

**SONA COLLEGE OF TECHNOLOGY
(AUTONOMOUS)**
SONA NAGAR, SALEM – 636005.



[An Autonomous Institution]

Name : **cccccccccccccc**
Branch : **CSE**
Semester : **2nd semester**
Register No : **cccccccccccc**
Name of the Laboratory : **C Programming Laboratory**

Certified that this is the bona-fide Record of practical work done by the above student in C Programming Laboratory during the year 2020 - 2021 .

Lab-In-Charge

Head of the Department

Submitted for University Practical Examination held on

Internal Examiner

External Examiner

CONTENTS

S.No	Date	Name of the Experiment	Page No	Marks Awarded	Signature of the Staff
1	05.04.21	Basics	4		
2	05.04.21	Operators	10		
3	19.04.21	Operators	20		
4	26.04.21	Decision Statements	30		
5	31.04.21	Switch operations	36		
6	03.05.21	Basic Loop operations Do the Following Programs Using for, while, do-while loops	47		
7	10.05.21	Advanced loops	55		
8	17.05.21	1-D arrays	63		
9	24.05.21	2-D arrays	73		
10	07.06.21	Strings	81		
11	07.06.21	Math Functions and I/O Functions	89		
12	14.06.21	Functions	95		
13	14.06.21	Functions and Recursion	105		

CONTENTS

BASICS

EXERCISE 1.1

EX. NO. 1	BASICS
DATE : 05/4/21	
1. write a program to print Sample Strings like "hello world", "welcome to c programming" with different formats using escape sequences.	
AIM : To write the C program to print Sample Strings like "hello world", "welcome to c programming" with different formats using escape sequences.	
ALGORITHM :	
Step 1 : Start	
Step 2 : print "hello word" and "welcome to programming" using escape sequence 'in'	
Step 3 : Stop	

SOURCE CODE :

The screenshot shows a code editor window titled "Start here" with the file "hello world and welcome.c" open. The code is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     printf("Hello world\nwelcome to C programming");
5
6     return 0;
7 }
8
```

OUTPUT :

The screenshot shows a terminal window with the following output:

```
C:\Users\Rolyx\Desktop\C\Exercise 1\hello world and welcome.exe
Hello world
welcome to C programming
Process returned 0 (0x0)   execution time : 1.203 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 1.2

2. write a program to print different data types in 'c' and their ranges.

AIM :

To write the C program to print different data types

ALGORITHM :

Step 1 : Start

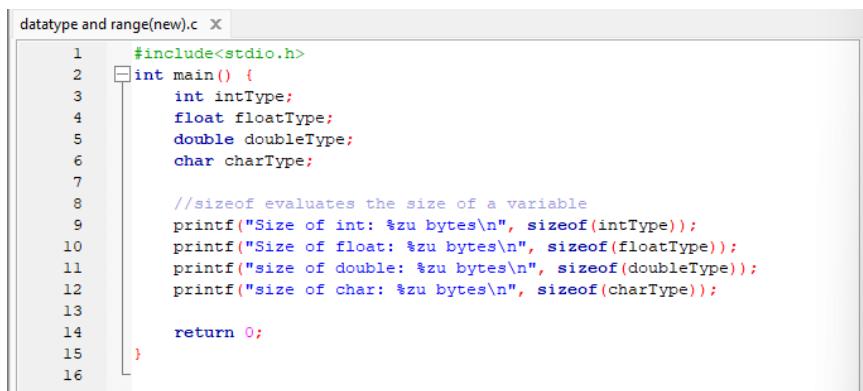
Step 2 : declare the data type

Step 3 : using the size of operator get the range of the data type.

Step 4 : And then print different data types and their ranges

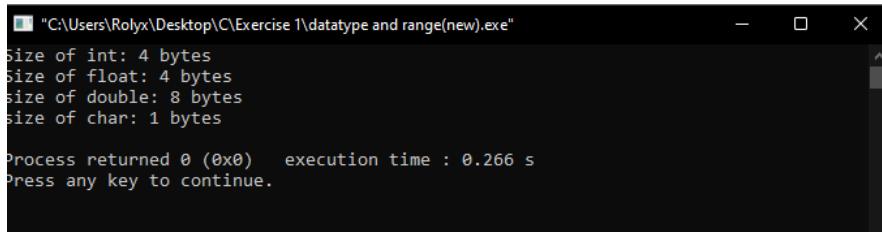
Step 5 : Stop

SOURCE CODE :



```
datatype and range(new).c X
1  #include<stdio.h>
2  int main() {
3      int intType;
4      float floatType;
5      double doubleType;
6      char charType;
7
8      //sizeof evaluates the size of a variable
9      printf("Size of int: %zu bytes\n", sizeof(intType));
10     printf("Size of float: %zu bytes\n", sizeof(floatType));
11     printf("size of double: %zu bytes\n", sizeof(doubleType));
12     printf("size of char: %zu bytes\n", sizeof(charType));
13
14     return 0;
15 }
16
```

OUTPUT:



```
C:\Users\Rolyx\Desktop\C\Exercise 1\datatype and range(new).exe
size of int: 4 bytes
size of float: 4 bytes
size of double: 8 bytes
size of char: 1 bytes

Process returned 0 (0x0)   execution time : 0.266 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 1.3

3. write a program to initialize, assignment & printing variables of different data types.

AIM :

To write a C program to initialize, assign and to print variables of different data types

Step 1 : Start

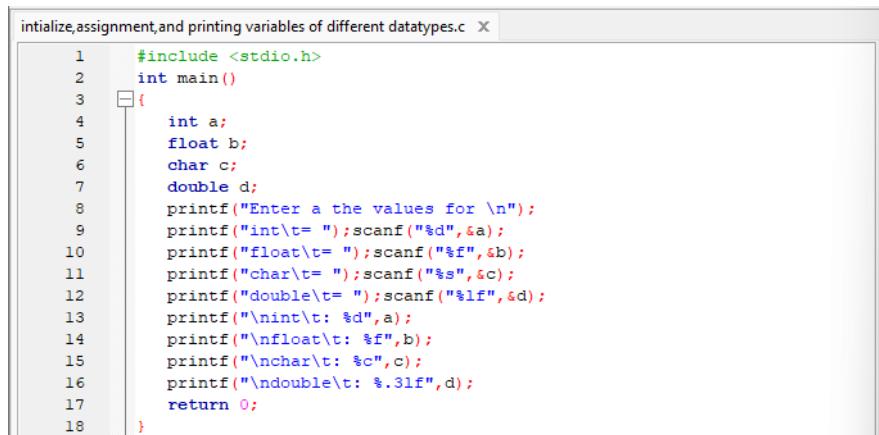
Step 2 : Initialize different data types and declare them with variables

Step 3 : Get the values from the user and assign them.

Step 4 : Then print the values.

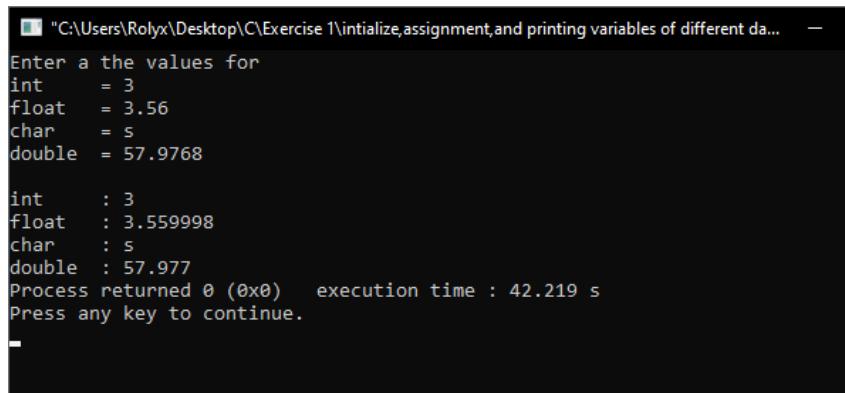
Step 5 : Stop

SOURCE CODE :



```
initialize,assignment,and printing variables of different datatypes.c
1 #include <stdio.h>
2 int main()
3 {
4     int a;
5     float b;
6     char c;
7     double d;
8     printf("Enter a the values for \n");
9     printf("int\t= ");scanf("%d",&a);
10    printf("float\t= ");scanf("%f",&b);
11    printf("char\t= ");scanf("%s",&c);
12    printf("double\t= ");scanf("%lf",&d);
13    printf("\nint\t: %d",a);
14    printf("\nfloat\t: %f",b);
15    printf("\nchar\t: %c",c);
16    printf("\ndouble\t: %.3lf",d);
17    return 0;
18 }
```

OUTPUT:



```
"C:\Users\RoLyX\Desktop\C\Exercise 1\initialize,assignment,and printing variables of different da...
Enter a the values for
int      = 3
float    = 3.56
char     = s
double   = 57.9768

int      : 3
float    : 3.559998
char     : s
double   : 57.977
Process returned 0 (0x0)  execution time : 42.219 s
Press any key to continue.
```

RESULT:

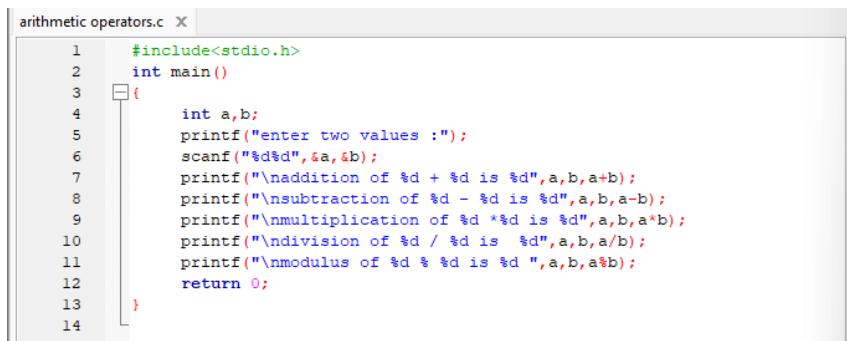
The Desired Output has been successfully Verified and Executed.

OPERATORS

EXERCISE 2.1

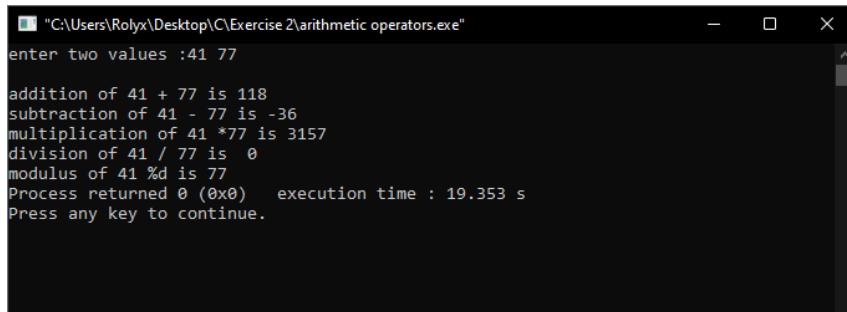
EX. NO. 2	OPERATORS
DATE: 5/4/21	
1. write a program to demonstrate arithmetic operators. (+, -, *, /, %)	
AIM :	
To write a C program to demonstrate arithmetic operators (+, -, *, /, %)	
ALGORITHM :	
STEP 1 : Start	
STEP 2 : Declare int variables a and b	
STEP 3 : get the values of a and b from the user compute the arithmetic operations [a+b, a-b, a*b, a/b, a%b] using the declared operands.	
STEP 4 : print the values obtained by performing the above arithmetic operations.	
STEP 5 : Stop.	

SOURCE CODE :



```
arithmetic operators.c x
1 #include<stdio.h>
2 int main()
3 {
4     int a,b;
5     printf("enter two values :");
6     scanf("%d%d",&a,&b);
7     printf("\naddition of %d + %d is %d",a,b,a+b);
8     printf("\nsubtraction of %d - %d is %d",a,b,a-b);
9     printf("\nmultiplication of %d * %d is %d",a,b,a*b);
10    printf("\ndivision of %d / %d is %d",a,b,a/b);
11    printf("\nmodulus of %d % %d is %d ",a,b,a%b);
12
13 }
14
```

OUTPUT :



```
C:\Users\RoLyx\Desktop\C\Exercise 2\arithmetic operators.exe"
enter two values :41 77
addition of 41 + 77 is 118
subtraction of 41 - 77 is -36
multiplication of 41 *77 is 3157
division of 41 / 77 is 0
modulus of 41 %d is 77
Process returned 0 (0x0)   execution time : 19.353 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 2.2

2. write a program to demonstrate logical operators.
(logical AND, logical OR)

AIM :

To write a C program to demonstrate
logical operators (logical AND, logical OR)

ALGORITHM:

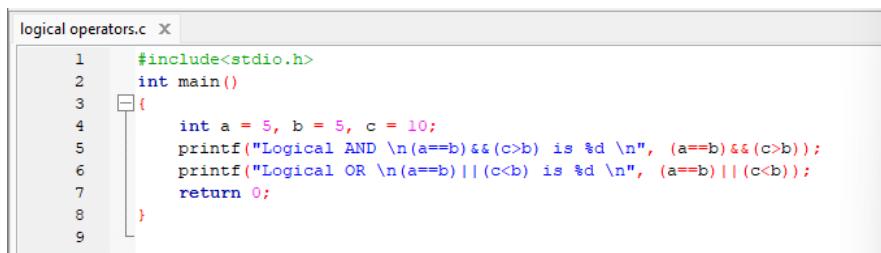
STEP 1 : start

STEP 2 : get values of any 3 int variables
for logical AND using " && " operator
check the condition and display result.

