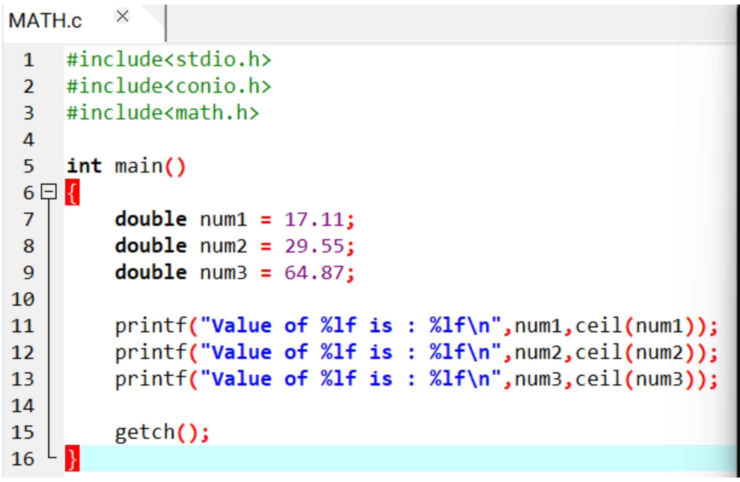
57. include<math.h> In C Programming

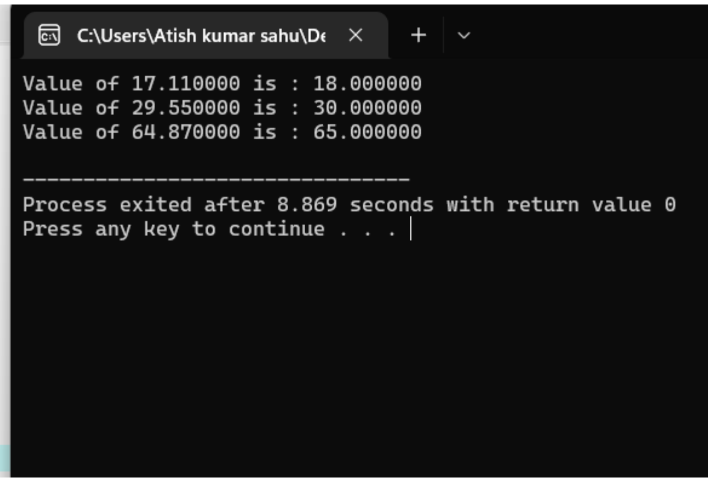
The math.h header file is used for using different types of mathematical function in programming. There are various functions are there like trignometric function floating point functions and many more are available in this header file.

1. ceil(number):

rounds up the given number. It returns the integer value which is greater than or equal to given number. The C library function double ceil(double x) returns the smallest integer value greater than or equal to x.



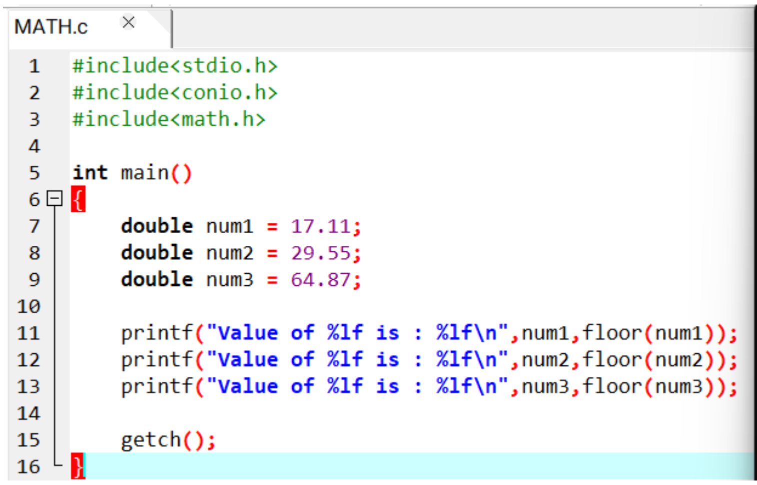
```c
#include<stdio.h>
#include<conio.h>
#include<math.h>

int main()
{
    double num1 = 17.11;
    double num2 = 29.55;
    double num3 = 64.87;

    printf("Value of %lf is : %lf\n",num1,ceil(num1));
    printf("Value of %lf is : %lf\n",num2,ceil(num2));
    printf("Value of %lf is : %lf\n",num3,ceil(num3));

    getch();
}
```



