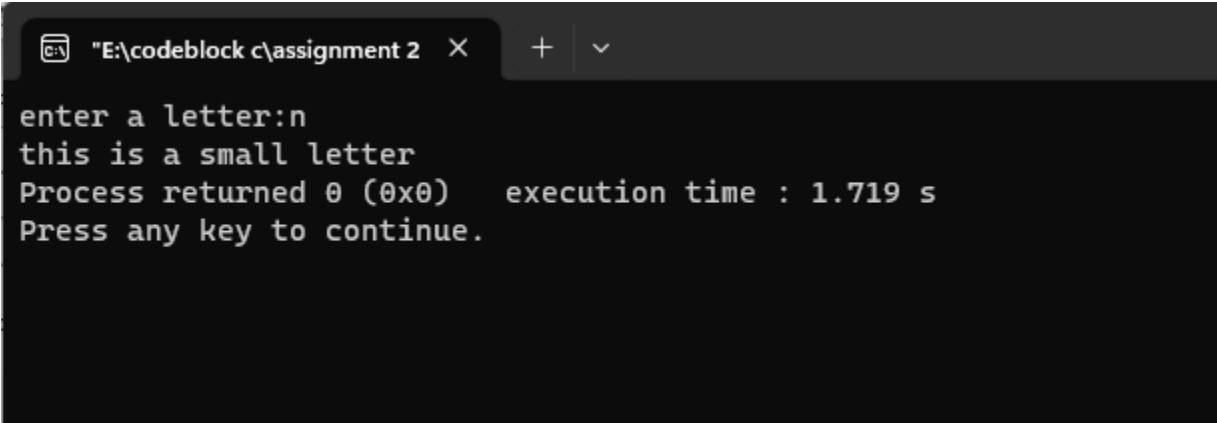


```
"E:\codeblock c\assignment 2" X + v  
Main menu:  
1.SUM  
2.SUB  
3.MULTIPLICATION  
4.DIVISION  
Enter your choice: 3  
Enter 1st value:34  
Enter 1st value:3  
Multiplication is: 102.00  
Process returned 0 (0x0) execution time : 10.773 s  
Press any key to continue.  
|
```

15.Capital and small letter recognition

```
#include<stdio.h>
int main()
{
    char ch;
    printf("enter a letter:");
    scanf("%c",&ch);
    if(ch>='a'&&ch<='z')
        printf("this is a small letter");
    else if(ch>='A'&&ch<='Z')
        printf("this is a capital letter");

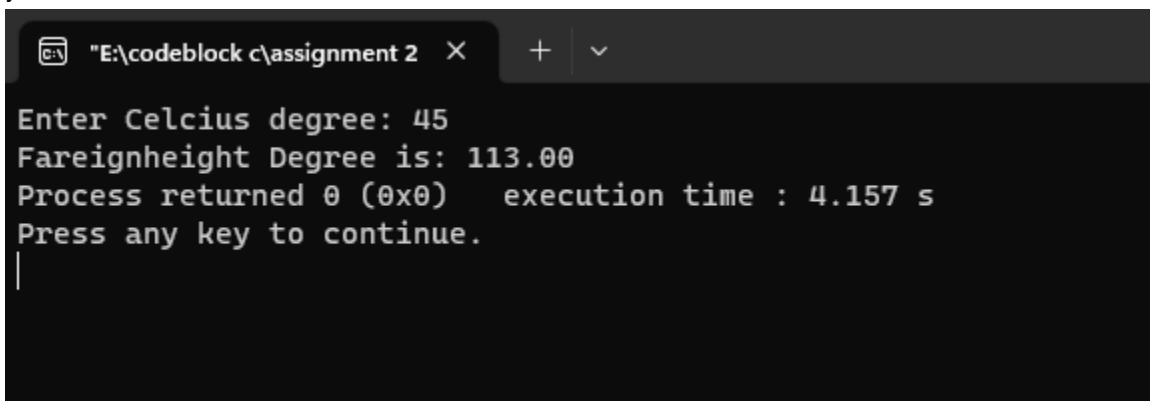
    return 0;
}
```



```
"E:\codeblock c\assignment 2" X + v
enter a letter:
this is a small letter
Process returned 0 (0x0) execution time : 1.719 s
Press any key to continue.
```

16.Celsius to fahrenheit

```
#include<stdio.h>
int main()
{
    float c,f;
    printf("Enter Celcius degree: ");
    scanf("%f",&c);
    f=(((9*c)/5)+32);
    printf("Fareignheight Degree is: %0.2f",f);
}
```

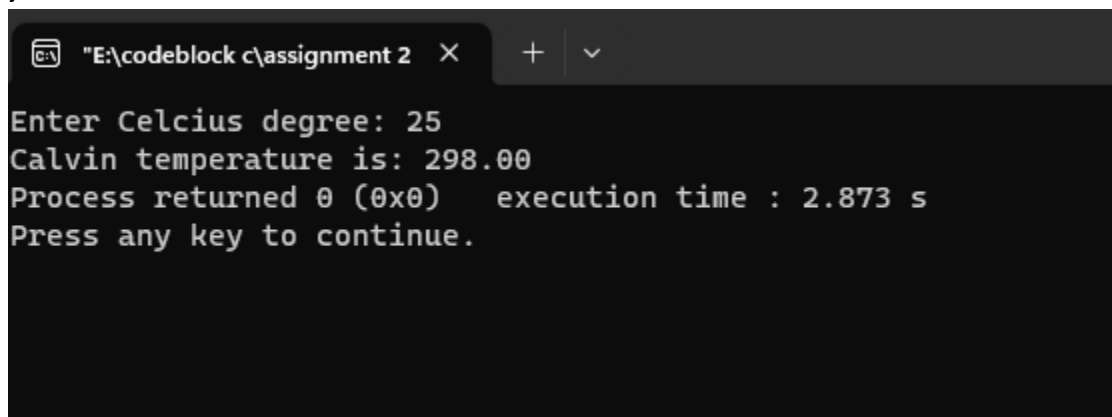


The screenshot shows a Code::Blocks IDE window with the title "E:\codeblock c\assignment 2". The output console displays the following text:

```
Enter Celcius degree: 45
Fareignheight Degree is: 113.00
Process returned 0 (0x0)   execution time : 4.157 s
Press any key to continue.
|
```

17.Celsius to kelvin

```
#include<stdio.h>
int main()
{
    float c,k;
    printf("Enter Celcius degree: ");
    scanf("%f",&c);
    k=(c+273);
    printf("Calvin temperature is: %0.2f",k);
}
```



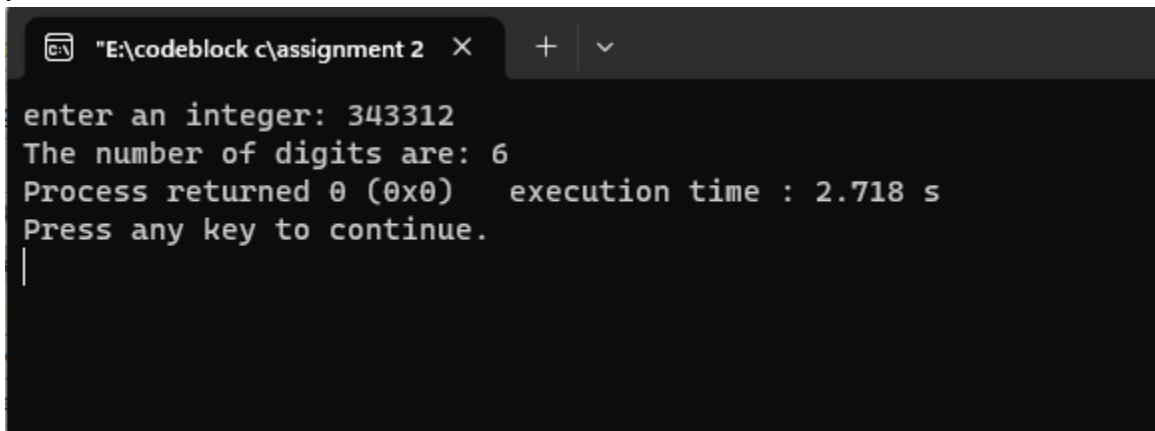
The screenshot shows a code editor window with the title bar "E:\codeblock c\assignment 2". The code is the same as shown in the previous block. The output of the program is displayed in the console area, showing the input "25" and the resulting Kelvin temperature "298.00". The console also shows the process return code "0 (0x0)", the execution time "2.873 s", and a prompt to "Press any key to continue."

```
E:\codeblock c\assignment 2
Enter Celcius degree: 25
Calvin temperature is: 298.00
Process returned 0 (0x0)   execution time : 2.873 s
Press any key to continue.
```


18.Counting the number of a digit in an integer.

```
#include<stdio.h>
int main()
{
    int n,r,sum=0;
    printf("enter an integer: ");
    scanf("%d",&n);
    while(n!=0)
    {

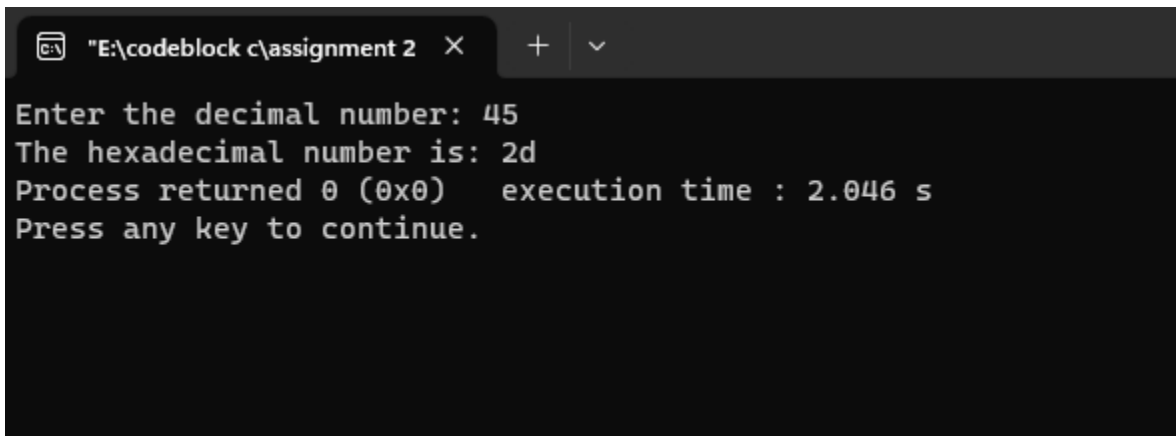
        n=n/10;
        sum++;
    }
    printf("The number of digits are: %d",sum);
    return 0;
}
```