STEP 3 : for logical OR using " || " operator
check the condition and display result

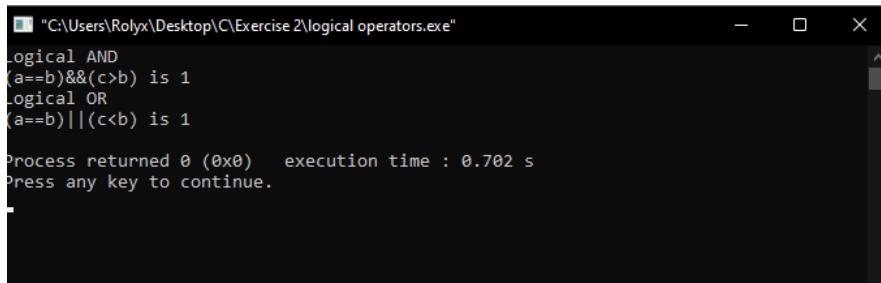
STEP 4 : stop .

SOURCE CODE :



```
logical operators.c X
1  #include<stdio.h>
2  int main()
3  {
4      int a = 5, b = 5, c = 10;
5      printf("Logical AND \n(a==b)&&(c>b) is %d \n", (a==b)&&(c>b));
6      printf("Logical OR \n(a==b)|| (c<b) is %d \n", (a==b) || (c<b));
7      return 0;
8  }
9
```

OUTPUT:



```
C:\Users\Rolyx\Desktop\C\Exercise 2\logical operators.exe
Logical AND
(a==b)&&(c>b) is 1
Logical OR
(a==b)|| (c<b) is 1

Process returned 0 (0x0)   execution time : 0.702 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 2.3

3. write a program to read radius value from the keyboard and calculate the area of circle and print the result in both floating and exponential notation.

AIM :

To write a c program to calculate the area of circle and print result in float and exponential notation.

ALGORITHM :

STEP 1 : start

STEP 2 : declare a variable

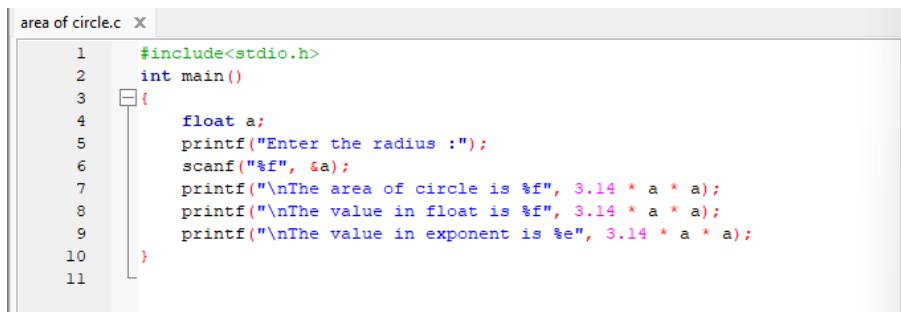
STEP 3 : get radius from the user

STEP 4 : calculate the area of circle = $3.14 * a * a$

STEP 5 : Then print the area in both float and exponent notation.

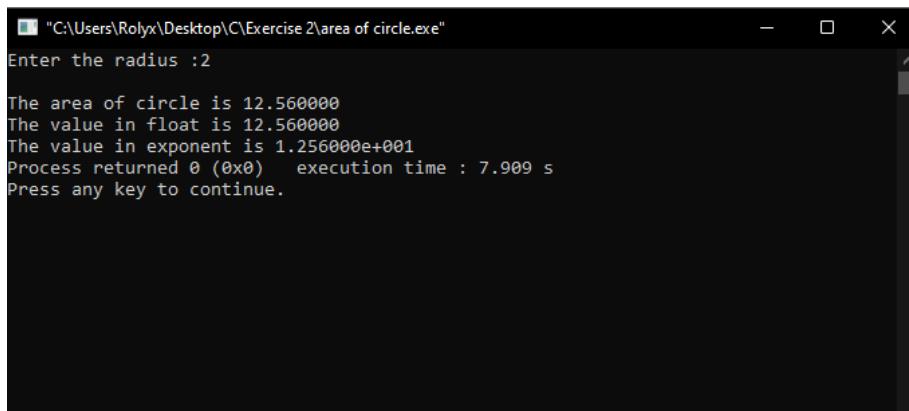
STEP 6 : stop.

SOURCE CODE :



```
area of circle.c x
1 #include<stdio.h>
2 int main()
3 {
4     float a;
5     printf("Enter the radius :");
6     scanf("%f", &a);
7     printf("\nThe area of circle is %f", 3.14 * a * a);
8     printf("\nThe value in float is %f", 3.14 * a * a);
9     printf("\nThe value in exponent is %e", 3.14 * a * a);
10 }
11
```

OUTPUT :



```
C:\Users\RoLyx\Desktop\C\Exercise 2\area of circle.exe
Enter the radius :2
The area of circle is 12.560000
The value in float is 12.560000
The value in exponent is 1.256000e+001
Process returned 0 (0x0) execution time : 7.909 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 2.4

4. write a program to calculate simple interest.

AIM :

To write a C program to calculate simple interest

ALGORITHM :

STEP 1 : Start

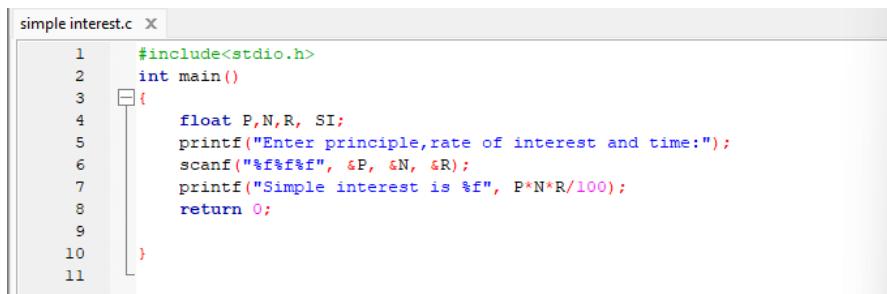
STEP 2 : declare P, N, R, SI

STEP 3 : Get the values of Principle, rate of interest and time from the user.

STEP 4 : calculate simple interest, $P * N * R / 100$
using this formula and display the value

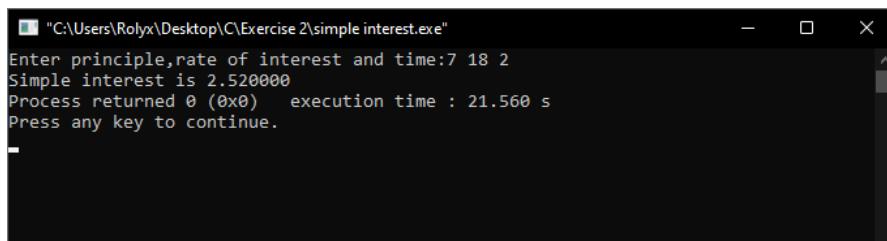
STEP 5 : Stop

SOURCE CODE :



```
simple interest.c x
1 #include<stdio.h>
2 int main()
3 {
4     float P,N,R, SI;
5     printf("Enter principle,rate of interest and time:");
6     scanf("%f%f%f", &P, &N, &R);
7     printf("Simple interest is %f", P*N*R/100);
8     return 0;
9
10}
11
```

OUTPUT :



```
"C:\Users\RoLyx\Desktop\C\Exercise 2\simple interest.exe"
Enter principle,rate of interest and time:7 18 2
Simple interest is 2.520000
Process returned 0 (0x0) execution time : 21.560 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 2.5

5. write a program to Convert temperature [Fahrenheit - centigrade and vice-versa]

AIM :

To write a program to convert temperature [Fahrenheit - centigrade and vice versa]

ALGORITHM :

STEP 1 : Start

STEP 2 : declare c, f, c_1 , f_1 ,

STEP 3 : Get the value of celsius from the user and convert into Fahrenheit using the formula $f = (c * 9/5) + 32$ and display the value.

STEP 4 : get the value of fahrenheit from the user and convert it into celsius using the formula $c_1 = (f_1 - 32) * 5/9$ and display the value.

STEP 5 : Stop.

SOURCE CODE :

```
fahrenheit centigrade and viceversa.c X
1 #include<stdio.h>
2 void main()
3 {
4     float c,f,cl,fl;
5     printf("Enter the Celcius:");
6     scanf("%f",&c);
7     f=(c*9/5)+32;
8     printf("The Fahrenheit is:%f",f);
9     printf("\n\nFahrenheit:");
10    scanf("%f",&fl);
11    cl=(fl-32)*5/9;
12    printf("The Celcius is:%f",cl);
13 }
14
```

OUTPUT :

```
C:\Users\Rolyx\Desktop\C\Exercise 2\fahrenheit centigrade and viceversa.exe"
Enter the Celcius:25
The Fahrenheit is:77.000000

Fahrenheit:128
The Celcius is:53.333332
Process returned 24 (0x18)  execution time : 3.436 s
Press any key to continue.
```

RESULT:

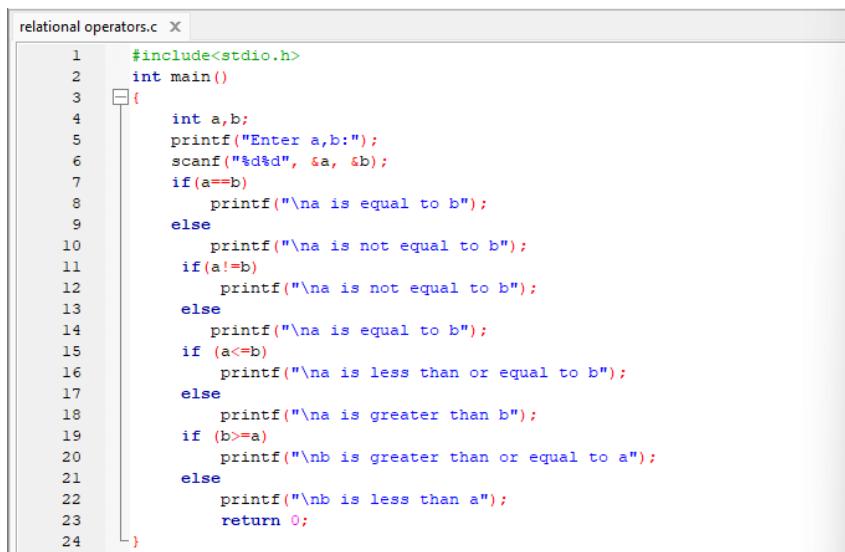
The Desired Output has been successfully Verified and Executed.

Operators

EXERCISE 3.1

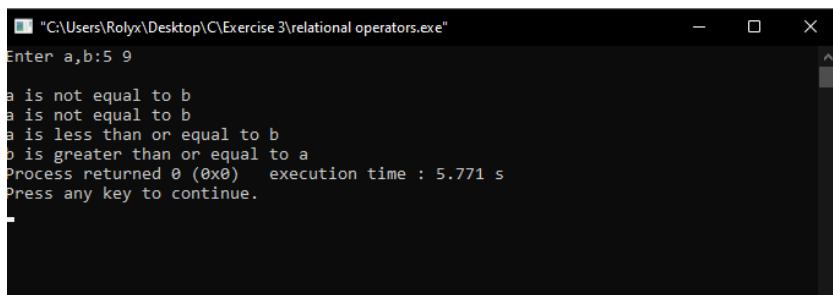
Ex. No. 3 19/04/21	OPERATORS
	<p>3. Write a program to demonstrate relational operators ($<=$, $>=$, $=$, \neq)</p> <p>AIM:</p> <p>To write a C program to demonstrate relational operators ($<=$, $>=$, $=$, \neq)</p> <p>ALGORITHM:</p> <p>STEP 1 : Start</p> <p>STEP 2: declare a and b</p> <p>STEP 3: Get the values of a and b from the user</p> <p>STEP 4: using the relational operators ($<=$, $>=$, $=$, \neq) check the Condition and display result</p> <p>STEP 5 : Stop</p>

SOURCE CODE :