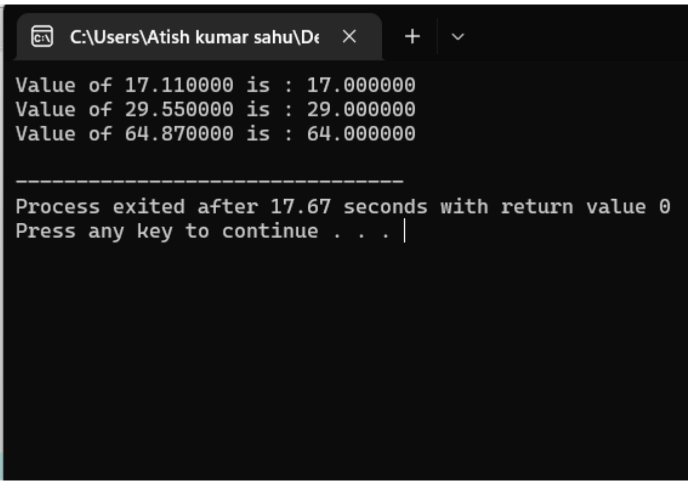
```
Value of 17.110000 is : 18.000000
Value of 29.550000 is : 30.000000
Value of 64.870000 is : 65.000000

--------------------------------
Process exited after 8.869 seconds with return value 0
Press any key to continue . . .
```

2. floor(number):

rounds down the given number. It returns the integer value which is less than or equal to given number. The C library function double floor(double x) returns the largest integer value less than or equal to x.



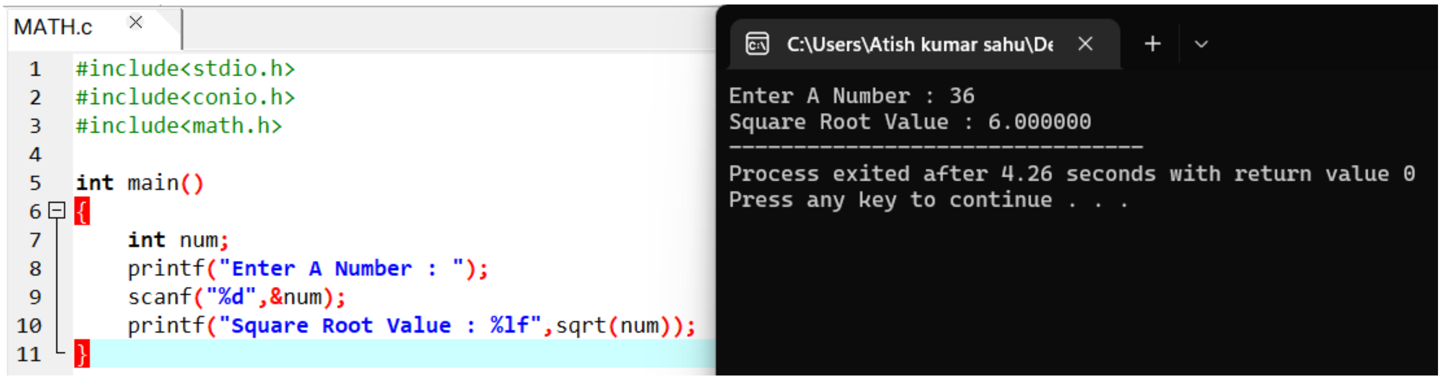
```c
#include<stdio.h>
#include<conio.h>
#include<math.h>

int main()
{
    double num1 = 17.11;
    double num2 = 29.55;
    double num3 = 64.87;

    printf("Value of %lf is : %lf\n",num1,floor(num1));
    printf("Value of %lf is : %lf\n",num2,floor(num2));
    printf("Value of %lf is : %lf\n",num3,floor(num3));

    getch();
}
```



```
Value of 17.110000 is : 17.000000
Value of 29.550000 is : 29.000000
Value of 64.870000 is : 64.000000

--------------------------------
Process exited after 17.67 seconds with return value 0
Press any key to continue . . .
```

3. sqrt(number):

returns the square root of given number. The C library function double sqrt(double x) returns the square root of x.