A screenshot of a code editor window titled "E:\codeblock c\assignment 2". The window shows the output of a C program. The user has entered the integer 343312. The program has calculated that there are 6 digits in this integer. The output text is: "enter an integer: 343312", "The number of digits are: 6", "Process returned 0 (0x0) execution time : 2.718 s", and "Press any key to continue." followed by a vertical cursor line.

```
enter an integer: 343312
The number of digits are: 6
Process returned 0 (0x0) execution time : 2.718 s
Press any key to continue.
|
```

19.Decimal to hexadecimal

```
#include<stdio.h>
int main()
{
    int n;
    printf("Enter the decimal number: ");
    scanf("%d",&n);
    printf("The hexadecimal number is: %x",n);
    return 0;
}
```

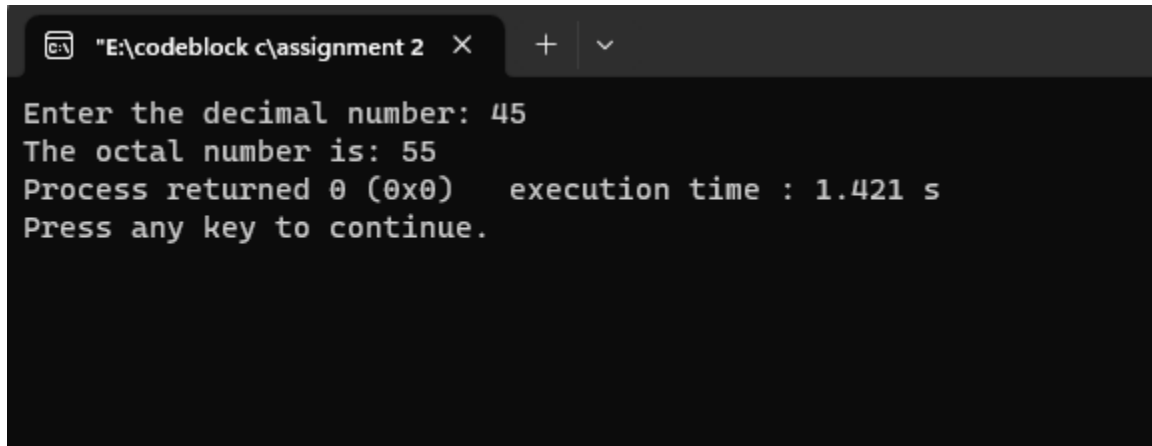


The screenshot shows a code editor window with the title "E:\codeblock c\assignment 2". The output of the program is displayed in a terminal window, showing the input "45" and the output "2d". The terminal also displays the process return code "0 (0x0)" and the execution time "2.046 s". The prompt "Press any key to continue." is visible at the bottom of the terminal output.

```
Enter the decimal number: 45
The hexadecimal number is: 2d
Process returned 0 (0x0)   execution time : 2.046 s
Press any key to continue.
```

20.Decimal to octal

```
#include<stdio.h>
int main()
{
    int n;
    printf("Enter the decimal number: ");
    scanf("%d",&n);
    printf("The octal number is: %o",n);
    return 0;
}
```



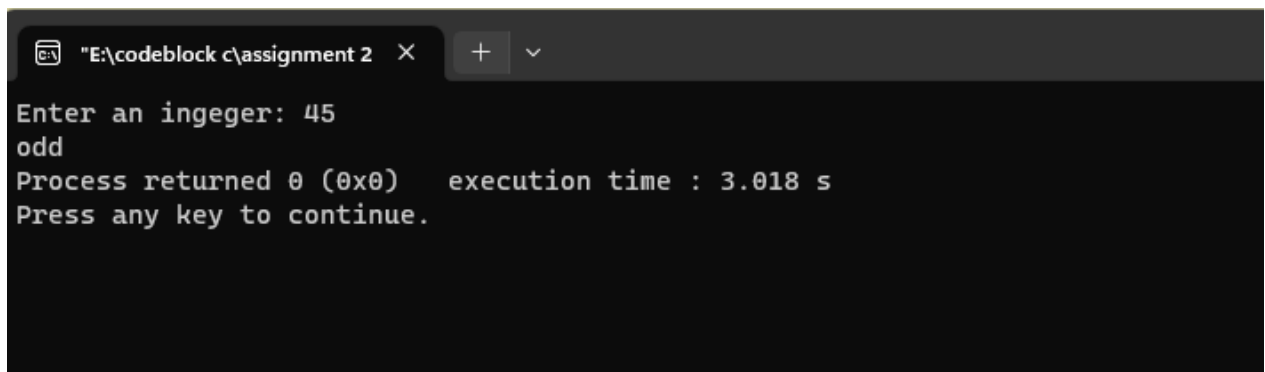
The screenshot shows a code editor window with a single tab titled "E:\codeblock c\assignment 2". The editor contains the C code from the previous block. Below the code, the output of the program is displayed in a terminal-like window. The output shows the user entering the decimal number 45, followed by the program printing the octal number 55. It also displays the process return code as 0 (0x0) and the execution time as 1.421 seconds. The prompt "Press any key to continue." is visible at the bottom of the output window.

```
"E:\codeblock c\assignment 2" X + v
Enter the decimal number: 45
The octal number is: 55
Process returned 0 (0x0)   execution time : 1.421 s
Press any key to continue.
```

21. Even or Odd number

```
#include<stdio.h>
int main()
{ int num;
  printf("Enter an ingeger: ");
  scanf("%d",&num);
  if(num%2==0)
    printf("Even");

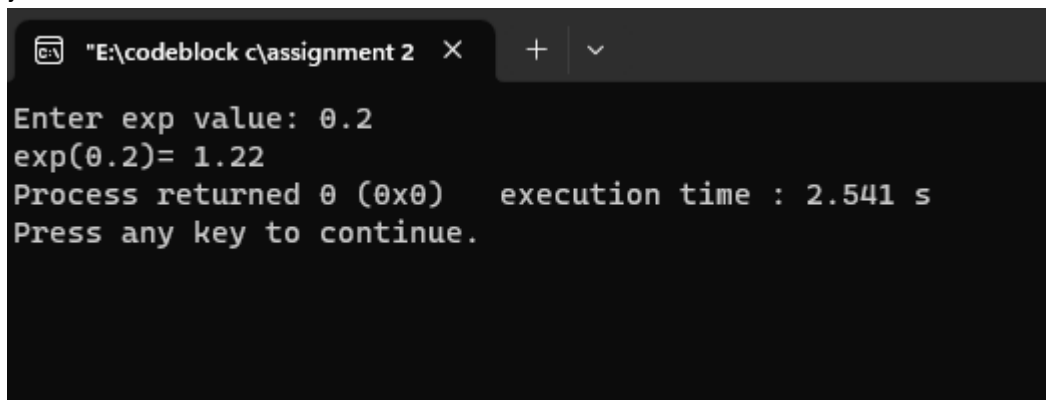
  else if(num!=0)
    printf("odd");
  else
    printf("the number is 0");
  return 0;
}
```



The screenshot shows a Code::Blocks IDE window with the title bar "E:\codeblock c\assignment 2". The console output displays the program's execution: it prompts "Enter an ingeger: 45", outputs "odd", shows "Process returned 0 (0x0) execution time : 3.018 s", and ends with "Press any key to continue.".

22.Exponential Function

```
#include<stdio.h>
int main()
{
    double result,x;
    printf("Enter exp value: ");
    scanf("%lf",&x);
    result=exp(x);
    printf("exp(%0.1f)= %0.2lf",x,result);
    return 0;
}
```

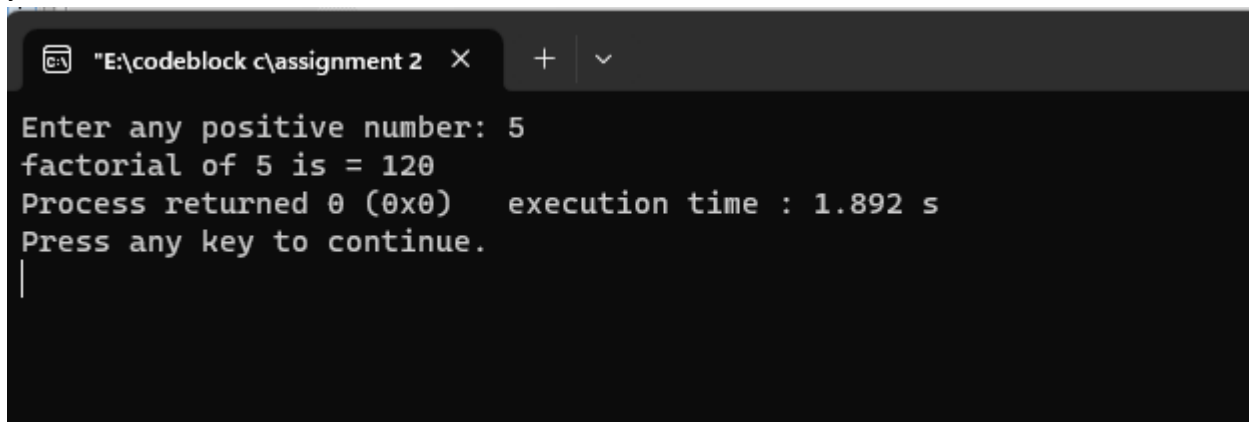


The screenshot shows a code editor window with the title bar "E:\codeblock c\assignment 2". The code is being executed, and the output is displayed in the console. The output shows the user entering the value 0.2, and the program calculating the exponential of 0.2, which is 1.22. The console also shows the process returned 0 (0x0) and the execution time was 2.541 seconds. The prompt "Press any key to continue." is visible at the bottom of the console output.

```
"E:\codeblock c\assignment 2" X + v
Enter exp value: 0.2
exp(0.2)= 1.22
Process returned 0 (0x0)   execution time : 2.541 s
Press any key to continue.
```

23. Factorial of a digit

```
#include<stdio.h>
int main()
{
    int n,i,fact=1;
    printf("Enter any positive number: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        fact=fact*i;
    }
    printf("factorial of %lf is = %d",n,fact);
}
```