```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b;
5     printf("Enter a,b:");
6     scanf("%d%d", &a, &b);
7     if(a==b)
8         printf("\na is equal to b");
9     else
10        printf("\na is not equal to b");
11        if(a!=b)
12            printf("\na is not equal to b");
13        else
14            printf("\na is equal to b");
15            if (a<=b)
16                printf("\na is less than or equal to b");
17                else
18                    printf("\na is greater than b");
19                    if (b>=a)
20                        printf("\nb is greater than or equal to a");
21                        else
22                            printf("\nb is less than a");
23                            return 0;
24 }
```

OUTPUT :



```
C:\Users\RoLyx\Desktop\C\Exercise 3\relational operators.exe"
Enter a,b:5 9
a is not equal to b
a is not equal to b
a is less than or equal to b
b is greater than or equal to a
Process returned 0 (0x0) execution time : 5.771 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 3.2

2. write a program to check equivalence of two numbers using Conditional operator

AIM:

To write a C program to check equivalence of two numbers using Conditional operator.

ALGORITHM:

STEP 1:

Start

STEP 2 : Declare any two int variable s and D

STEP 3 : Get the value of s and D from the user

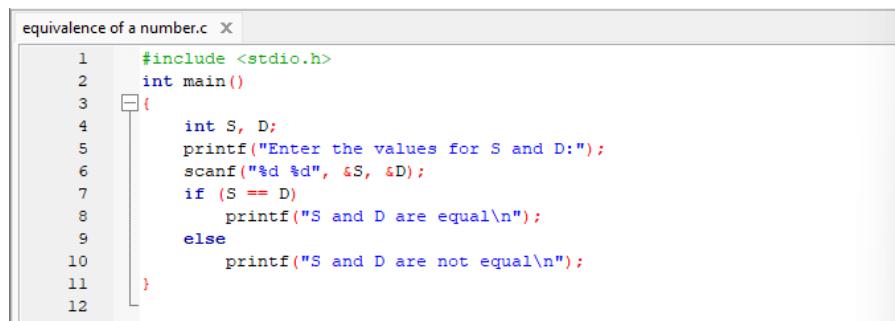
STEP 4 : check the values using conditional operator

STEP 5 : If $s == D$ then Print "s and D are equal"

STEP 6 : If s is not equal to D then Print "s and D are not equal"

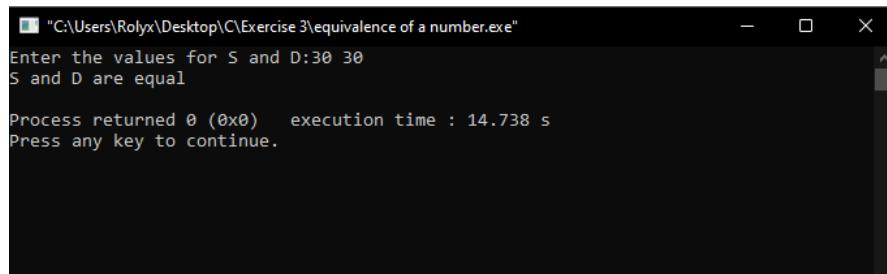
STEP 7 : Stop

SOURCE CODE :



```
equivalence of a number.c  X
1 #include <stdio.h>
2 int main()
3 {
4     int S, D;
5     printf("Enter the values for S and D:");
6     scanf("%d %d", &S, &D);
7     if (S == D)
8         printf("S and D are equal\n");
9     else
10        printf("S and D are not equal\n");
11 }
12 }
```

OUTPUT :



```
C:\Users\RoLyx\Desktop\C\Exercise 3\equivalence of a number.exe
Enter the values for S and D:30 30
S and D are equal

Process returned 0 (0x0)   execution time : 14.738 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 3.3

3. Write a program to demonstrate pre increment and post increment ($++a$, $a++$ where a is a value to be initialized)

AIM :

To write the C program to demonstrate pre increment and post increment

ALGORITHM:

Step 1 : Start

Step 2 : Declare an int variable 'a'

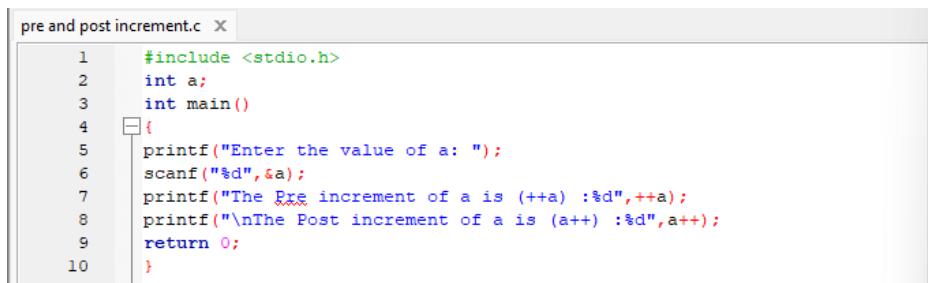
Step 3 : Get the value of 'a' from the user

Step 4 : Pre increment [$++a$] the value and display it

Step 5 : Post increment [$a++$] the value and display it

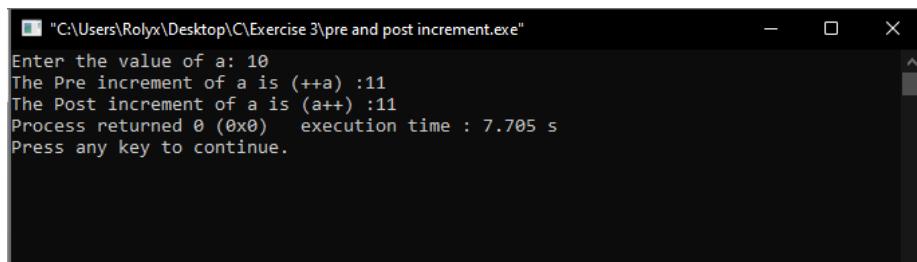
Step 6 : Stop

SOURCE CODE :



```
pre and post increment.c X
1 #include <stdio.h>
2 int a;
3 int main()
4 {
5     printf("Enter the value of a: ");
6     scanf("%d",&a);
7     printf("The Pre increment of a is (++a) :%d",++a);
8     printf("\nThe Post increment of a is (a++) :%d",a++);
9     return 0;
10 }
```

OUTPUT :



```
C:\Users\RoLyx\Desktop\C\Exercise 3\pre and post increment.exe
Enter the value of a: 10
The Pre increment of a is (++a) :11
The Post increment of a is (a++) :11
Process returned 0 (0x0) execution time : 7.705 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 3.4

4. write a program to demonstrate Pre decrement and post decrement (-a, a- where a is a value to be initialized)

AIM :

TO write a C Program to demonstrate pre decrement and post decrement

ALGORITHM :

- Step 1 : Start
- Step 2 : Declare an int variable 'a'
- Step 3 : Get the value of 'a' from the user
- Step 4 : Pre decrement [-a] the value and display it
- Step 5 : Post decrement [a--] the value and display it
- Step 6 : Stop

SOURCE CODE :

```
Start here X pre and post decrement.c X
1 #include <stdio.h>
2 int a;
3 int main()
4 {
5     printf("Enter the value of a:");
6     scanf("%d",&a);
7     printf("The Pre decrement of a is (--a) :%d",--a);
8     printf("\nThe Post decrement of a is (a--) :%d",a--);
9     return 0;
10 }
11
```

OUTPUT :

```
"C:\Users\RoLyx\Desktop\C\Exercise 3\pre and post decrement.exe"
Enter the value of a:10
The Pre decrement of a is (--a) :9
The Post decrement of a is (a--) :9
Process returned 0 (0x0) execution time : 2.211 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 3.5

5 write a program for computing the volume of sphere, cone and cylinder assume that dimensions are integer's use type casting where ever necessary

AIM :

TO write a c program to compute volume of sphere, cone and cylinder using type casting

ALGORITHM :

- STEP 1 : start
- STEP 2 : initialise a, b, c, d, e and include pi 3.14
- STEP 3 : Get the radius of sphere from the user
- STEP 4 : Calculate volume of the sphere (a_1) using the formula $a_1 = \frac{4}{3} \pi r^3$
- STEP 5 : Then display the volume of sphere in cubic cm.
- STEP 6 : Get the radius and height of the cone from the user
- STEP 7 : Calculate the volume cone (a_2) using the formula, $a_2 = \frac{1}{3} \pi r^2 h$
- STEP 8 : Then display the volume of cone in cubic cm.
- STEP 9 : Get the value of radius and height of the cylinder from the user.
- STEP 10 : calculate the volume of cylinder (a_3) using the formula, $a_3 = \pi r^2 h$
- STEP 11 : Then display the volume of cylinder in cubic cm
- STEP 12 : Stop

SOURCE CODE :

```
vol_of_sphere_cone_cylinder.c X
1  #include<stdio.h>
2  #define pi 3.14
3  int main()
4  {
5      int a,b,c,d,e;
6      printf("SPHERE\nEnter the radius of sphere : ");
7      scanf("%d",&a);
8      int a1;
9      a1=4*pi*a*a/3;
10     printf("\nVolume of sphere is %d cubic cm",a1);
11     printf("\nCONE\nEnter the radius and height of cone : ");
12     scanf("%d %d",&b,&c);
13     int a2;
14     a2=pi*b*b*c/3;
15     printf("The volume of cone is %d cubic cm",a2);
16     printf("\nCYLINDER\nEnter the radius and height of cylinder : ");
17     scanf("%d %d",&d,&e);
18     int a3;
19     a3=pi*d*d*e;
20     printf("The volume of cylinder is %d cubic cm",a3);
21     return 0;
22 }
```

OUTPUT :