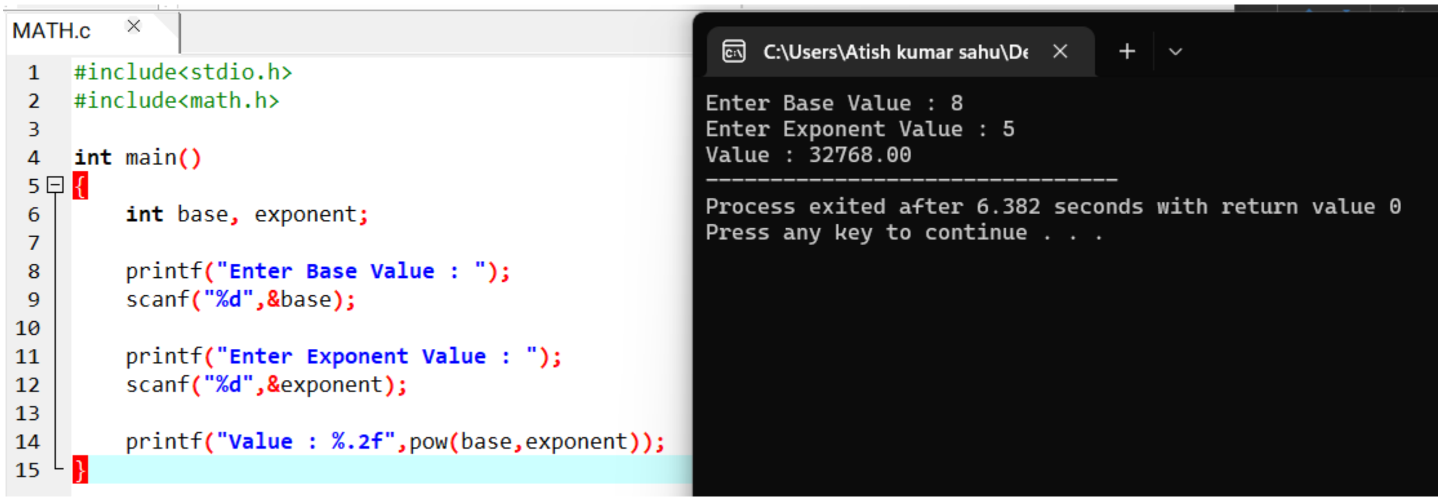


```c
#include<stdio.h>
#include<conio.h>
#include<math.h>

int main()
{
	int num;
	printf("Enter A Number : ");
	scanf("%d",&num);
	printf("Square Root Value : %lf",sqrt(num));
}
```

```
Enter A Number : 36
Square Root Value : 6.000000
--------------------------------
Process exited after 4.26 seconds with return value 0
Press any key to continue . . .
```

4. pow(base, exponent):

returns the power of given number. The C library function double pow(double x, double y) returns x raised to the power of y i.e. xy.

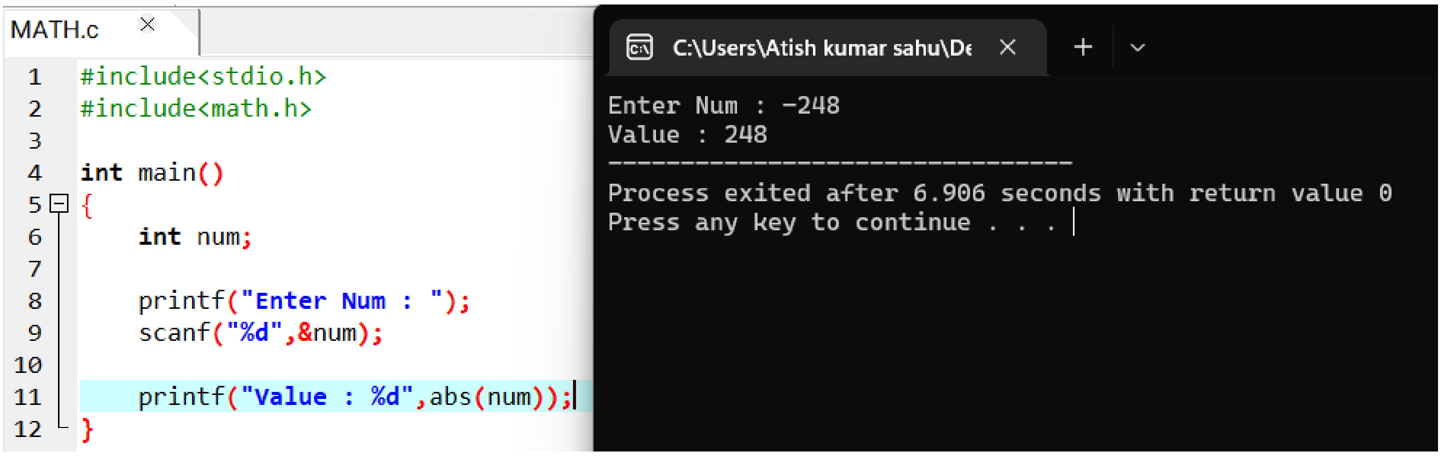


```c
#include<stdio.h>
#include<math.h>

int main()
{
	int base, exponent;
	
	printf("Enter Base Value : ");
	scanf("%d",&base);
	
	printf("Enter Exponent Value : ");
	scanf("%d",&exponent);
	
	printf("Value : %.2f",pow(base,exponent));
}
```

```
Enter Base Value : 8
Enter Exponent Value : 5
Value : 32768.00
--------------------------------
Process exited after 6.382 seconds with return value 0
Press any key to continue . . .
```