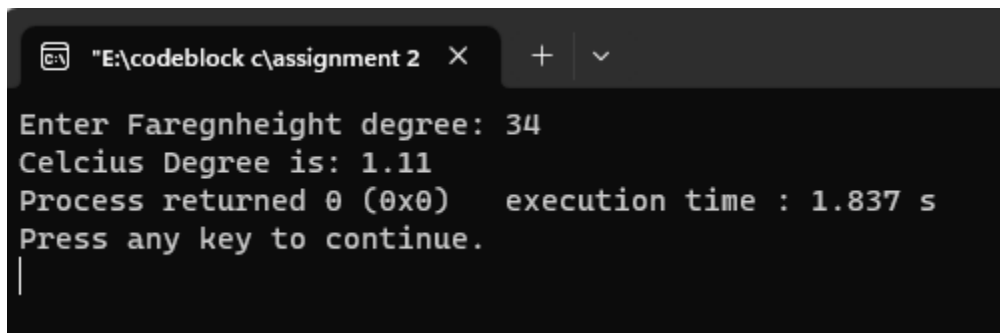
The screenshot shows a terminal window titled "E:\codeblock c\assignment 2". The program prompts the user to "Enter any positive number:" and the user enters "5". The program then outputs "factorial of 5 is = 120". Below this, it shows "Process returned 0 (0x0)" and "execution time : 1.892 s". Finally, it prompts "Press any key to continue." and a cursor is visible on the next line.

```
E:\codeblock c\assignment 2  X  +  v
Enter any positive number: 5
factorial of 5 is = 120
Process returned 0 (0x0)   execution time : 1.892 s
Press any key to continue.
|
```

24.Fahrenheit to celsius

```
#include<stdio.h>
int main()
{
    float c,f;
    printf("Enter Faregnheight degree: ");
    scanf("%f",&f);
    c=(((f-32)*5)/9);
    printf("Celcius Degree is: %0.2f",c);

}
```

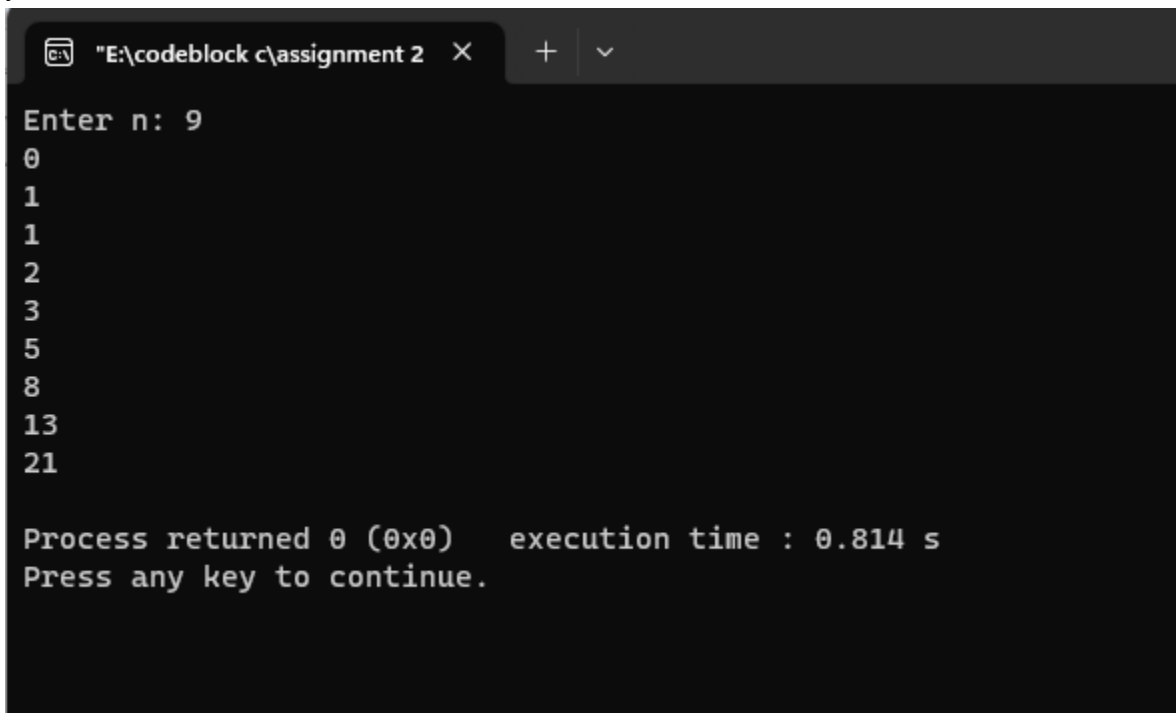


The screenshot shows a code editor window with the title "E:\codeblock c\assignment 2". The output of the program is displayed in a terminal window, showing the input "34" for the Fahrenheit degree, the calculated Celsius degree "1.11", the process return code "0 (0x0)", the execution time "1.837 s", and a prompt to "Press any key to continue." followed by a vertical bar cursor.

```
Enter Faregnheight degree: 34
Celcius Degree is: 1.11
Process returned 0 (0x0)   execution time : 1.837 s
Press any key to continue.
|
```

25.Fibonacci number

```
#include<stdio.h>
int main()
{ //fibonacci number
  int n,i,num1=0,num2=1,fib;
  printf("Enter n: ");
  scanf("%d",&n);
  printf("%d\n",num1);
  printf("%d\n",num2);
  for(i=0;i<=n-3;i++)
  {
    fib=num1+num2;
    num1=num2;
    num2=fib;
    printf("%d\n",fib);
  }
}
```



The screenshot shows a code editor window with the title "E:\codeblock c\assignment 2". The output of the program is displayed in a terminal window. The user enters '9' for 'n'. The program then prints the first 10 Fibonacci numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21. Below the output, it shows "Process returned 0 (0x0) execution time : 0.814 s" and "Press any key to continue."

```
Enter n: 9
0
1
1
2
3
5
8
13
21

Process returned 0 (0x0)   execution time : 0.814 s
Press any key to continue.
```


26.Find the number position

```
#include<stdio.h>
int main()
{

    int position=-1,i,n,a[100],value,pos;
    printf("Enter n: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter numbers: ");
        scanf("%d",&a[i]);

    }
    printf("Enter the number: ");
    scanf("%d",&value);
    for(i=0;i<n;i++){
        if(value==a[i])
        {
            pos=i+1;
            break;
        }
    }
    if(position== -1)
        printf("not found");
    else
    {
        printf("The position of this number is %d",pos);
    }
}
```

27.Area of a triangle using function

```
#include<stdio.h>
double areatriangle(double a,double b);
double main()

{
    double height,weight;
    printf("Enter height: ");
    scanf("%lf",&height);
    printf("Enter weight: ");
    scanf("%lf",&weight);
    double area= areatriangle(height,weight);
    printf("The area of the triangle is : %0.2lf",area);
}

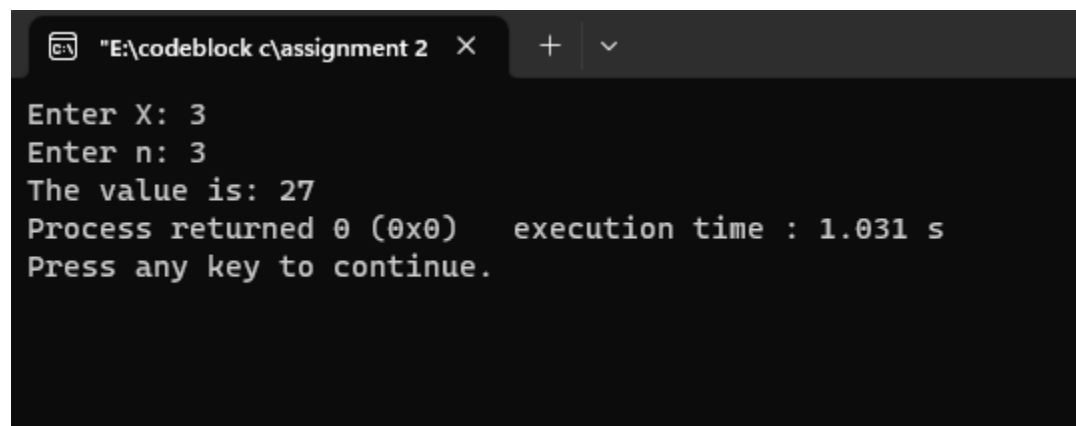
(

    return 0.5*a*b ;
)
```

28.Power value using function

```
#include<stdio.h>
int main()
{
    int a,b;
    printf("Enter X: ");
    scanf("%d",&a);
    printf("Enter n: ");
    scanf("%d",&b);
    int powervalue = value(a,b);
    printf("The value is: %d",powervalue);

}
int value(int x,int n)
{
    return pow(x,n);
}
```

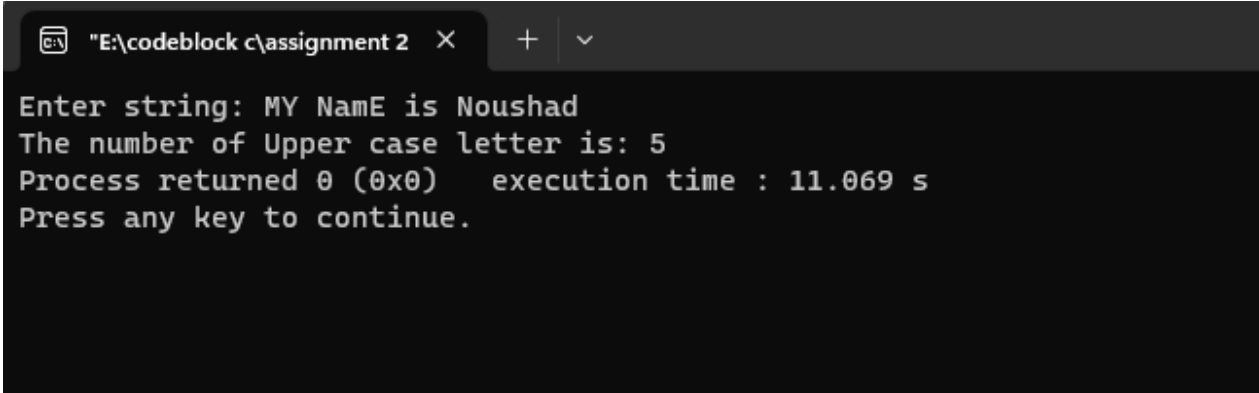


The screenshot shows a terminal window titled "E:\codeblock c\assignment 2". The output of the program is as follows:

```
Enter X: 3
Enter n: 3
The value is: 27
Process returned 0 (0x0)   execution time : 1.031 s
Press any key to continue.
```

29.String using function

```
#include<stdio.h>
int main()
{
    char s[100];
    printf("Enter string: ");
    gets(s);
    int f = string(s);
    printf("The number of Upper case letter is: %d",f);
}
int string(char x[])
{
    int u=0,l=0,i=0,o=0;
    while(x[i]!='\0')
    {
        if(x[i]>=65 && x[i]<=90 )
        { u++;}
        else if(x[i]>=97 && x[i]<=122)
        {
            l++;
        }
        else
        {
            o++;
        }
        i++;
    }
    return u;
}
```