```
C:\Users\Rolyx\Desktop\C\Exercise 3\vol_of_sphere_cone_cylinder.exe
SPHERE
Enter the radius of sphere : 5
Volume of sphere is 523 cubic cm
CONE
Enter the radius and height of cone : 6 7
The volume of cone is 263 cubic cm
CYLINDER
Enter the radius and height of cylinder : 4 3
The volume of cylinder is 150 cubic cm
Process returned 0 (0x0) execution time : 5.945 s
Press any key to continue.
```

RESULT:

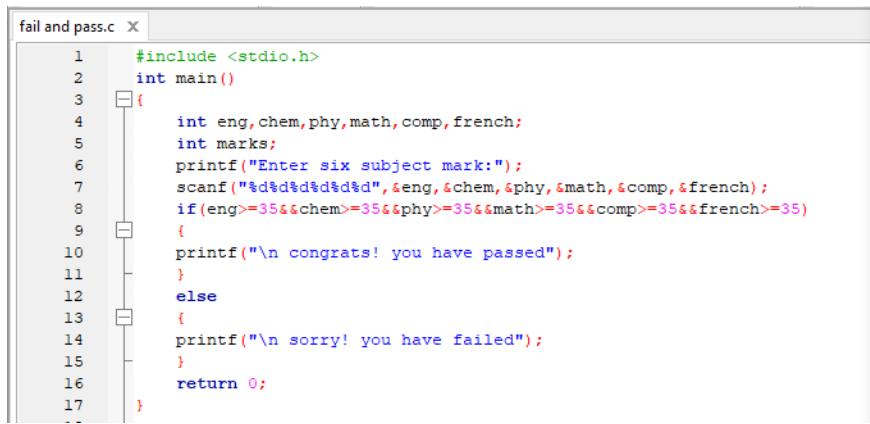
The Desired Output has been successfully Verified and Executed.

DECISION STATEMENTS

EXERCISE 4.1

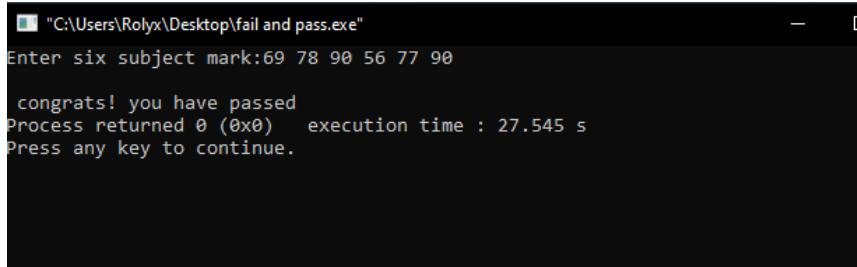
Ex. No. 4	DECISION STATEMENTS
DATE: 26/04/21	
1. write a program to read marks of a student in six subjects and print whether pass or fail (using if-else).	<p>AIM :</p> <p>To read marks of a student in six subjects and print whether the student is pass or fail</p> <p>ALGORITHM :</p> <p>Step 1 : start</p> <p>Step 2 : Initialise six int Subjects for marks.</p> <p>Step 3 : Get the marks of six subjects from the user.</p> <p>Step 4 : check each subject mark, if the value of mark is greater than or equal to 35</p> <p>Step 5 : print, "congrats! you have passed"</p> <p>Step 6 : Else print "sorry! you have failed."</p> <p>Step 7 : stop</p>

SOURCE CODE :



```
fail and pass.c X
1 #include <stdio.h>
2 int main()
3 {
4     int eng,chem,phy,math,comp,french;
5     int marks;
6     printf("Enter six subject mark:");
7     scanf("%d%d%d%d%d", &eng, &chem, &phy, &math, &comp, &french);
8     if(eng>=35&&chem>=35&&phy>=35&&math>=35&&comp>=35&&french>=35)
9     {
10        printf("\n congrats! you have passed");
11    }
12    else
13    {
14        printf("\n sorry! you have failed");
15    }
16    return 0;
17 }
```

OUTPUT :



```
C:\Users\Rolyx\Desktop\fail and pass.exe
Enter six subject mark:69 78 90 56 77 90
congrats! you have passed
Process returned 0 (0x0) execution time : 27.545 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 4.2

2. write a program to calculate roots of quadratic equation (using if else)

AIM :

To calculate roots of quadratic equation

ALGORITHM:

Step 1 : Start

Step 2 : Initialize float variable $x, y, z, \det,$
 $\text{root}_1, \text{root}_2, \text{realPart}, \text{imgPart}$
and include `<math.h>`

Step 3 : Get x, y, z from the user

Step 4 : Compute $\det = (y * y) - (x * z)$

Step 5 : If $\det > 0$ then Compute $\text{root}_1 = (-y + \sqrt{\det}) / (2 * x)$ and $\text{root}_2 = (-y - \sqrt{\det}) / (2 * x)$

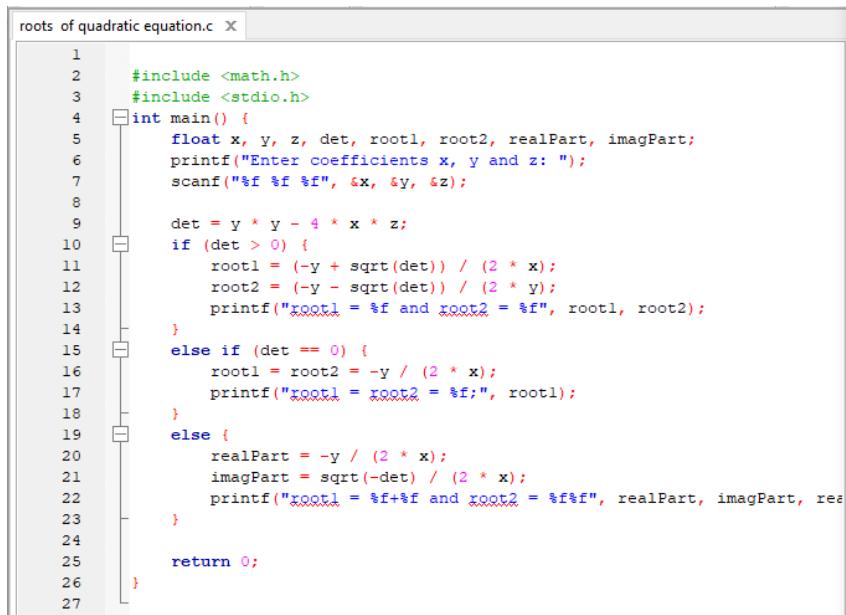
Step 6 : If $\det == 0$ then Compute $\text{root}_1 = \text{root}_2 = -y / (2 * x)$

Step 7 : Else Compute $\text{real} = -y / (2 * x)$ and
 $\text{img} = \sqrt{-\det} / (2 * x)$

Step 8 : Print the result obtained from
the condition Step 5, Step 6 and Step 7

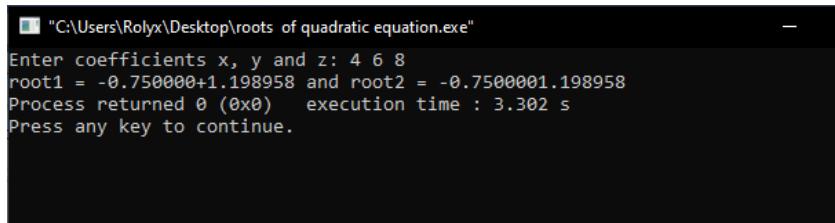
Step 9 : Stop.

SOURCE CODE :



```
roots of quadratic equation.c X
1  #include <math.h>
2  #include <stdio.h>
3
4  int main() {
5      float x, y, z, det, root1, root2, realPart, imagPart;
6      printf("Enter coefficients x, y and z: ");
7      scanf("%f %f %f", &x, &y, &z);
8
9      det = y * y - 4 * x * z;
10     if (det > 0) {
11         root1 = (-y + sqrt(det)) / (2 * x);
12         root2 = (-y - sqrt(det)) / (2 * x);
13         printf("root1 = %f and root2 = %f", root1, root2);
14     }
15     else if (det == 0) {
16         root1 = root2 = -y / (2 * x);
17         printf("root1 = root2 = %f", root1);
18     }
19     else {
20         realPart = -y / (2 * x);
21         imagPart = sqrt(-det) / (2 * x);
22         printf("root1 = %f+%fj and root2 = %f-%fj", realPart, imagPart, realPart, imagPart);
23     }
24
25
26 }
27
```

OUTPUT :



```
C:\Users\Rolyx\Desktop\roots of quadratic equation.exe
Enter coefficients x, y and z: 4 6 8
root1 = -0.750000+1.98958 and root2 = -0.7500001.198958
Process returned 0 (0x0)   execution time : 3.302 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 4.3

3. write a program to calculate electricity bill
Read starting and ending meter reading. The charges are as follows.
No of unit Consumed Rate in (RS)
1-100 1.50 per unit
101-300 2.00 per unit for excess of 100 units
301-500 2.50 per unit for excess of 300 unit
501-above 3.25 per unit for excess of 500 units.

AIM:

To calculate the electricity Bill by getting starting and ending meter reading and Compute based on given data

ALGORITHM :

Step 1 : Start

Step 2 : Declare unit and amount

Step 3 : Get the units used from the user

Step 4 : If unit is > 500 , calculate $amt = (\text{unit} * 3.25)$ and print amt.

Step 5 : else If unit is ≤ 500 and > 300 , Compute $amt = (\text{unit} * 2.50)$ and Print amt

Step 6 : else if unit is ≤ 300 and > 100 , Compute $amt = (\text{unit} * 2.00)$ and Print amt.

Step 7 : else if unit is < 100 , Compute $amt = (\text{unit} * 1.50)$ and Print amt.

Step 8 : stop.

SOURCE CODE :

```
electricity bill.c x
1 #include<stdio.h>
2 int main()
3 {
4     int a;
5     float b,c;
6     printf("Enter the units used : ");
7     scanf("%d",&a);
8     if(a>=1&&a<=100)
9     {
10         c=a*1.5;
11     }
12     else if(a>=101&&a<=300)
13     {
14         b=a-100;
15         c=2.0*b+1.5*100;
16     }
17     else if(a>=301&&a<=500)
18     {
19         b=a-300;
20         c=2.5*b+200*2+1.5*100;
21     }
22     else
23     {
24         b=a-500;
25         c=3.25*b+200*2.5+200*2.5+1.5*100;
26     }
27     printf("\nNumber of units : %d",a);
28     printf("\nTotal amount    : %f",c);
29     return 0;
30 }
```

OUTPUT :

```
C:\Users\RoLyx\Desktop\C\Exercise 4\electricity bill.exe
Enter the units used : 78
Number of units : 78
Total amount    : 117.000000
Process returned 0 (0x0) execution time : 7.961 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

SWITCH OPERATIONS

EXERCISE 5.1

Ex. No. 5

DATE: 31/4/21

SWITCH OPERATIONS

1. write a program to perform arithmetic operations using switch case.

AIM :

To write a c program to perform arithmetic operations

ALGORITHM :

Step 1 : Start

Step 2 : Initialize a, b and operation

Step 3 : Get the values of a and b from the user

Step 4 : Then get the choice of operation from the user.

Step 5 : For case '1' compute sum, a+b and break the statement.

Step 6 : For case '2' compute Difference, a-b and break the statement.

Step 7 : For Case '3' Compute multiplication, a*b and break the statement

Step 8 : For Case '4' Compute division, a/b and break the statement

Step 9 : For default, print "Enter Valid Input"

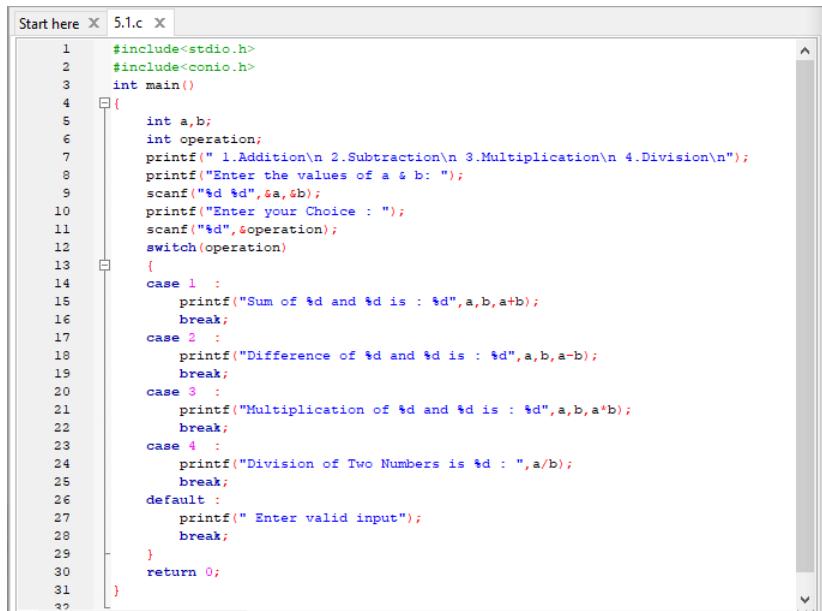
Step 10 : Print the values obtained from

step 5 to step 8 if default
statement is not occurred step 8
if default statement is not
occurred

Step 11 :