5. abs(number):

returns the absolute value of given number.

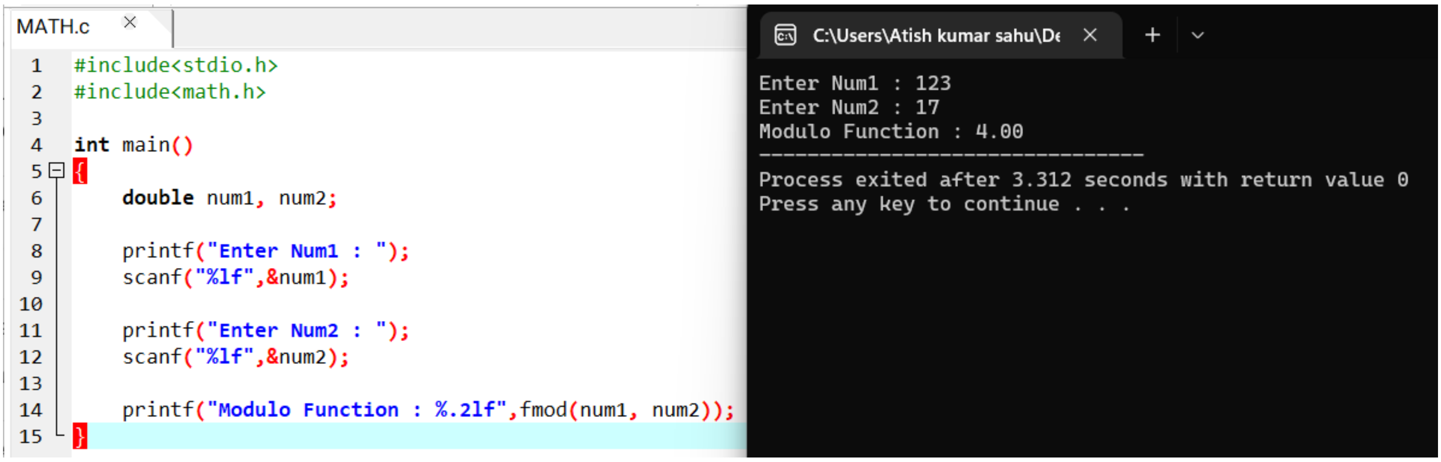


```c
#include<stdio.h>
#include<math.h>

int main()
{
	int num;
	
	printf("Enter Num : ");
	scanf("%d",&num);
	
	printf("Value : %d",abs(num));
}
```

```
Enter Num : -248
Value : 248
--------------------------------
Process exited after 6.906 seconds with return value 0
Press any key to continue . . .
```

6. fmod(double num1, double num2):

The C library function double fmod(double x, double y) returns the remainder of x divided by y. x is the floating point value with the division numerator i.e. x. y is the floating point value with the division denominator i.e. y.