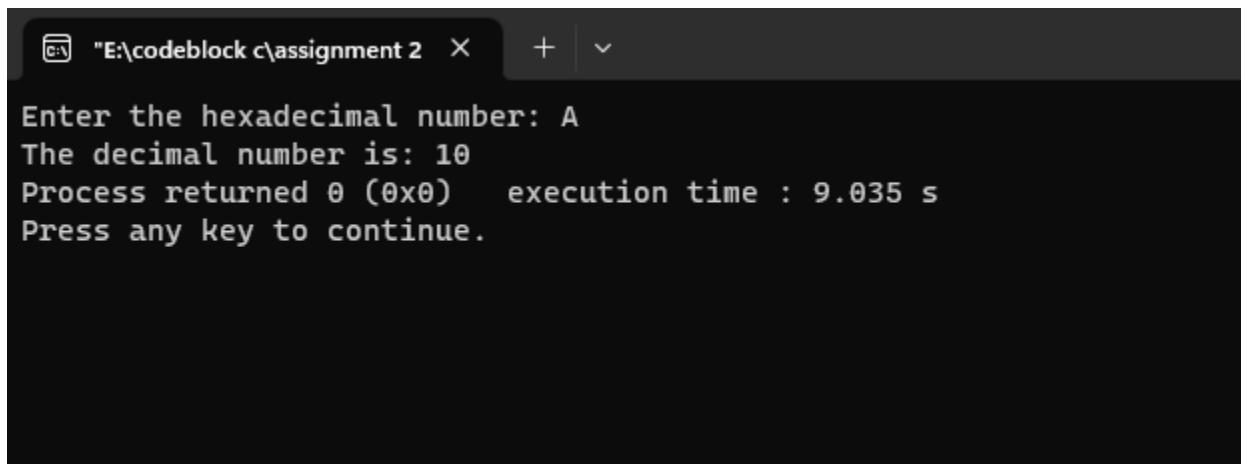


The screenshot shows a code editor window with the title "E:\codeblock c\assignment 2". The output of the program is displayed in a terminal-like window. The user has entered the string "MY NamE is Noushad". The program correctly counts 5 uppercase letters. The execution time is 11.069 seconds. The prompt "Press any key to continue." is shown at the bottom.

```
Enter string: MY NamE is Noushad
The number of Upper case letter is: 5
Process returned 0 (0x0)   execution time : 11.069 s
Press any key to continue.
```

30.Hexadecimal to decimal

```
#include<stdio.h>
int main()
{
    int n;
    printf("Enter the hexadecimal number: ");
    scanf("%x",&n);
    printf("The decimal number is: %d",n);
    return 0;
}
```

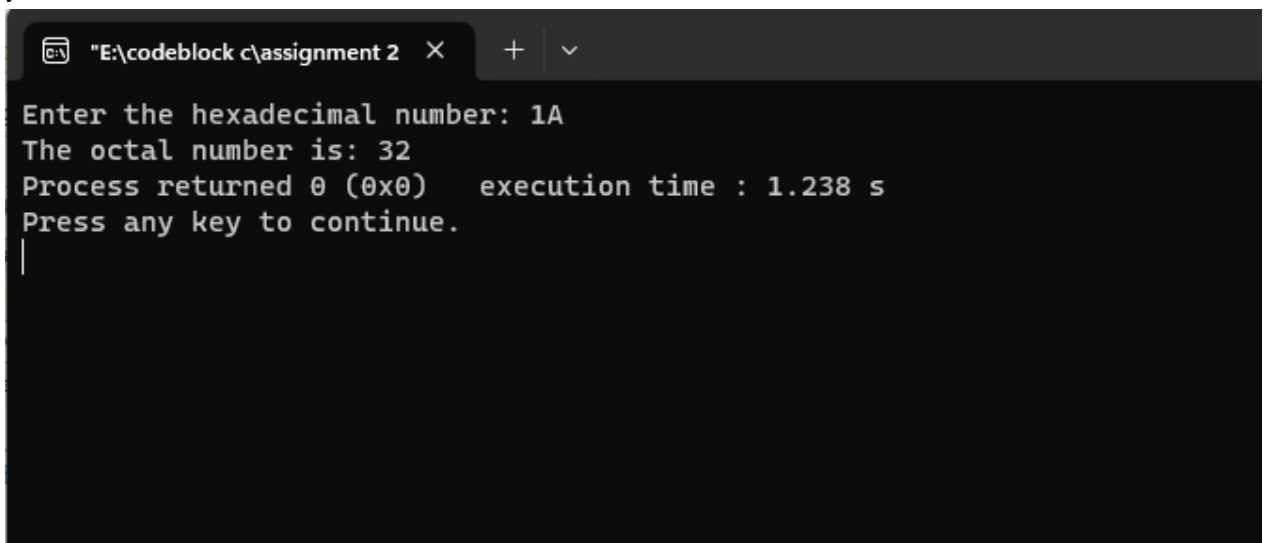


The screenshot shows a code editor window with the title bar "E:\codeblock c\assignment 2". The editor contains the C code for converting a hexadecimal number to decimal. The output of the program is displayed in the console area, showing the prompt "Enter the hexadecimal number: A", the result "The decimal number is: 10", and the execution time "9.035 s". The console also shows the process return code "0 (0x0)" and the instruction "Press any key to continue."

```
E:\codeblock c\assignment 2  X  +  v
Enter the hexadecimal number: A
The decimal number is: 10
Process returned 0 (0x0)   execution time : 9.035 s
Press any key to continue.
```

31.Hexadecimal to octal

```
#include<stdio.h>
int main()
{
    int n;
    printf("Enter the hexadecimal number: ");
    scanf("%x",&n);
    printf("The octal number is: %o",n);
    return 0;
}
```



The screenshot shows a terminal window titled "E:\codeblock c\assignment 2". The program prompts the user to "Enter the hexadecimal number: 1A". The user inputs "1A", and the program outputs "The octal number is: 32". Below this, the terminal shows "Process returned 0 (0x0) execution time : 1.238 s" and "Press any key to continue." followed by a vertical cursor line.

32.Higher number and position from 5 number

```
#include<stdio.h>
int main()
{
    int a,b,c,d,e;
    printf("Enter a: ");
    scanf("%d",&a);

    printf("Enter b: ");
    scanf("%d",&b);

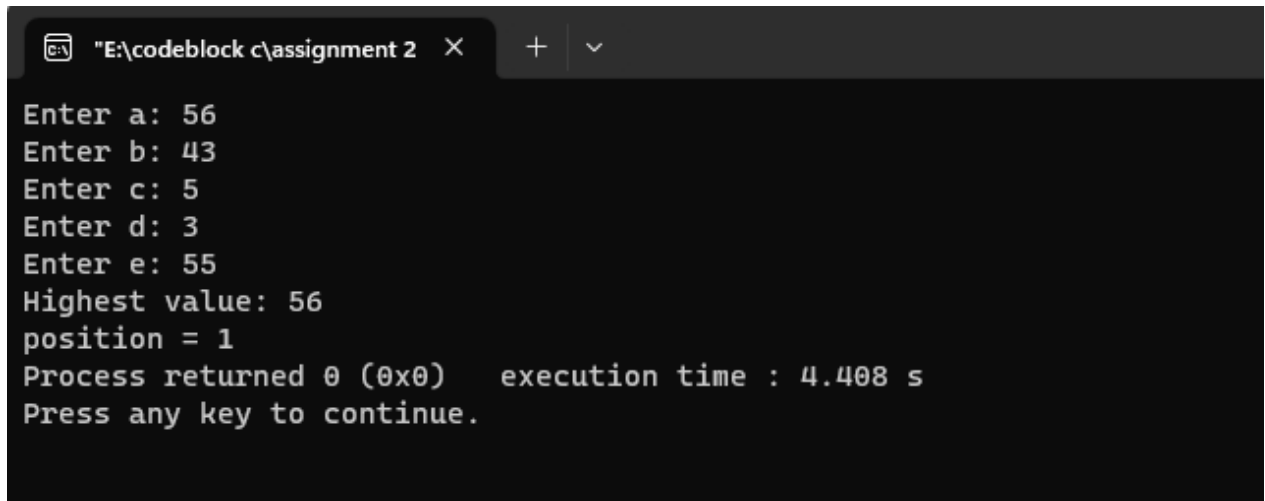
    printf("Enter c: ");
    scanf("%d",&c);

    printf("Enter d: ");
    scanf("%d",&d);

    printf("Enter e: ");
    scanf("%d",&e);

    if(a>b&&a>c&&a>d&&a>e)
    {
        printf("Highest value: %d\nposition = 1",a);
    }
    else if(b>a&&b>c&&b>d&&b>e)
    {
        printf("Highest value: %d\nposition = 2",b);
    }
    else if(c>b&&c>a&&c>d&&c>e)
    {
        printf("Highest value: %d\nposition = 3",c);
    }
    else if(d>b&&d>c&&d>a&&d>e)
    {
        printf("Highest value: %d\nposition = 4",d);
    }
    else
    {
        printf("Highest value: %d\nposition = 5",e);
    }
}
```

```
}  
    return 0;  
}
```

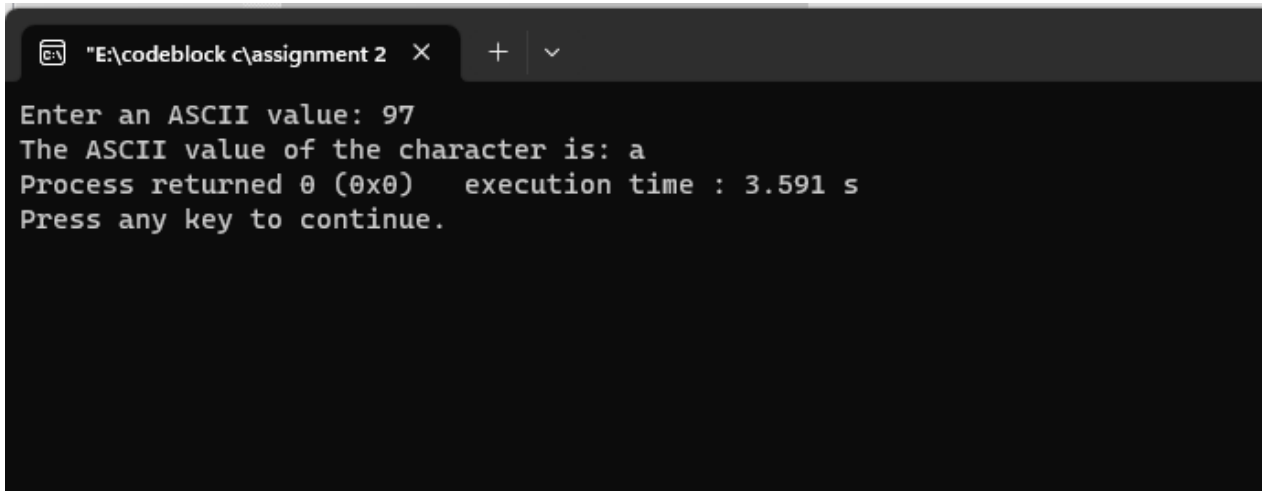


The screenshot shows a terminal window from the Code::Blocks IDE. The title bar indicates the file path is "E:\codeblock c\assignment 2". The terminal output shows the program's execution: it prompts for five integers (a, b, c, d, e), finds the highest value (56) at position 1, and reports an execution time of 4.408 seconds. The program ends with a return code of 0 and a prompt to press any key to continue.

```
"E:\codeblock c\assignment 2" X + v  
Enter a: 56  
Enter b: 43  
Enter c: 5  
Enter d: 3  
Enter e: 55  
Highest value: 56  
position = 1  
Process returned 0 (0x0)   execution time : 4.408 s  
Press any key to continue.
```