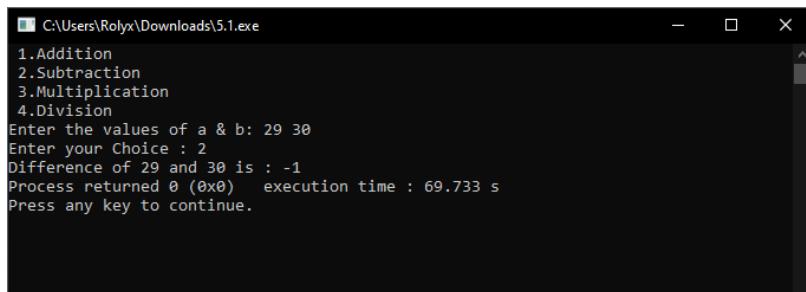
Stop

SOURCE CODE :



```
Start here | 5.1.c | X
1 #include<stdio.h>
2 #include<conio.h>
3 int main()
4 {
5     int a,b;
6     int operation;
7     printf(" 1.Addition\n 2.Subtraction\n 3.Multiplication\n 4.Division\n");
8     printf("Enter the values of a & b: ");
9     scanf("%d %d",&a,&b);
10    printf("Enter your Choice : ");
11    scanf("%d",&operation);
12    switch(operation)
13    {
14        case 1 :
15            printf("Sum of %d and %d is : %d",a,b,a+b);
16            break;
17        case 2 :
18            printf("Difference of %d and %d is : %d",a,b,a-b);
19            break;
20        case 3 :
21            printf("Multiplication of %d and %d is : %d",a,b,a*b);
22            break;
23        case 4 :
24            printf("Division of Two Numbers is %d : ",a/b);
25            break;
26        default :
27            printf(" Enter valid input");
28            break;
29    }
30    return 0;
31 }
```

OUTPUT :



```
C:\Users\Rolyx\Downloads\5.1.exe
1.Addition
2.Subtraction
3.Multiplication
4.Division
Enter the values of a & b: 29 30
Enter your Choice : 2
Difference of 29 and 30 is : -1
Process returned 0 (0x0)   execution time : 69.733 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 5.2

2. write a program to display colour using switch case (VIBGYOR)

AIM : To display Colours of VIBGYOR using Switch Case

ALGORITHM :

Step 1 : Start

Step 2 : initialize an 'char' letter and get its value from user under 'VIBGYOR'

Step 3 : Declare letter inside the switch case

Step 4 : FOR Case V, Print "violet" and break the statement

Step 5 : FOR Case I, Print "Indigo" and break the statement

Step 6 : FOR Case B, Print "Blue" and break the statement

Step 7 : For Case G, Print "Green" and break the statement

Step 8 : For Case Y, Print "Yellow" and break the statement

Step 9 : For Case O, print "orange" and break the statement

Step 10 : For Case R, print "Red" and break the statement

Step 11 : FOR default, print "Not a letter from VIBGYOR"

Step 12: Print the values obtained from step 4 to
step 10 if default statement is not occurred

Step 13: Stop

SOURCE CODE :

The screenshot shows a code editor window with three tabs at the top: "Start here", "5.1.c", and "VIBGYOR.c". The "VIBGYOR.c" tab is active, displaying the following C code:

```
1 #include <stdio.h>
2 int main()
3 {
4     char ch;
5     printf("Enter a letter from VIBGYOR :");
6     scanf("%c",&ch);
7     switch(ch)
8     {
9         case 'V':
10        printf("Violet");
11        break;
12        case 'I':
13        printf("Indigo");
14        break;
15        case 'B':
16        printf("Blue");
17        break;
18        case 'G':
19        printf("Green");
20        break;
21        case 'Y':
22        printf("Yellow");
23        break;
24        case 'O':
25        printf("Orange");
26        break;
27        case 'R':
28        printf("Red");
29        break;
30    default:
31        printf("Not a letter from VIBGYOR");
32    }
33 }
```

OUTPUT :

The screenshot shows a terminal window with the following text:

```
C:\Users\Roxy\Downloads\VIBGYOR.exe
Enter a letter from VIBGYOR :S
Not a letter from VIBGYOR
Process returned 0 (0x0)   execution time : 8.032 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 5.3

3. write a program to display vowels and consonants using switch case.

AIM:

To write a c Program to display vowels and consonants using Switch Case

ALGORITHM:

Step1 : Start

Step2 : Initialize "char" ch and get the value of the char from the user

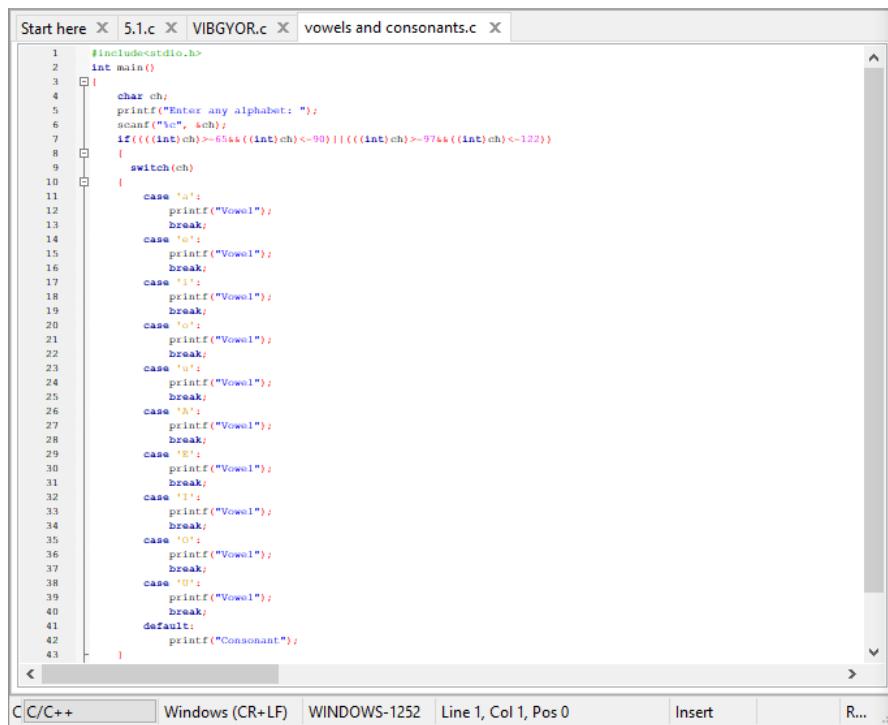
Step3 : declare ch in switch case

Step4 : FOR Case 'a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U' print "vowel" and break the statement

Step5 : FOR default Print "Consonant"

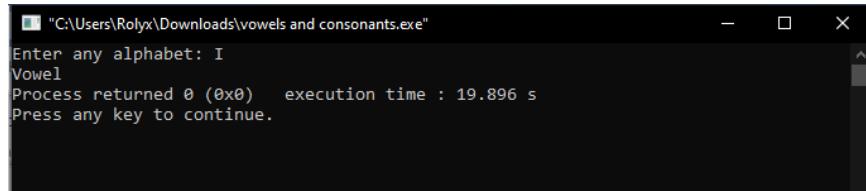
Step6 : Stop

SOURCE CODE :



```
Start here X 5.1.c X VIBGYOR.c X vowels and consonants.c X
1 #include<stdio.h>
2 int main()
3 {
4     char ch;
5     printf("Enter any alphabet: ");
6     scanf("%c", &ch);
7     if(((int)ch)>=65&&((int)ch)<=90)||(((int)ch)>=97&&((int)ch)<=122)
8     {
9         switch(ch)
10        {
11             case 'a':
12                 printf("Vowel");
13                 break;
14             case 'e':
15                 printf("Vowel");
16                 break;
17             case 'i':
18                 printf("Vowel");
19                 break;
20             case 'o':
21                 printf("Vowel");
22                 break;
23             case 'u':
24                 printf("Vowel");
25                 break;
26             case 'A':
27                 printf("Vowel");
28                 break;
29             case 'E':
30                 printf("Vowel");
31                 break;
32             case 'I':
33                 printf("Vowel");
34                 break;
35             case 'O':
36                 printf("Vowel");
37                 break;
38             case 'U':
39                 printf("Vowel");
40                 break;
41         default:
42             printf("Consonant");
43     }
44 }
C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert R... 
```

OUTPUT :



```
"C:\Users\RoLyx\Downloads\vowels and consonants.exe"
Enter any alphabet: I
Vowel
Process returned 0 (0x0)  execution time : 19.896 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 5.4

4. write a program to display names of days in a week using switch case

AIM :

TO write a c Program to display names of days in a week using switch case

ALGORITHM:

step 1 : start

Step 2 : Declare week and get the value of week from the user

Step 3 : Delare week in switch case

Step 4 : FOR Case 1, print "Monday" and break the statement

Step 5 : FOR Case 2, Print "Tuesday" and break the statement

Step 6 : FOR Case 3, print "wednesday" and break the statement

Step 7 : FOR Case 4, Print "Thursday" and break the statement

Step 8 : FOR Case 5, Print "Friday" and break the statement

Step 9 : FOR Case 6, print "saturday" and break the statement

Step 10 : FOR Case 7, Print "Sunday" and

break the Statement

Step 11 : If any of the above case doesn't
match then print default, "Invalid
input! Please enter week number
between 1-7".

Step 12 : Stop

SOURCE CODE :

```
Start here X 5.1.c X VIBGYOR.c X vowels and consonants.c X days of week.c X
1 #include<stdio.h>
2 int main()
3 {
4     int week;
5     printf("Enter week number(1-7): ");
6     scanf("%d", &week);
7     switch(week)
8     {
9         case 1:
10            printf("Monday");
11            break;
12        case 2:
13            printf("Tuesday");
14            break;
15        case 3:
16            printf("Wednesday");
17            break;
18        case 4:
19            printf("Thursday");
20            break;
21        case 5:
22            printf("Friday");
23            break;
24        case 6:
25            printf("Saturday");
26            break;
27        case 7:
28            printf("Sunday");
29            break;
30        default:
31            printf("Invalid input! Please enter week number between 1-7.");
32    }
33
34    return 0;
35 }
```

OUTPUT :

```
"C:\Users\Rolyx\Downloads\days of week.exe"
Enter week number(1-7): 4
Thursday
Process returned 0 (0x0)   execution time : 10.503 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

Basic Loop operations Do the Following Programs Using for, while, do-while loops

EXERCISE 6.1

EX. NO 6

DATE: 3/5/21

BASIC LOOP OPERATIONS DO THE FOLLOWING PROGRAMS USING FOR, WHILE, DO-WHILE LOOPS

1. write a program to calculate sum of individual digits of a given number

AIM :

To write a c program to calculate sum of individual digits of a given number

ALGORITHM :

Step 1 : Start

Step 2 : Initialise n,a and sum=0

Step 3 : Get the value of n from the user

Step 4 : while ($n \neq 0$)

{

$a = n \% 10;$

$sum = sum + a;$

$n = n / 10;$

}

and print sum

Step 5 : Stop

SOURCE CODE :

```
6.1.c X
1 #include<stdio.h>
2 void main()
3 {
4     int n,sum,a;
5     printf("Enter a number:");
6     scanf("%d",&n);
7     sum=0;
8     while(n!=0)
9     {
10        a=n%10;
11        sum=sum+a;
12        n=n/10;
13    }
14    printf("sum is:%d",sum);
15 }
16
```

OUTPUT :

```
C:\Users\Rolyx\Desktop\C\6.1.exe
Enter a number:352
sum is:10
Process returned 9 (0x9)  execution time : 11.948 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 6.2

2. write a program to check whether given number is Palindrome or not

AIM:

To check whether given number is Palindrome or not.

ALGORITHM :

Step 1: Start

Step 2: Initialise $n, n_1, \text{rem}, \text{rev} = 0$

Step 3: Get the value of n from the user

Step 4: while $n > 0$, do $\text{rem} = n \% 10$, $\text{rev} = \text{rev} * 10 + \text{rem}$,
 $n = n / 10$

Step 5: if ($n_1 == \text{rev}$), print "Given number is a Palindromic number".

Step 6: Else, Print "Given number is not a Palindromic number".

Step 7: Stop

SOURCE CODE :

```
6.1.c x 6.2.c x
1 #include <stdio.h>
2 void main()
3 {
4     int n, nl, rev = 0, rem;
5     printf("Enter any number: ");
6     scanf("%d", &n);
7     nl = n;
8         while (n > 0)
9     {
10         rem = n % 10;
11         rev = rev * 10 + rem;
12         n = n / 10;
13     }
14     if (nl == rev)
15     {
16         printf("Given number is a palindromic number");
17     }
18     else
19     {
20         printf("Given number is not a palindromic number");
21     }
22 }
```

OUTPUT :