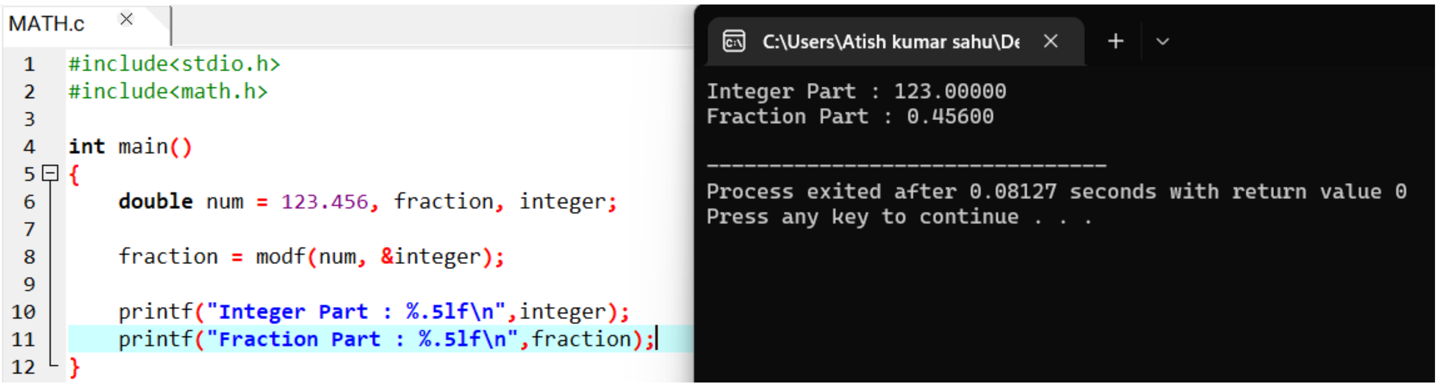


```c
#include<stdio.h>
#include<math.h>

int main()
{
    double num1, num2;

    printf("Enter Num1 : ");
    scanf("%lf",&num1);

    printf("Enter Num2 : ");
    scanf("%lf",&num2);

    printf("Modulo Function : %.2lf",fmod(num1, num2));
}
```

```
Enter Num1 : 123
Enter Num2 : 17
Modulo Function : 4.00
--------------------------------
Process exited after 3.312 seconds with return value 0
Press any key to continue . . .
```

7. modf(double num, double *integer):

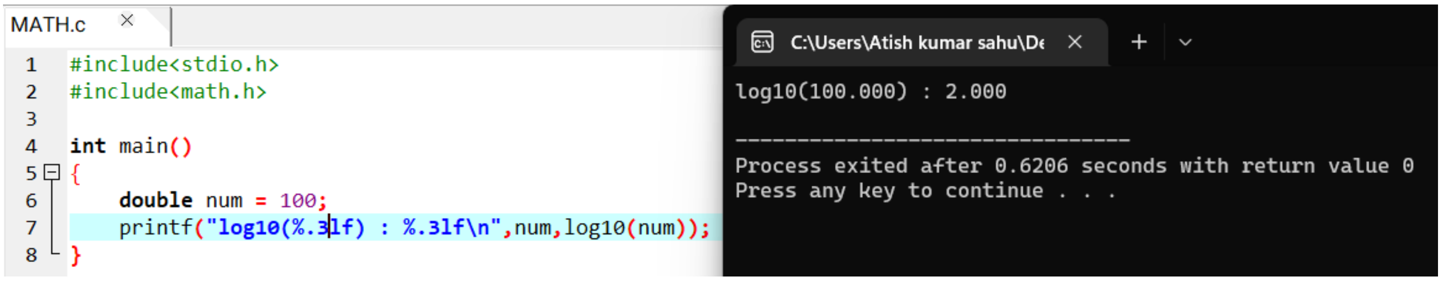
The C library function double modf(double x, double *integer) returns the fraction component (part after the decimal), and sets integer to the integer component. X is the floating point value. “double *integer” is the pointer to an object where the integral part is to be stored.