33.Integer number to ASCII value

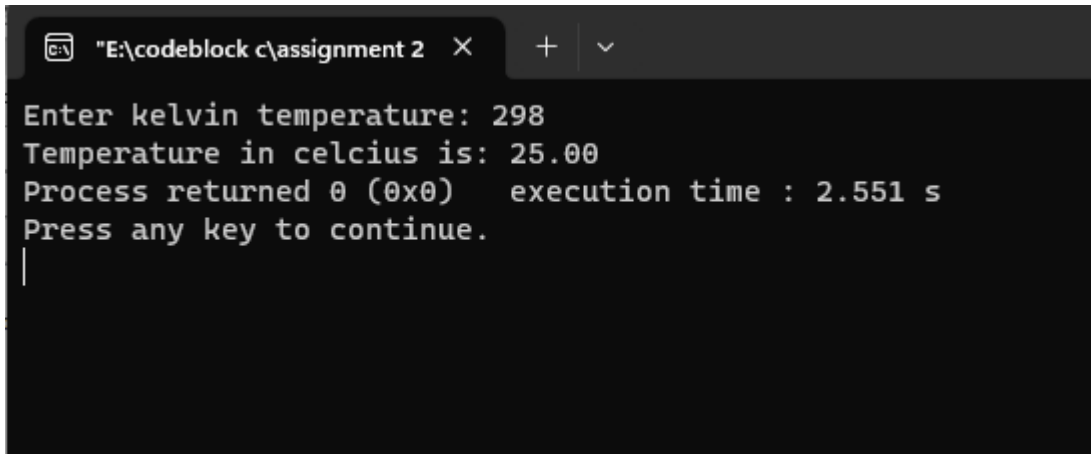
```
#include<stdio.h>
int main()
{ char n;
  printf("Enter an ASCII value: ");
  scanf("%d",&n);
  printf("The ASCII value of the character is: %c",n);
  return 0;
}
```



```
E:\codeblock c\assignment 2 X + v
Enter an ASCII value: 97
The ASCII value of the character is: a
Process returned 0 (0x0)   execution time : 3.591 s
Press any key to continue.
```

34.Kelvin to Celsius

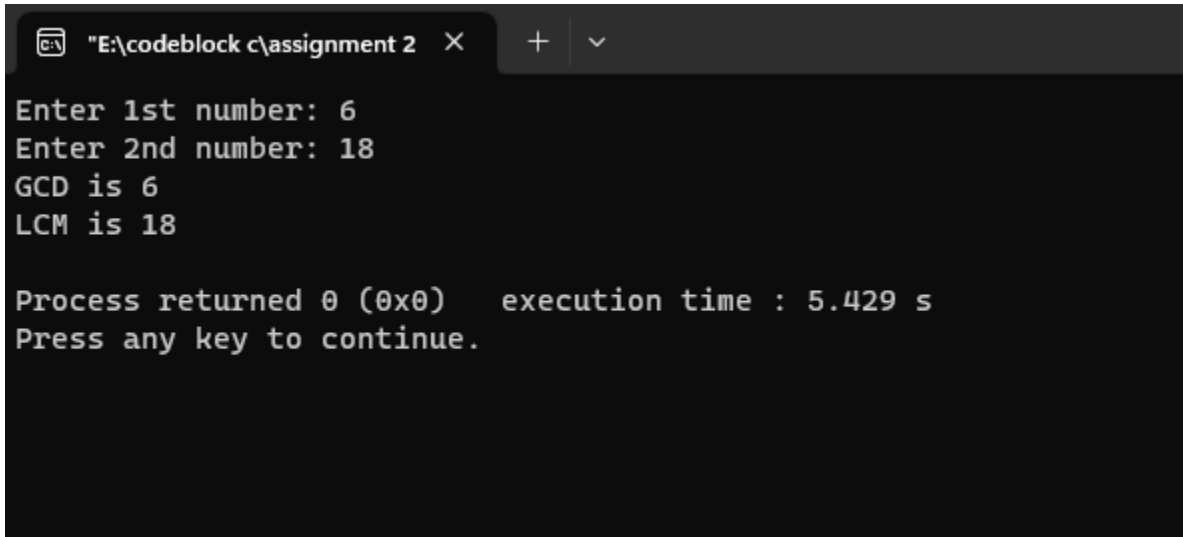
```
#include<stdio.h>
int main()
{
    float c,k;
    printf("Enter kelvin temperature: ");
    scanf("%f",&k);
    c=(k-273));
    printf("Temperature in celcius is: %0.2f",c);
    return 0;
}
```

A screenshot of a code editor window with a dark theme. The title bar shows the file path "E:\codeblock c\assignment 2" and standard window controls. The editor contains the output of a C program: "Enter kelvin temperature: 298", "Temperature in celcius is: 25.00", "Process returned 0 (0x0) execution time : 2.551 s", and "Press any key to continue." followed by a vertical cursor line. The text is in a monospaced font.

```
"E:\codeblock c\assignment 2" X + v
Enter kelvin temperature: 298
Temperature in celcius is: 25.00
Process returned 0 (0x0) execution time : 2.551 s
Press any key to continue.
|
```

35.LCM and GCD

```
#include<stdio.h>
int main()
{
    int n1,n2,rem,lcm,gcd,num1,num2;
    printf("Enter 1st number: ");
    scanf("%d",&num1);
    printf("Enter 2nd number: ");
    scanf("%d",&num2);
    n1=num1;
    n2=num2;
    while(n2!=0)
    {
        rem=n1%n2;
        n1=n2;
        n2=rem;
    }
    gcd=n1;
    lcm=((num1*num2)/gcd);
    printf("GCD is %d\n",gcd);
    printf("LCM is %d\n",lcm);
}
```



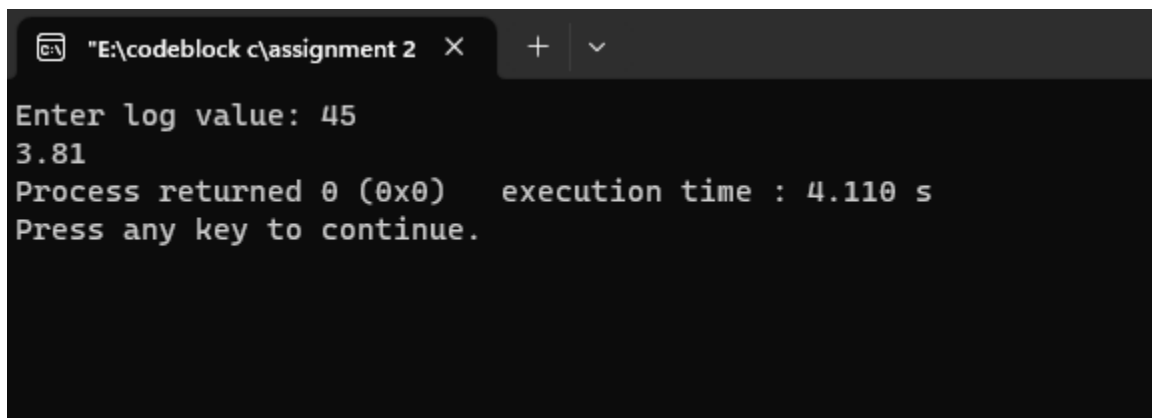
The screenshot shows a code editor window with the title "E:\codeblock c\assignment 2". The program's output is displayed in a terminal window below the code. The user enters 6 for the first number and 18 for the second number. The program calculates the GCD as 6 and the LCM as 18. The execution time is 5.429 seconds, and the process returned 0 (0x0). The terminal prompt asks the user to press any key to continue.

```
"E:\codeblock c\assignment 2" X + v
Enter 1st number: 6
Enter 2nd number: 18
GCD is 6
LCM is 18

Process returned 0 (0x0)   execution time : 5.429 s
Press any key to continue.
```


36. Use of Log function

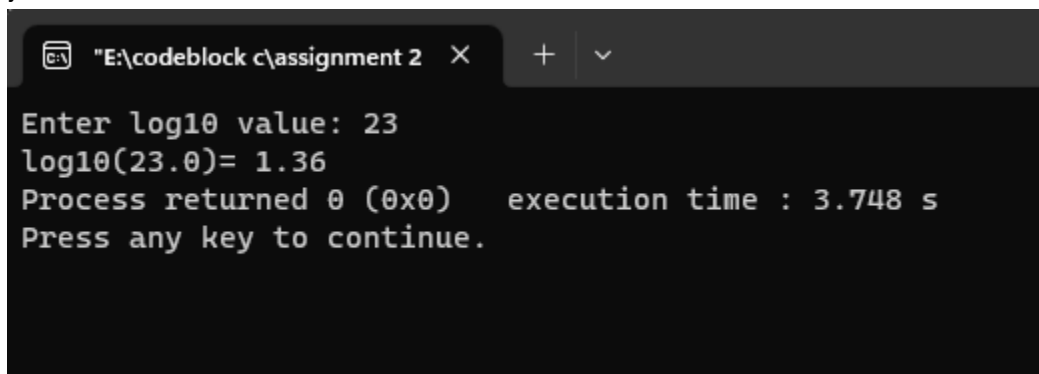
```
#include<stdio.h>
int main()
{
    double result,x;
    printf("Enter log value: ");
    scanf("%lf",&x);
    result=log(x);
    printf("%0.2lf",result);
    return 0;
}
```



The screenshot shows a code editor window titled "E:\codeblock c\assignment 2". The code is the same as shown in the previous block. The output of the program is displayed in the console: "Enter log value: 45", "3.81", "Process returned 0 (0x0) execution time : 4.110 s", and "Press any key to continue.".