```
C:\Users\RoLyx\Desktop\CL6.2.exe
Enter any number: 232
Given number is a palindromic number
Process returned 36 (0x24)   execution time : 9.383 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 6.3

3 write a program to print prime numbers in the given range

AIM:

To write a c program to print prime numbers in the given range

ALGORITHM :

Step 1 : Start

Step 2 : Declare low, high, i, flag

Step 3 : Get two number high, low from the user

Step 4 : while ($low < high$), $flag = 0$ then
declare for loop($i = 2$; $i \leq low/2$; $i++$)
then declare if ($low \% i == 0$) $flag = 1$ then give break statement

Step 5 : if ($flag == 0$) Print the numbers
then perform post increment of low

Step 6 : Stop

SOURCE CODE :

```
6.1.c X 6.2.c X 6.3.c X
1 #include <stdio.h>
2 void main() {
3     int low, high, i, flag;
4     printf("Enter two numbers(intervals): ");
5     scanf("%d %d", &low, &high);
6     printf("Prime numbers between %d and %d are: ", low, high);
7
8     while (low < high)
9     {
10         flag = 0;
11         for (i = 2; i<= low / 2; ++i) {
12             if (low % i == 0) {
13                 flag = 1;
14                 break;
15             }
16         }
17
18         if (flag == 0)
19             printf("%d ", low);
20         ++low;
21     }
}
```

OUTPUT:

```
C:\Users\RoLyX\Desktop\C\6.3.exe
Enter two numbers(intervals): 1 15
Prime numbers between 1 and 15 are: 1 2 3 5 7 11 13
Process returned 15 (0xF)   execution time : 126.672 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 6.4

4. write a program to display multiplication tables of 6 (from 1 to 10)

AIM :

To write a c program to display multiplication tables of from 1 to 10

ALGORITHM :

Step 1 : Start

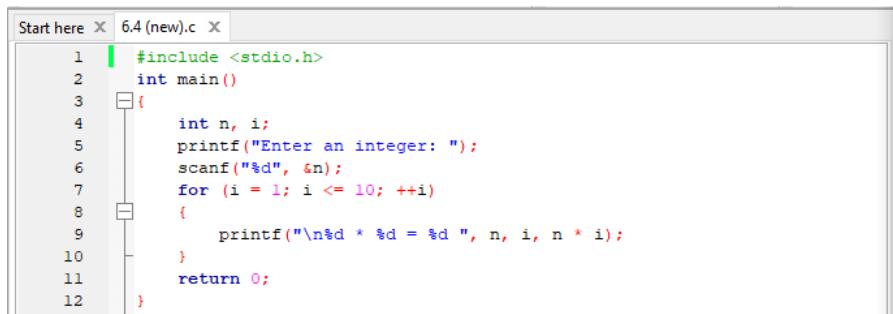
Step 2 : Initialize n and $i = 1$

Step 3 : Get the value of n from the user

Step 4 : For $i = 1, i < 10$ and $i++$, Print $n * i$

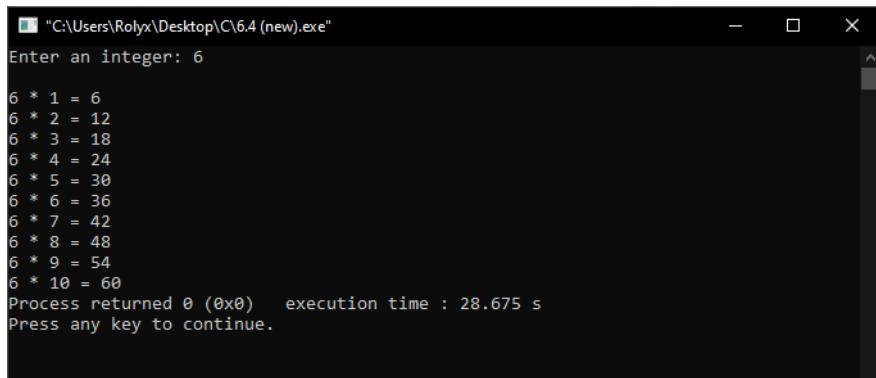
Step 5 : Stop

SOURCE CODE :



```
Start here x 6.4 (new).c x
1 #include <stdio.h>
2 int main()
3 {
4     int n, i;
5     printf("Enter an integer: ");
6     scanf("%d", &n);
7     for (i = 1; i <= 10; ++i)
8     {
9         printf("\n%d * %d = %d ", n, i, n * i);
10    }
11 }
12 }
```

OUTPUT :



```
"C:\Users\Rolyx\Desktop\C\6.4 (new).exe"
Enter an integer: 6
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54
6 * 10 = 60
Process returned 0 (0x0)  execution time : 28.675 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

Advanced loops

EXERCISE 7.1

Ex. No. 7

ADVANCED LOOPS

DATE: 10/5/21

1. write a program to print the Fibonacci Series for given 'N' value

AIM:

TO write a c program to print the fibonacci series for given 'n' value

ALGORITHM :

Step 1 : Start

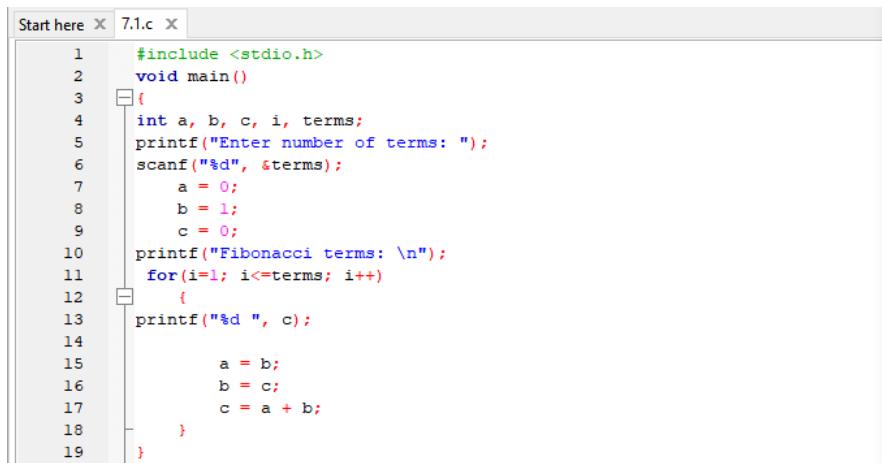
Step 2 : Declare $a=0$, $b=1$, $c=0$, i and terms.

Step 3 : Get Number of terms from the user

Step 4 : For ($i=1$, $i \leq$ terms, $i+1$), Print value of c and swap $a=b$, $b=c$, Compute $c=a+b$

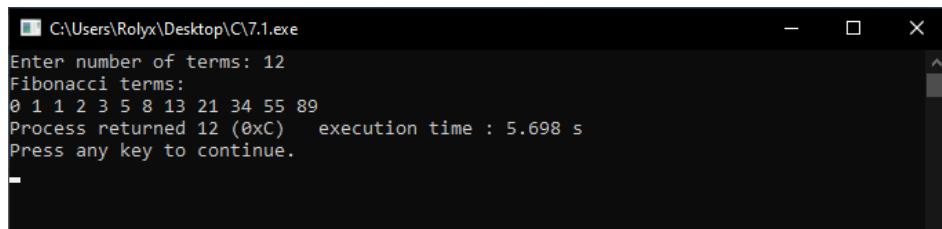
Step 5 : Stop

SOURCE CODE :



```
Start here 7.1.c
1 #include <stdio.h>
2 void main()
3 {
4     int a, b, c, i, terms;
5     printf("Enter number of terms: ");
6     scanf("%d", &terms);
7     a = 0;
8     b = 1;
9     c = 0;
10    printf("Fibonacci terms: \n");
11    for(i=1; i<=terms; i++)
12    {
13        printf("%d ", c);
14        a = b;
15        b = c;
16        c = a + b;
17    }
18 }
19 }
```

OUTPUT:



```
C:\Users\RoLyx\Desktop\C\7.1.exe
Enter number of terms: 12
Fibonacci terms:
0 1 1 2 3 5 8 13 21 34 55 89
Process returned 12 (0xC)   execution time : 5.698 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 7.2

2. write a program to check whether a given number is a fibonacci number or not.

AIM:

To check whether a given number is a fibonacci number or not.

ALGORITHM:

Step 1 : Start

Step 2 : initialise a, b, c, num

Step 3 : get the number from user

Step 4 : If ($\text{num} == 0$) || ($\text{num} == 1$)), then
Print "It is a fibonacci term."

Step 5 : Else declare a=0, b=1, c=a+b and
while ($c < \text{num}$) declare a=b, b=c
and compute c=a+b

Step 6 : If ($c == \text{num}$), then print "It is
a fibonacci term".

Step 7 : Else print "It is not a fibonacci
term."

Step 8 : Stop

SOURCE CODE :

```
7.1.c X 7.2.c X
1 #include<stdio.h>
2 void main()
3 {
4     int a,b,c,num;
5     printf("Enter any number: ");
6     scanf("%d", &num);
7     if((num==0) || (num==1))
8         printf("\n%d is a Fibonacci term",num);
9     else
10    {
11        a=0;
12        b=1;
13        c=a+b;
14        while(c<num)
15        {
16            a=b;
17            b=c;
18            c=a+b;
19        }
20        if(c==num)
21            printf("\n%d is a Fibonacci term",num);
22        else
23            printf("\n%d is not a Fibonacci term",num);
24    }
25 }
```

OUTPUT :

```
C:\Users\RoLyx\Desktop\C\7.2.exe
Enter any number: 8
8 is a Fibonacci term
Process returned 22 (0x16)   execution time : 6.234 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 7.3

3 write a program to read a number x and n then compute sum of the Geometric progression $1+x+x^2+x^3+\dots+x^n$

AIM:

To read x and n and Compute geometric progression as $1+x+x^2+x^3+\dots+x^n$

ALGORITHM:

Step 1: Start

Step 2: Initialise $n, x, i, \text{sum} = 0$ and get the values of n and x from the user.

Step 3: if ($x < 0 \text{ or } n < 0$), print "Illegal value".

Step 4: Else, for ($i = 0; i \leq n$ at $i++$, Compute $\text{sum} = \text{sum} + \text{pow}(x, i)$ using the header file `<math.h>`)

Step 5: Then print the result

Step 6: Stop

SOURCE CODE :

The screenshot shows a code editor window with three tabs at the top: 7.1.c, 7.2.c, and 7.3.c. The 7.3.c tab is active. The code is as follows:

```
1 #include <stdio.h>
2 #include <math.h>
3 void main()
4 {
5     int n, x, i, sum = 0;
6     printf("Enter the limit\n");
7     scanf("%d", &n);
8     printf("Enter the value of x\n");
9     scanf("%d", &x);
10    if(x < 0 || n < 0)
11    {
12        printf("illegal value");
13    }
14    else
15    {
16        for(i = 0; i <= n; i++)
17            sum=sum + pow(x, i);
18    }
19    printf("sum=%d", sum);
20 }
```

OUTPUT :

The screenshot shows a terminal window with the following output:

```
C:\Users\Rolyx\Desktop\C\7.3.exe
Enter the limit
4
Enter the value of x
2
sum=31
Process returned 6 (0x6)   execution time : 12.580 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 7.4

4. write a program to print the following formats

```
    1      *
   1 2    **
  1 2 3   ***
 1 2 3 4  **** *
```

AIM:

To print the patterns in the given format

ALGORITHM:

Step 1 : Start

Step 2 : Initialise i, j, k, l, n.

Step 3 : Get number of rows(n) from the user

Step 4 : for i=1; i<=n and i++ and for
j=1, j<=i and ++j, print "*" inside
for (j) loop and print "\n" in for
(i) loops

Step 5 : After new line sequence, for k=1, k<=n
and ++k and for l=1, l<=k, l++,
print l inside for(l) and print
"\n" for (k) loops

Step 6 : stop

SOURCE CODE :

The screenshot shows a code editor window with five tabs at the top: 7.1.c, 7.2.c, 7.3.c, 7.4.c, and 7.4.c (the active tab). The code is as follows:

```
1 #include<stdio.h>
2 void main()
3 {
4     int i,j,k,l,n;
5     printf("Enter number of rows:");
6     scanf("%d",&n);
7     for(i=1;i<=n;++i)
8     {
9         for(j=1;j<=i;++j)
10        {
11            printf(" *");
12        }
13        printf("\n");
14    }
15    printf("\n");
16    printf("\n");
17    for(k=1;k<=n;++k)
18    {
19        for(l=1;l<=k;++l)
20        {
21            printf("%d",l);
22        }
23        printf("\n");
24    }
25 }
```

OUTPUT:

The screenshot shows a terminal window with the following output:

```
C:\Users\RoLyx\Desktop\C\7.4.exe
Enter number of rows:4
*
**
***
****

1
12
123
1234

Process returned 4 (0x4)  execution time : 5.750 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

1-D Arrays

EXERCISE 8.1

Exc. No. 8

1-D ARRAYS

DATE: 17/05/21

1. write a program to store 10 elements in the 1-D array and print sum of the array.