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

int main()
{
    double num = 123.456, fraction, integer;

    fraction = modf(num, &integer);

    printf("Integer Part : %.5lf\n",integer);
    printf("Fraction Part : %.5lf\n",fraction);
}
```

```
Integer Part : 123.00000
Fraction Part : 0.45600

--------------------------------
Process exited after 0.08127 seconds with return value 0
Press any key to continue . . .
```

8. log10(double num): The C library function double log10(double x) returns the common logarithm (base-10 logarithm) of x.

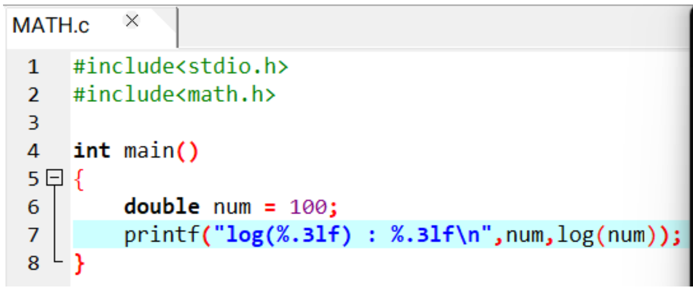


```c
#include<stdio.h>
#include<math.h>

int main()
{
    double num = 100;
    printf("log10(%.3lf) : %.3lf\n",num,log10(num));
}
```

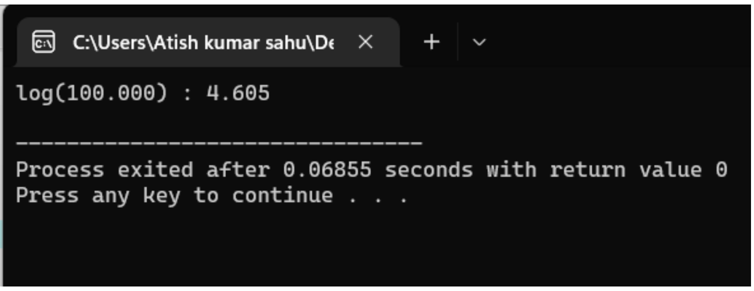
```
log10(100.000) : 2.000

--------------------------------
Process exited after 0.6206 seconds with return value 0
Press any key to continue . . .
```

9. log(double num):

The C library function double log(double x) returns the natural logarithm (base-e logarithm) of x. x is the floating point value.



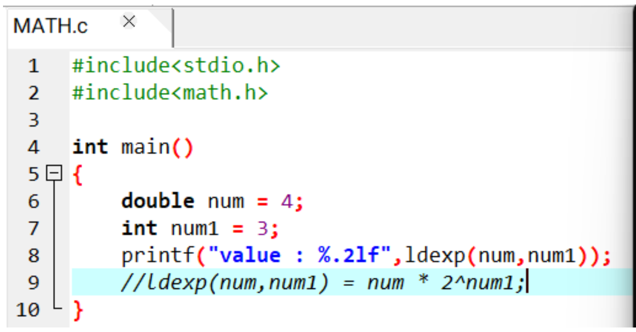
```c
#include<stdio.h>
#include<math.h>

int main()
{
    double num = 100;
    printf("log(%.3lf) : %.3lf\n",num,log(num));
}
```



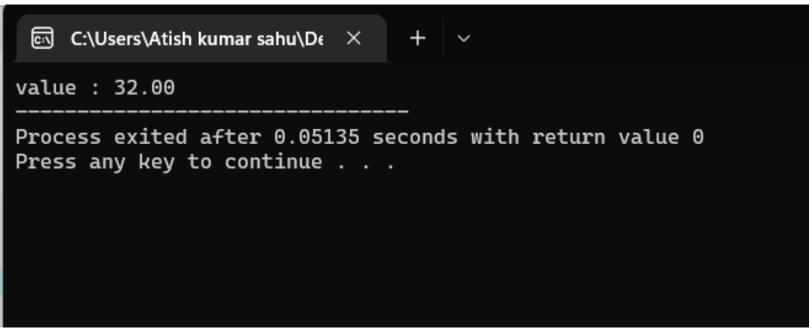
```
log(100.000) : 4.605

--------------------------------
Process exited after 0.06855 seconds with return value 0
Press any key to continue . . .
```

10. ldexp(double num, int exponent):

The C library function double ldexp(double x, int exponent) returns x multiplied by 2 raised to the power of exponent. X is the floating point value representing the significand. Exponent is the value of the exponent.



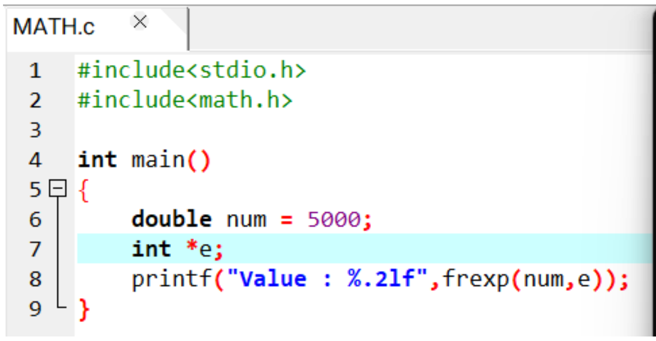
```c
#include<stdio.h>
#include<math.h>