37.Using of Log10 function

```
#include<stdio.h>
int main()
{
    double result,x;
    printf("Enter log10 value: ");
    scanf("%lf",&x);
    result=log10(x);
    printf("log10(%0.1f)= %0.2lf",x,result);
    return 0;
}
```

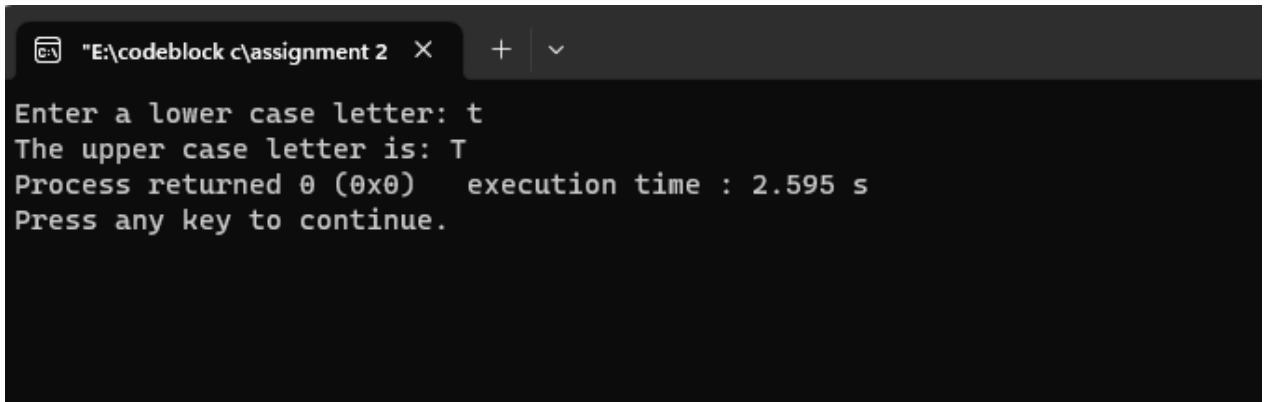


The screenshot shows a code editor window with the title bar "E:\codeblock c\assignment 2". The code is the same as shown in the previous block. The output of the program is displayed in the console area, showing the input value 23 and the calculated log10 value 1.36. The console also shows the process return time and a prompt to press any key to continue.

```
Enter log10 value: 23
log10(23.0)= 1.36
Process returned 0 (0x0)   execution time : 3.748 s
Press any key to continue.
```

38. Lower case letter to upper case letter without using function

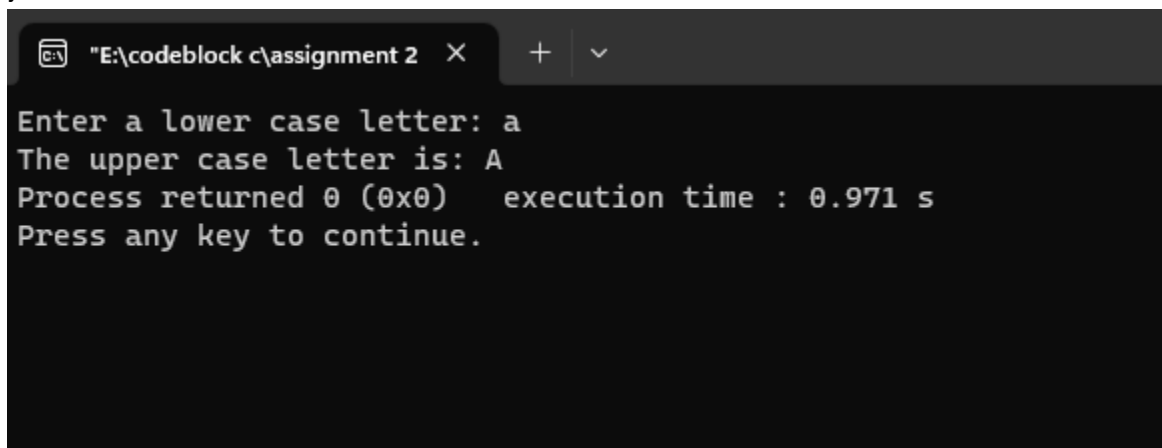
```
#include<stdio.h>
int main()
{
    char lower;
    printf("Enter a lower case letter: ");
    scanf("%c",&lower);
    printf("The upper case letter is: %c",lower-32);
}
```



```
"E:\codeblock c\assignment 2" X + v
Enter a lower case letter: t
The upper case letter is: T
Process returned 0 (0x0)   execution time : 2.595 s
Press any key to continue.
```

39. Lower case letter to upper case letter using library function

```
#include<stdio.h>
int main()
{
    char lower,upper;
    printf("Enter a lower case letter: ");
    scanf("%c",&lower);
    upper = toupper(lower);
    printf("The upper case letter is: %c",upper);
}
```

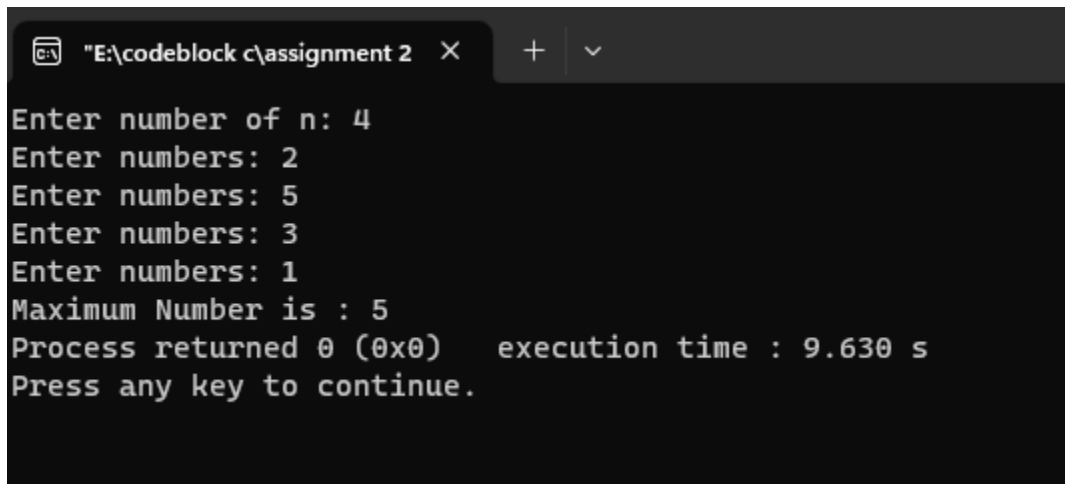


```
E:\codeblock c\assignment 2 X + v
Enter a lower case letter: a
The upper case letter is: A
Process returned 0 (0x0)   execution time : 0.971 s
Press any key to continue.
```


40.Maximum number from array

```
#include<stdio.h>
int main()
{

    int n,a[100],i;
    printf("Enter number of n: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter numbers: ");
        scanf("%d",&a[i]);
    }
    int max= a[0];
    for(i=1;i<n;i++)
    {
        if(max<a[i])
        {
            max=a[i];
        }
    }
    printf("Maximum Number is : %d",max);
}
```



The screenshot shows a Code::Blocks IDE window with the title bar "E:\codeblock c\assignment 2". The console output displays the program's execution: it prompts for the number of elements (4), then for each element (2, 5, 3, 1), and finally outputs the maximum number (5). It also shows the process returned 0 and the execution time was 9.630 seconds.

```
"E:\codeblock c\assignment 2" X + v
Enter number of n: 4
Enter numbers: 2
Enter numbers: 5
Enter numbers: 3
Enter numbers: 1
Maximum Number is : 5
Process returned 0 (0x0)    execution time : 9.630 s
Press any key to continue.
```

41.Menu based temperature converter

```
#include<stdio.h>
int main()

    int choice;
    float c,f;
    printf("Temperature converter manu:\n1.Fahrenheit To Celsius.\n2.Celsius To
Fahrenheit.\nEnter Your choice: ");
    scanf("%d",&choice);
    switch(choice)
    {
    case 1:
        {

            printf("Enter Faregnheight degree: ");
            scanf("%f",&f);
            c=(((f-32)*5)/9);
            printf("Celcius Degree is: %0.2f",c);

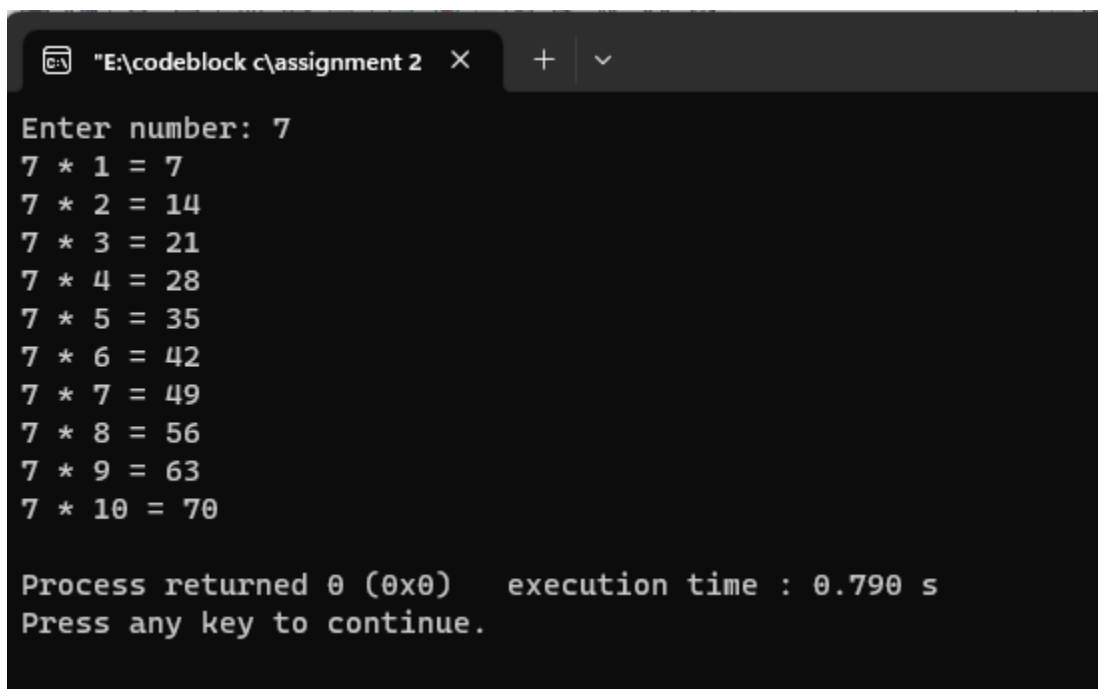
        }
        break;
    case 2:
        {

            printf("Enter Celcius degree: ");
            scanf("%f",&c);
            f=(((9*c)/5)+32);
            printf("Fareignheight Degree is: %0.2f",f);

        }
        break;
    }
}
```