AIM :

To store 10 elements in the 1-D array and Print Sum of the array

ALGORITHM :

Step 1 : Start

Step 2 : Declare n and get the size of the array from the user

Step 3 : Declare int arr(n); int i, sum = 0 and get the elements from the user

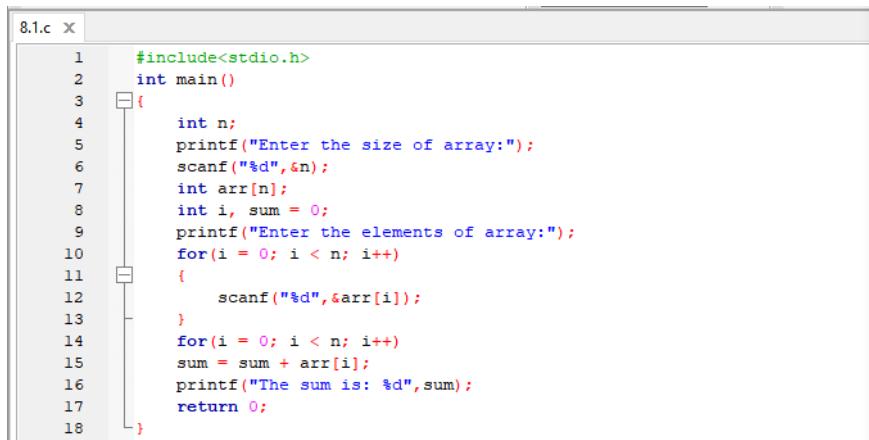
Step 4 : for (i=0; i < n; i++) read the element arr(i)

Step 5 : for i=0; i < n; i++, sum = sum + arr[i]

Step 6 : and print sum

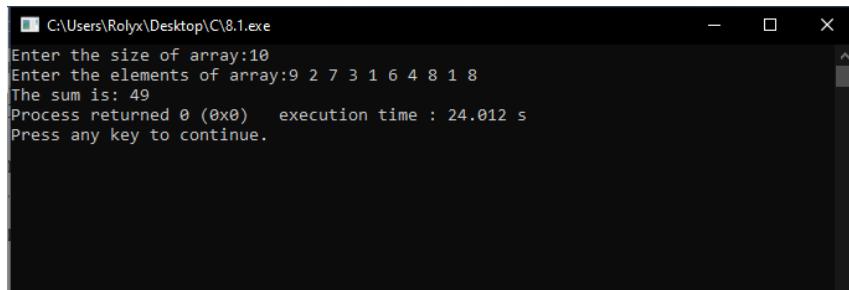
Step 7 : Stop

SOURCE CODE :



```
8.1.c X
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     printf("Enter the size of array:");
6     scanf("%d",&n);
7     int arr[n];
8     int i, sum = 0;
9     printf("Enter the elements of array:");
10    for(i = 0; i < n; i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    for(i = 0; i < n; i++)
15    sum = sum + arr[i];
16    printf("The sum is: %d",sum);
17    return 0;
18 }
```

OUTPUT :



```
C:\Users\Rolyx\Desktop\C\8.1.exe
Enter the size of array:10
Enter the elements of array:9 2 7 3 1 6 4 8 1 8
The sum is: 49
Process returned 0 (0x0) execution time : 24.012 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 8.2

2

write a program to print minimum and maximum elements in the 1-D array.

AIM:

TO print minimum and maximum elements in 1-D array.

ALGORITHM:

Step 1 : Start

Step 2 : Declare arr [max-size], int i,
max, min, size

Step 3 : get the size and elements of
the array from the user

Step 4 : for i=0; i< size; i++; read the
elements of array.

Step 5: Declare max = arr[0], min = arr[0].
for i=1; i< size; i++ if (arr[i] > max), max = arr[i] if
(arr[i] < min), min = arr[i] print
values of maximum and minimum
Element.

Step 6 : Stop

SOURCE CODE :

```
8.1.c x 8.3.c x 8.2.c x
1 #include <stdio.h>
2 #define max_size 100
3 int main()
4 {
5     int arr[max_size];
6     int i, max, min, size;
7     printf("Enter size of the array: ");
8     scanf("%d", &size);
9     printf("Enter elements in the array: ");
10    for(i=0; i<size; i++)
11    {
12        scanf("%d", &arr[i]);
13    }
14    max = arr[0];
15    min = arr[0];
16    for(i=1; i<size; i++)
17    {
18        if(arr[i] > max)
19        {
20            max = arr[i];
21        }
22        if(arr[i] < min)
23        {
24            min = arr[i];
25        }
26    }
27    printf("Maximum element = %d\n", max);
28    printf("Minimum element = %d", min);
29    return 0;
30 }
```

OUTPUT :

```
C:\Users\Roxy\Desktop\CV8.2.exe
Enter size of the array: 5
Enter elements in the array: 1 7 9 4 3
Maximum element = 9
Minimum element = 1
Process returned 0 (0x0) execution time : 13.759 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 8.3

3 write a program to count no. of positive numbers, negative numbers and zeros in the array.

AIM:

To Count no. of positive, negative numbers and zero in the array

ALGORITHM:

Step 1 : Start

Step 2 : Declare a [100], n ; i, Positive = 0, negative = 0, zero = 0 and get the size and elements of the array from the user.

Step 3 : for i=0; i<n, i++, read the elements of a [i] if (a[i]>0) positive++, else if (a[i]==0) zero++, else negative ++

Step 4 : Print the number of positive, negative numbers and zero count.

Step 5 : Stop

SOURCE CODE :

```
8.1.c x 8.3.c x
1 #include<stdio.h>
2 int main()
3 {
4     int a[100], n, i, positive=0, negative=0, zero=0;
5     printf("Enter array size: ");
6     scanf("%d", &n);
7     printf("Enter %d elements: ",n);
8
9     for(i=0; i<n; i++)
10    {
11        scanf("%d", &a[i]);
12        if(a[i]>0) positive++;
13        else if(a[i]==0) zero++;
14        else negative++;
15    }
16    printf("Total Number of Positive numbers in array:%d", positive);
17    printf("\nTotal Number of Negative numbers in array:%d", negative);
18    printf("\nZeros count is %d", zero);
19    return 0;
20 }
```

OUTPUT:

```
Enter array size: 5
Enter 5 elements: 5 -4 0 9 -3
Total Number of Positive numbers in array:2
Total Number of Negative numbers in array:2
Zeros count is 1
Process returned 0 (0x0)   execution time : 19.405 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 8.4

4. write a program to search the given element by using linear search

AIM:

To search the given elements by using line search

ALGORITHM:

Step1: Start

Step2: declare array (100), c, n and get the value of n from the user

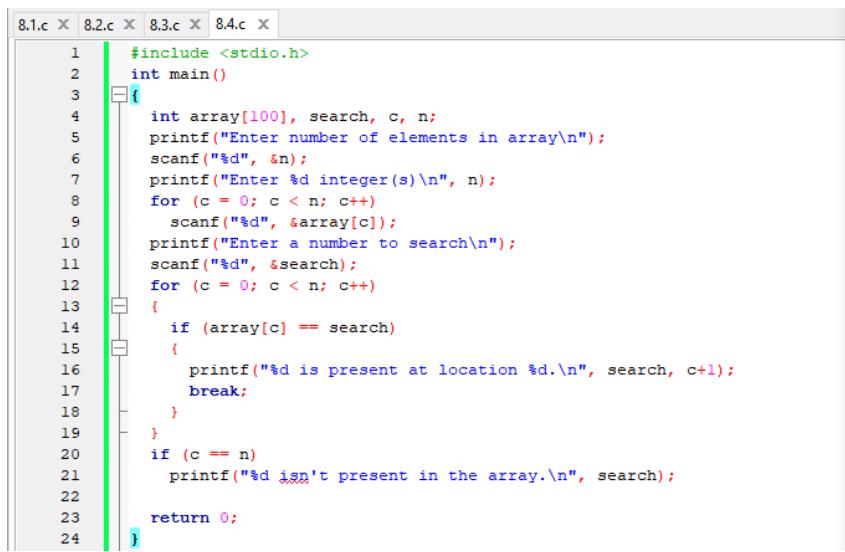
Step3: For $c = 0$, $c < n$ and $c++$, get the search element from the user

Step4: For $c = 0$, $c < n$ and $c++$, if $\text{array}[c] == \text{search}$ Print the position of the search element

Step5: If $c == n$, print the element isn't present in the array

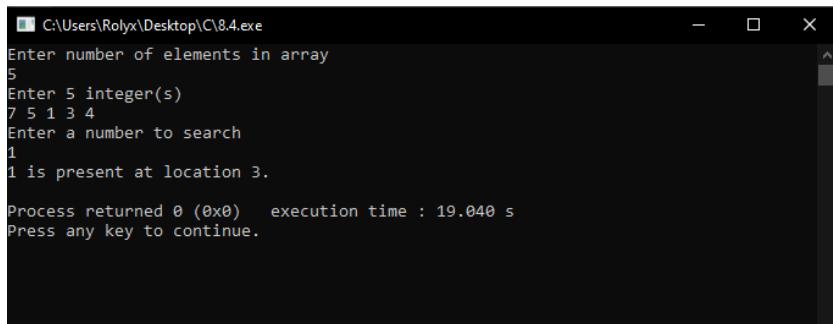
Step6 : Stop

SOURCE CODE AND OUTPUT:



```
8.1.c X 8.2.c X 8.3.c X 8.4.c X
1 #include <stdio.h>
2 int main()
3 {
4     int array[100], search, c, n;
5     printf("Enter number of elements in array\n");
6     scanf("%d", &n);
7     printf("Enter %d integer(s)\n", n);
8     for (c = 0; c < n; c++)
9         scanf("%d", &array[c]);
10    printf("Enter a number to search\n");
11    scanf("%d", &search);
12    for (c = 0; c < n; c++)
13    {
14        if (array[c] == search)
15        {
16            printf("%d is present at location %d.\n", search, c+1);
17            break;
18        }
19    }
20    if (c == n)
21        printf("%d isn't present in the array.\n", search);
22
23    return 0;
24 }
```

OUTPUT :



```
C:\Users\RoLyX\Desktop\C\8.4.exe
Enter number of elements in array
5
Enter 5 integer(s)
7 5 1 3 4
Enter a number to search
1
1 is present at location 3.