int main()
{
    double num = 4;
    int num1 = 3;
    printf("value : %.2lf",ldexp(num,num1));
    //ldexp(num,num1) = num * 2^num1;
}
```



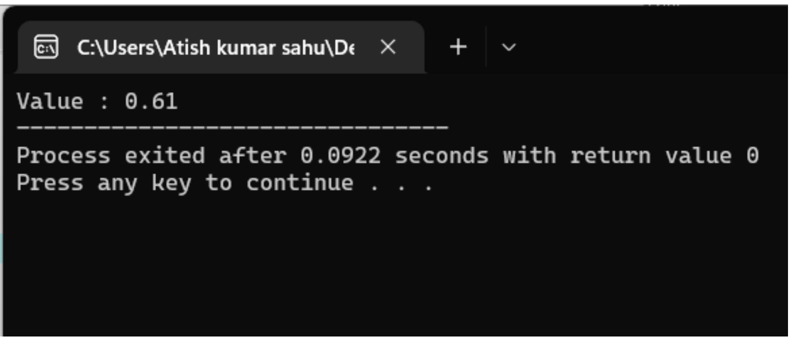
```
value : 32.00
--------------------------------
Process exited after 0.05135 seconds with return value 0
Press any key to continue . . .
```

11. frexp(double num, int *exponent):

The C library function double frexp(double x, int *exponent) return value is the mantissa, and the integer pointed to by exponent is the exponent. The resultant value is x = mantissa * 2 ^ exponent.



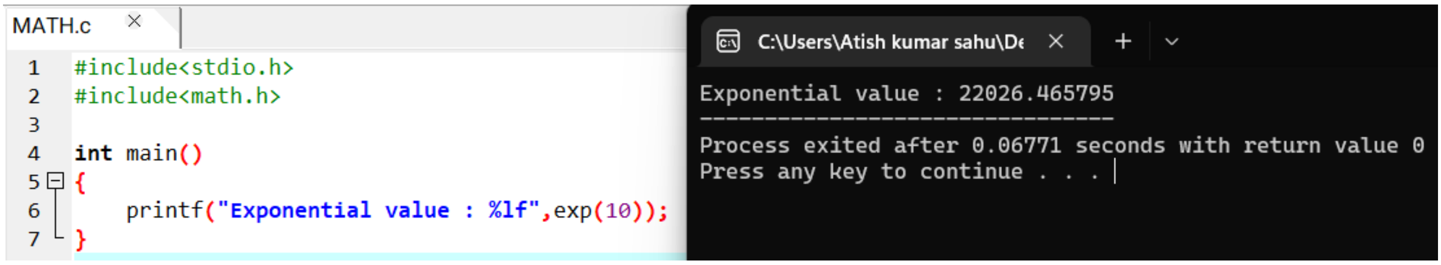
```c
#include<stdio.h>
#include<math.h>

int main()
{
    double num = 5000;
    int *e;
    printf("Value : %.2lf",frexp(num,e));
}
```



```
Value : 0.61
--------------------------------
Process exited after 0.0922 seconds with return value 0
Press any key to continue . . .
```

12. exp(double num):

The C library function double exp(double x) returns the value of e raised to the xth power.

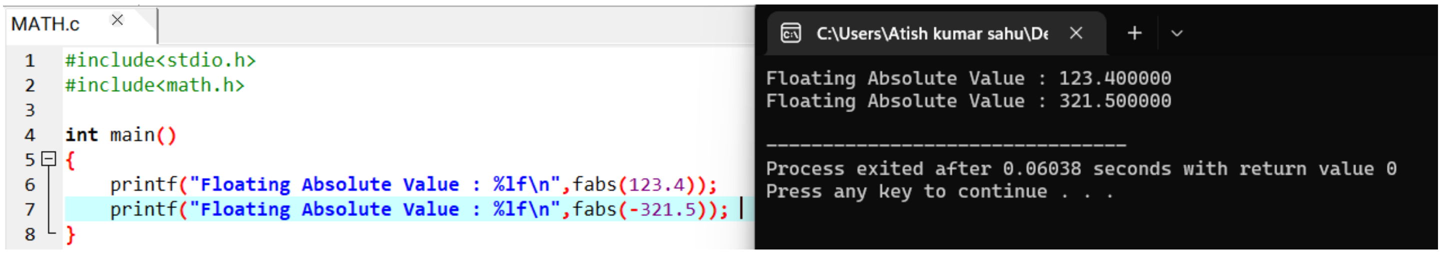