42.Multiplication table

```
#include<stdio.h>
int main()
{
    int n,i;
    printf("Enter number: ");
    scanf("%d",&n);
    for(i=1;i<=10;i++)
    {
        printf("%d * %d = %d",n,i,n*i);
        printf("\n");
    }
}
```



The screenshot shows a Code::Blocks IDE window with the title bar "E:\codeblock c\assignment 2". The console output displays the program's execution. It prompts the user to "Enter number: 7". Following this, it prints a multiplication table for the number 7, with rows for multipliers from 1 to 10. The output is as follows:

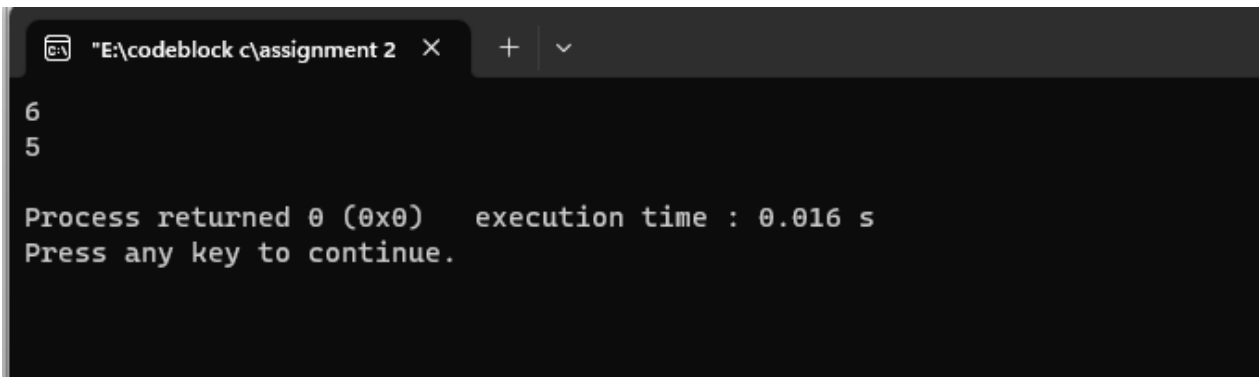
```
Enter number: 7
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
7 * 10 = 70

Process returned 0 (0x0)   execution time : 0.790 s
Press any key to continue.
```

43.Copy one variable to another variable

```
#include<stdio.h>
int main()
{
    int n,x=5,num,y;
    n=6;
    num=n;
    n=x;
    printf("%d\n",num);
    printf("%d\n",n);

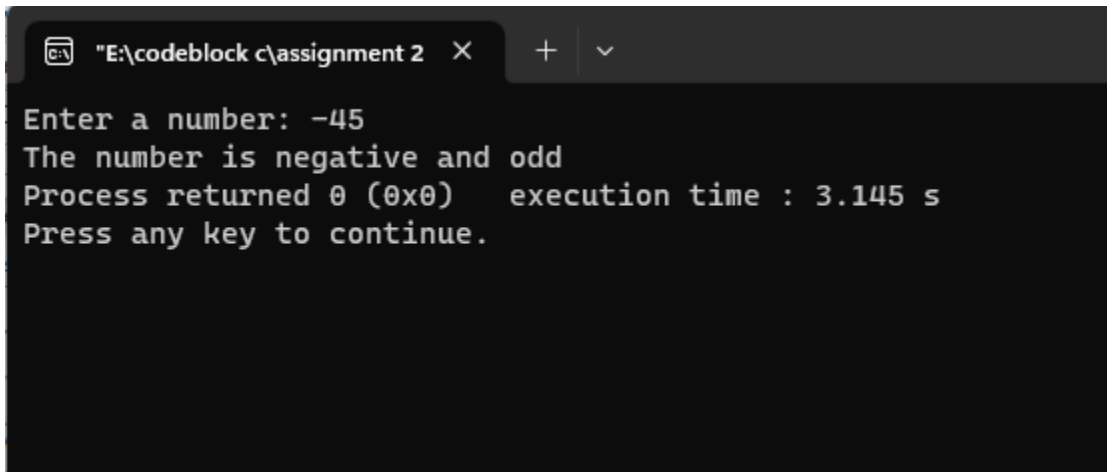
}
```



```
E:\codeblock c\assignment 2  X + v
6
5
Process returned 0 (0x0)  execution time : 0.016 s
Press any key to continue.
```

44.Value is positive or negative or even or odd

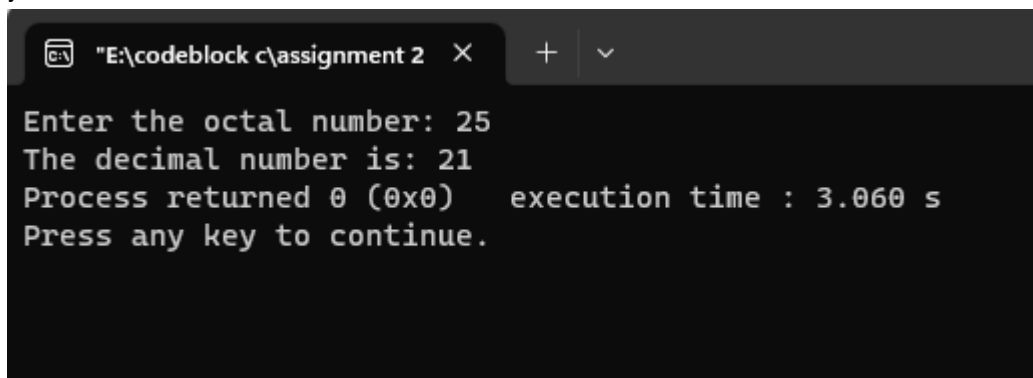
```
#include<stdio.h>
int main()
{
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    if(n>=0 && n%2==0){
        printf("The number is positive and even");
    }
    else if(n<0 && n%2==0){
        printf("The number is negative and even");
    }
    else if(n>=0 && n%2!=0){
        printf("The number is positive and odd");
    }
    else{
        printf("The number is negative and odd");
    }
}
```



```
"E:\codeblock c\assignment 2" X + v
Enter a number: -45
The number is negative and odd
Process returned 0 (0x0)   execution time : 3.145 s
Press any key to continue.
```

45. Octal to decimal

```
#include<stdio.h>
int main()
{
    int n;
    printf("Enter the octal number: ");
    scanf("%o",&n);
    printf("The decimal number is: %d",n);
    return 0;
}
```

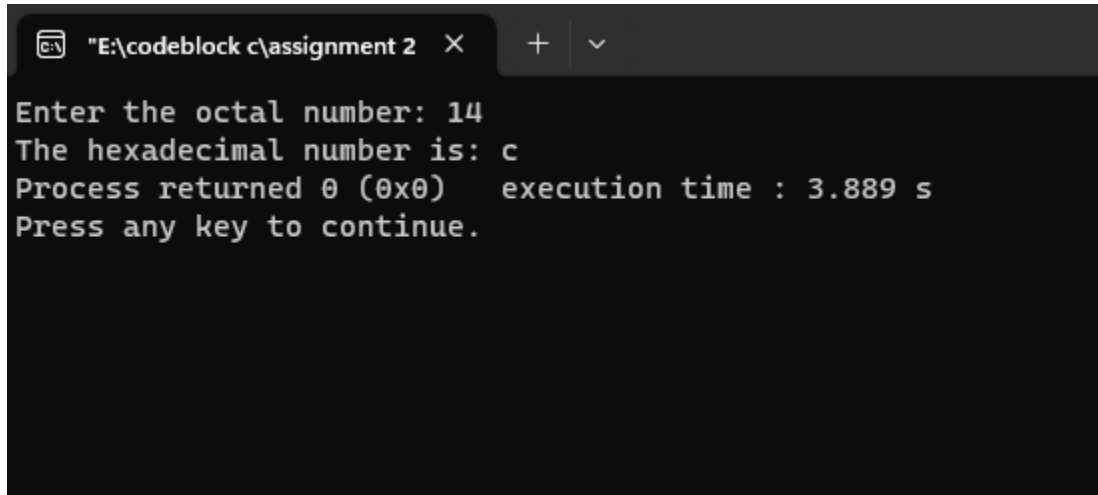


The screenshot shows a Code::Blocks IDE window with the title bar "E:\codeblock c\assignment 2". The console output displays the program's execution: it prompts for an octal number, receives the input "25", and outputs "The decimal number is: 21". Below this, it shows "Process returned 0 (0x0)" and "execution time : 3.060 s". The prompt "Press any key to continue." is visible at the bottom of the console window.

```
Enter the octal number: 25
The decimal number is: 21
Process returned 0 (0x0)   execution time : 3.060 s
Press any key to continue.
```

46.Octal to hexadecimal

```
#include<stdio.h>
int main()
{
    int n;
    printf("Enter the octal number: ");
    scanf("%o",&n);
    printf("The hexadecimal number is: %x",n);
    return 0;
}
```

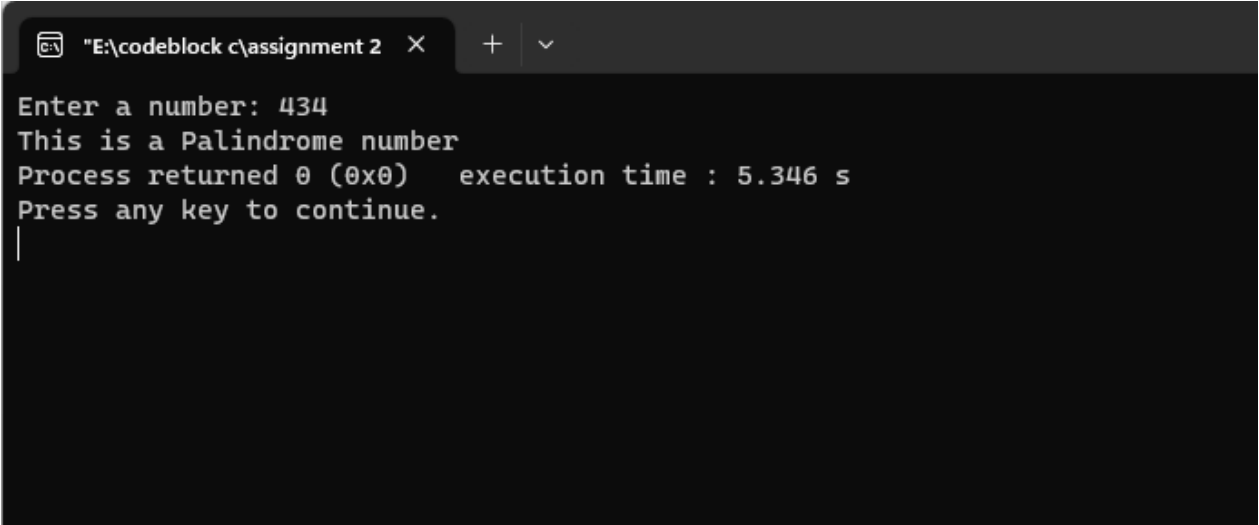


The screenshot shows a terminal window titled "E:\codeblock c\assignment 2". The program has been executed with the input '14'. The output shows 'The hexadecimal number is: c'. Below the output, it says 'Process returned 0 (0x0) execution time : 3.889 s' and 'Press any key to continue.'.

```
E:\codeblock c\assignment 2
Enter the octal number: 14
The hexadecimal number is: c
Process returned 0 (0x0) execution time : 3.889 s
Press any key to continue.
```

47. Palindrome number or not

```
#include<stdio.h>
int main()
{
    int n,num,r,sum=0;
    printf("Enter a number: ");
    scanf("%d",&num);
    n=num;
    while(n!=0)
    {
        r=n%10;
        sum=sum*10+r;
        n=n/10;
    }
    if(sum==num)
        printf("This is a Palindrome number");
    else
        printf("Not a palimdrome number");
    return 0;
}
```



The screenshot shows a terminal window with a dark background. The title bar at the top reads "E:\codeblock c\assignment 2" with a close button on the right. The terminal content shows the program's execution: it prompts "Enter a number:" and the user enters "434". The program then outputs "This is a Palindrome number". Below this, it shows "Process returned 0 (0x0) execution time : 5.346 s" and "Press any key to continue." followed by a vertical cursor line.

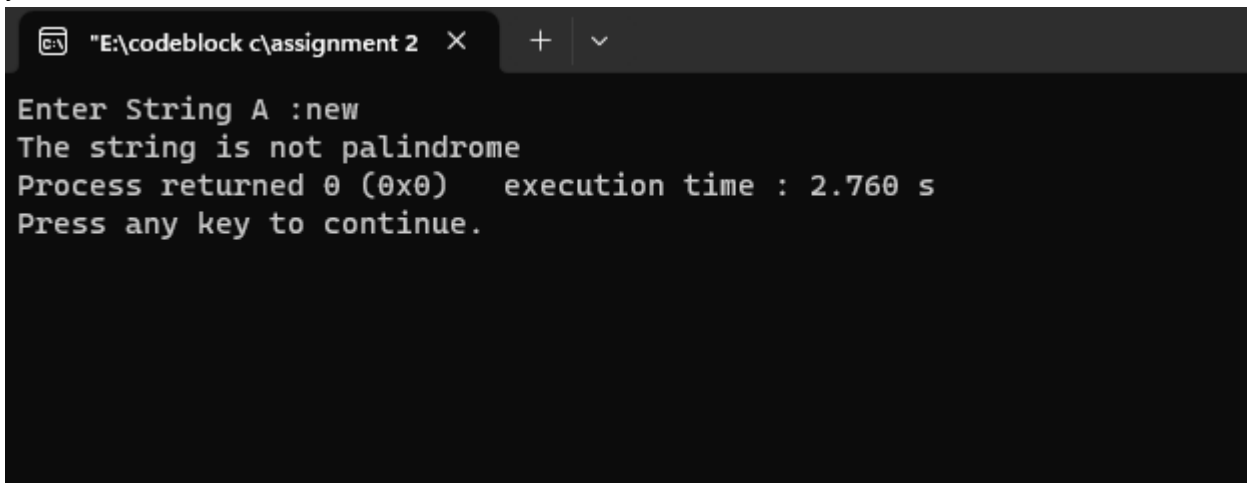
```
"E:\codeblock c\assignment 2" X + v
Enter a number: 434
This is a Palindrome number
Process returned 0 (0x0) execution time : 5.346 s
Press any key to continue.
|
```


48. Palindrome number using String

```
#include<stdio.h>
int main()
{
    char b[100],a[100];
    printf("Enter String A :");
    gets(a);

    strcpy(b,a);

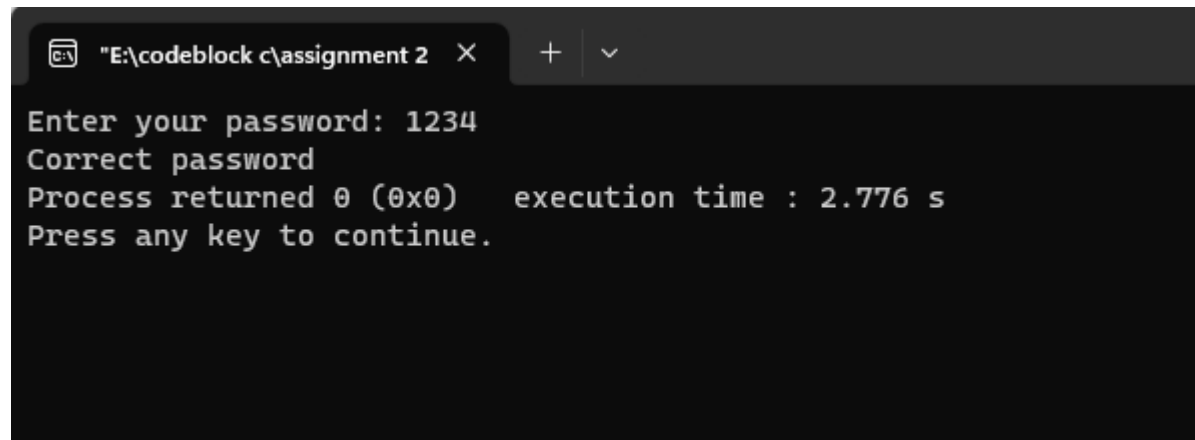
    if(b==(strrev(a)))
    {
        printf("String is palindrome");
    }
    else
        printf("The string is not palindrome");
}
```



```
E:\codeblock c\assignment 2  X  +  v
Enter String A :new
The string is not palindrome
Process returned 0 (0x0)  execution time : 2.760 s
Press any key to continue.
```

49.Password 1234

```
#include<stdio.h>
int main()
{
    int pass;
    printf("Enter your password: ");
    scanf("%d",&pass);
    if(pass==1234)
    {
        printf("Correct password");
    }
    else
    {
        printf("Incorrect Password");
    }
    return 0;
}
```



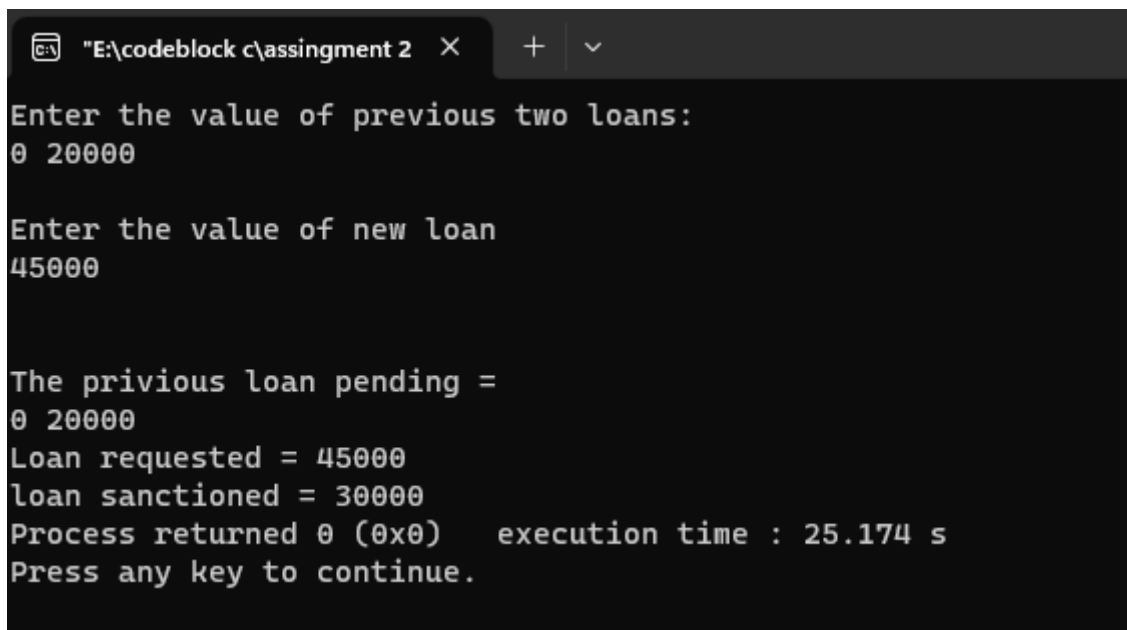
The screenshot shows a code editor window with a single tab titled "E:\codeblock c\assignment 2". The code from the previous block is visible in the editor. Below the code, the output of the program is displayed in a dark-themed console window. The output shows the prompt "Enter your password: 1234", followed by "Correct password", then "Process returned 0 (0x0) execution time : 2.776 s", and finally "Press any key to continue.".

```
"E:\codeblock c\assignment 2" X + v
Enter your password: 1234
Correct password
Process returned 0 (0x0) execution time : 2.776 s
Press any key to continue.
```

50.Loan from a company

```
#define MAXLOAN 50000
main()
{
    long int loan1,loan2,loan3,sancloan,sum23;
    printf("Enter the value of previous two loans: \n");
    scanf("%ld %ld",&loan1,&loan2);
    printf("\nEnter the value of new loan\n");
    scanf("%ld",&loan3);
    sum23=loan2+loan3;
    sancloan = (loan1>0)?0:((sum23>MAXLOAN)?MAXLOAN-loan2:loan3);
    printf("\n\n");
    printf("The privious loan pending = \n%ld %ld\n",loan1,loan2);
    printf("Loan requested = %ld\n",loan3);
    printf("loan sanctioned = %ld",sancloan);

}
```



```
"E:\codeblock c\assingment 2" X + v
Enter the value of previous two loans:
0 20000

Enter the value of new loan
45000

The privious loan pending =
0 20000
Loan requested = 45000
loan sanctioned = 30000
Process returned 0 (0x0)   execution time : 25.174 s
Press any key to continue.
```