```c
#include<stdio.h>
#include<math.h>

int main()
{
    printf("Exponential value : %lf",exp(10));
}
```

```
Exponential value : 22026.465795
--------------------------------
Process exited after 0.06771 seconds with return value 0
Press any key to continue . . .
```

13. fabs(double num):

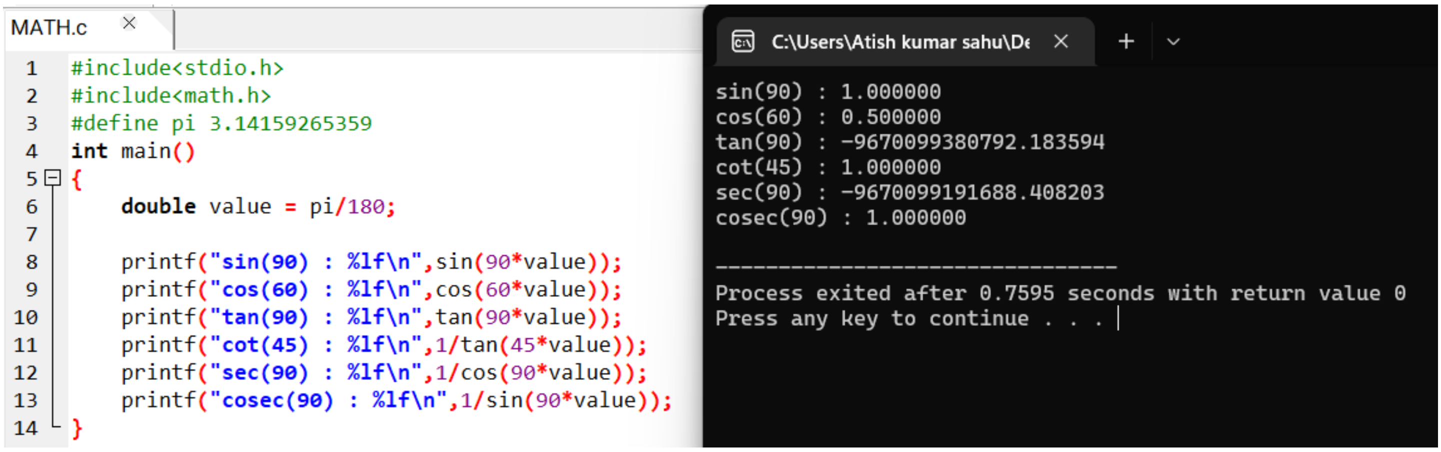
It returns the absolute value of x in floating value.



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

int main()
{
    printf("Floating Absolute Value : %lf\n",fabs(123.4));
    printf("Floating Absolute Value : %lf\n",fabs(-321.5));
}
```

```
Floating Absolute Value : 123.400000
Floating Absolute Value : 321.500000

--------------------------------
Process exited after 0.06038 seconds with return value 0
Press any key to continue . . .
```

14. Trigonometric Function:



```c
#include<stdio.h>
#include<math.h>
#define pi 3.14159265359
int main()
{
    double value = pi/180;

    printf("sin(90) : %lf\n",sin(90*value));
    printf("cos(60) : %lf\n",cos(60*value));
    printf("tan(90) : %lf\n",tan(90*value));
    printf("cot(45) : %lf\n",1/tan(45*value));
    printf("sec(90) : %lf\n",1/cos(90*value));
    printf("cosec(90) : %lf\n",1/sin(90*value));
}
```

```
sin(90) : 1.000000
cos(60) : 0.500000
tan(90) : -9670099380792.183594
cot(45) : 1.000000
sec(90) : -9670099191688.408203
cosec(90) : 1.000000

--------------------------------
Process exited after 0.7595 seconds with return value 0
Press any key to continue . . .
```