Process returned 0 (0x0)  execution time : 19.040 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 8.5

5. Write a program to sort the given elements using bubble sort technique

AIM :

To sort the given elements using bubble sort technique

ALGORITHM :

Step 1 : Start

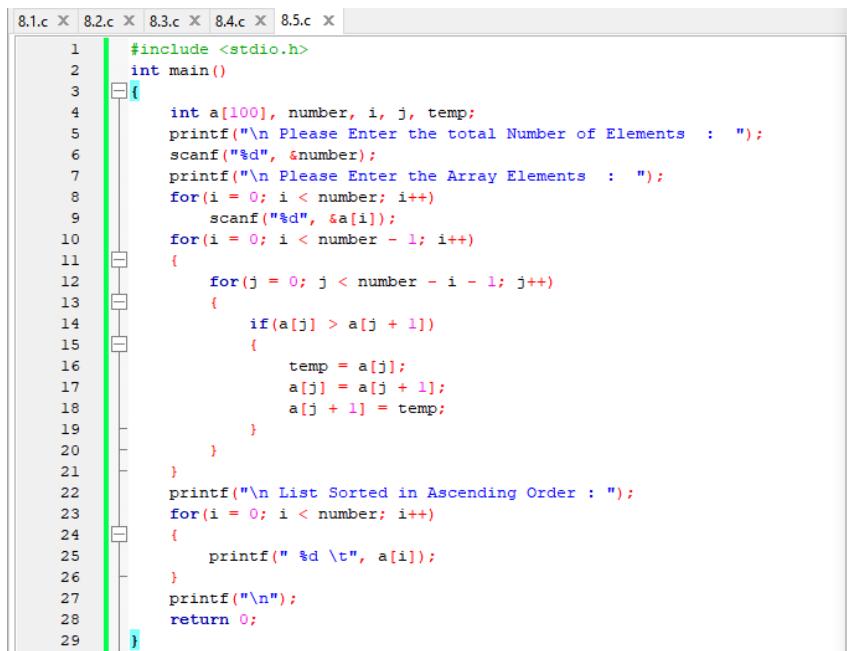
Step 2 : Initialize $a[100]$, number, i, j, temp, get the values of number and $a[i]$ from the user.

Step 3 : for $i=0$, $i < \text{number}$ and $i++$, and for $(i=0; i < \text{number}-1)$, $i++$, for $j=0$, $j < \text{number}-i-1$, $j++$, check if $a[j] > a[j+1]$ then swap $\text{temp} = a[j], a[j+1] \text{ and } a[j+1] = \text{temp}$.

Step 4 : After Step 3 print $a[i]$ using for loop in ascending order

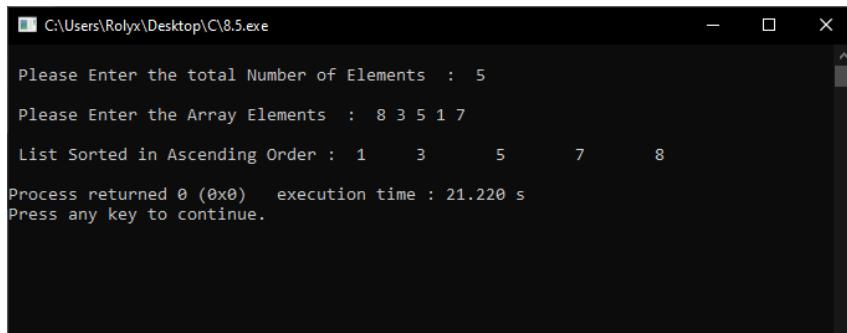
Step 5 : Stop

SOURCE CODE :



```
8.1.c X 8.2.c X 8.3.c X 8.4.c X 8.5.c X
1 #include <stdio.h>
2 int main()
3 {
4     int a[100], number, i, j, temp;
5     printf("\n Please Enter the total Number of Elements : ");
6     scanf("%d", &number);
7     printf("\n Please Enter the Array Elements : ");
8     for(i = 0; i < number; i++)
9         scanf("%d", &a[i]);
10    for(i = 0; i < number - 1; i++)
11    {
12        for(j = 0; j < number - i - 1; j++)
13        {
14            if(a[j] > a[j + 1])
15            {
16                temp = a[j];
17                a[j] = a[j + 1];
18                a[j + 1] = temp;
19            }
20        }
21    }
22    printf("\n List Sorted in Ascending Order : ");
23    for(i = 0; i < number; i++)
24    {
25        printf(" %d \t", a[i]);
26    }
27    printf("\n");
28    return 0;
29 }
```

OUTPUT :



```
C:\Users\Roly\Desktop\C\8.5.exe
Please Enter the total Number of Elements : 5
Please Enter the Array Elements : 8 3 5 1 7
List Sorted in Ascending Order : 1      3      5      7      8
Process returned 0 (0x0)  execution time : 21.220 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

2-D Arrays

EXERCISE 9.1

EX-No. 9	2 - D ARRAYS
DATE: 24/05/21	
1.	write a program to perform matrix addition and matrix subtraction.

AIM :
To perform matrix addition and matrix subtraction

ALGORITHM :

Step 1 : Start

Step 2 : Declare a $[10][10]$, $b[10][10]$, $c[10][10]$, $d[10][10]$, row col, i, j ,

Step 3 : get the no. of rows and columns from the user.

Step 4 : For $i=0, i < \text{row}$ and $i++$ and for $j=0, j < \text{col}$ and $j++$, get the values of $a[i][j]$ from the user for matrix A.

Step 5 : For $i=0, i < \text{row}$ and $i++$ and for $j=0, j < \text{col}$ and $j++$, get the values of $b[i][j]$ from the user for matrix B

Step 6 : For $i=0, i < \text{row}$ and $i++$ and for $j=0, j < \text{col}$, $j++$, Compute $c[i][j] = a[i][j] + b[i][j]$ and Perform addition.

Step 7 : For $i=0, i < \text{row}, i++$ and for $j=0, j < \text{col}$
 $j++$ print the addition matrix

Step 8 : For $i=0, i < \text{row}$ and $i++$ and
 $j=0, j < \text{col}, j++$ compute $c_{ij}(i) = a_{ij} - b_{ij}c_{ij}$

Step 9 : For $i=0, i < \text{row}$ and $i++$ and for
 $j=0, j < \text{col}$ and $j++$ display the
subtraction matrix

Step 10 : Stop

SOURCE CODE :

The screenshot shows the Code::Blocks IDE interface with the file "matrix addition and subtraction.c" open. The code implements matrix addition and subtraction for two 3x3 matrices. It prompts the user for the number of rows and columns, then for the elements of both matrices A and B. It then calculates the sum (A+B) and difference (A-B) and prints them out.

```
#include<stdio.h>
int main()
{
    int a[10][10],b[10][10],c[10][10],d[10][10], row, col,i,j;
    printf("Enter the rows number of both matrices\n");
    scanf("%d",&row);
    printf("Enter the columns number of both matrices\n");
    scanf("%d",&col);
    printf("Enter the elements of matrix A\n");
    for(i=0;i<row;i++){
        for(j = 0;j < col;j++){
            printf("Enter the element at [%d%d] : ",i,j);
            scanf("%d",&a[i][j]);
        }
        printf("\n");
    }
    printf("Enter the elements of matrix B\n");
    for(i=0;i<row;i++){
        for(j = 0;j < col;j++){
            printf("Enter the element at [%d%d] : ",i,j);
            scanf("%d",&b[i][j]);
        }
        printf("\n");
    }
    printf("The addition of A and B is: \n");
    for(i=0;i<row;i++){
        for(j = 0;j < col;j++){
            c[i][j]=a[i][j] + b[i][j];
        }
    }
    for(i=0;i<row;i++){
        for(j = 0;j < col;j++){
            printf(" %d ",c[i][j]);
        }
        printf("\n");
    }
    printf("\nThe subtraction of A and B is: \n");
    for(i=0;i<row;i++){
        for(j = 0;j < col;j++){
            d[i][j]=a[i][j] - b[i][j];
        }
    }
    for(i=0;i<row;i++){
        for(j = 0;j < col;j++){
            printf(" %d ",d[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT :

The screenshot shows a terminal window titled "C:\Users\Rolyx\Downloads\matrix addition and subtraction.exe". The program prompts the user for the dimensions of the matrices (3x3), then asks for the elements of both matrices A and B. It then calculates and prints the sum (A+B) and difference (A-B) of the matrices.

```
C:\Users\Rolyx\Downloads\matrix addition and subtraction.exe
Enter the rows number of both matrices
3 3
Enter the columns number of both matrices
Enter the elements of matrix A
Enter the element at [00] : 1
Enter the element at [01] : 2
Enter the element at [02] : 3
Enter the element at [10] : 4
Enter the element at [11] : 5
Enter the element at [12] : 6
Enter the element at [20] : 7
Enter the element at [21] : 8
Enter the element at [22] : 9
Enter the elements of matrix B
Enter the element at [00] : 12
Enter the element at [01] : 1
Enter the element at [02] : 3
Enter the element at [10] : 5
Enter the element at [11] : 6
Enter the element at [12] : 7
Enter the element at [20] : 8
Enter the element at [21] : 9
Enter the element at [22] : 0
The addition of A and B is:
    13  3  6
    9  11  13
   15  17  9
The subtraction of A and B is:
   -11  1  0
   -1  -1  -1
   -1  -1  9
Process returned 0 (0x0)  execution time : 192.734 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 9.2

2. write a program to perform matrix multiplication by checking the compatibility

AIM:

To perform matrix multiplication
by checking the compatibility

ALGORITHM:

Step 1: Start

Step 2: Declare $a[10][10]$, $b[10][10]$, result
 $[10][10]$, $r_1, c_1, r_2, c_2, i, j, k$.

Step 3: Get the values of r_1, r_2, c_1 and c_2
from the user

Step 4: while $c_1 \neq r_2$ print "Error! Column
of first matrix not equal to row
of second" and go to step 5

Step 5: Get the values of r_1, c_1, r_2, c_2 from
the user for ($i=0, j < r_1$ and $+i$) and
for ($i=0, j < c_1$, $+j$) get the values
of $a[i][j]$ and for $i=0, i < r_2, +i$
and $j=0, j < c_2$ and $+j$ get the
values of $b[i][j]$

Step 6 : For $i=0$, $i < r_1$, and $+ti$, for $j=0$, $j < c_2$
and $+tj$ and for $k=0$, $k < c_1$ and $+tk$
display result $a[i][j] = a[i][k]^*$
 $b[k][j]$

Step 7 : Stop

SOURCE CODE :

The screenshot shows the Code::Blocks IDE interface with the file "matrix multiplication.c" open. The code implements matrix multiplication of two 3x3 matrices. It prompts the user for input rows and columns, then for elements of both matrices, and finally prints the resulting matrix. The code uses nested loops for matrix multiplication and includes error handling for unequal column counts.

```
#include <stdio.h>
int main()
{
    int a[10][10], b[10][10], result[10][10], r1, c1, r2, c2, i, j, k;
    printf("Enter rows and column for first matrix: ");
    scanf("%d %d", &r1, &c1);
    printf("Enter rows and column for second matrix: ");
    scanf("%d %d", &r2, &c2);
    if(c1 != r2)
    {
        printf("Error! column of first matrix not equal to row of second.\n\n");
        return 0;
    }
    for(i=0; i<r1; ++i)
        for(j=0; j<c1; ++j)
    {
        printf("Enter elements a%d%d: ", i+1, j+1);
        scanf("%d", &a[i][j]);
    }
    for(i=0; i<r2; ++i)
        for(j=0; j<c2; ++j)
    {
        printf("Enter elements b%d%d: ", i+1, j+1);
        scanf("%d", &b[i][j]);
    }
    for(i=0; i<r1; ++i)
        for(j=0; j<c2; ++j)
    {
        result[i][j] = 0;
        for(k=0; k<c1; ++k)
        {
            result[i][j] += a[i][k]*b[k][j];
        }
    }
    printf("\nOutput Matrix:\n");
    for(i=0; i<r1; ++i)
        for(j=0; j<c2; ++j)
    {
        printf("%d ", result[i][j]);
        if(j == c2-1)
            printf("\n");
    }
    return 0;
}
```

OUTPUT:

The screenshot shows a terminal window running the program "matrix multiplication.exe". The user inputs 3 3 for both matrices. The program then prompts for elements of both matrices. The output shows the resulting 3x3 matrix:

```
C:\Users\Rolyx\Downloads\matrix multiplication.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 41, Col 20, Pos 955 Insert Read/Write default
C:\Users\Rolyx\Downloads\matrix multiplication.exe
Enter rows and column for first matrix: 3 3
Enter rows and column for second matrix: 3 3

Enter elements of matrix 1:
Enter elements a11: 1
Enter elements a12: 4
Enter elements a13: 7
Enter elements a21: 5
Enter elements a22: 8
Enter elements a23: 3
Enter elements a31: 6
Enter elements a32: 9
Enter elements a33: 4

Enter elements of matrix 2:
Enter elements b11: 5
Enter elements b12: 21
Enter elements b13: 7
Enter elements b21: 8
Enter elements b22: 12
Enter elements b23: 32
Enter elements b31: 21
Enter elements b32: 0
Enter elements b33: 9

Output Matrix:
184 69 198
152 201 318
186 234 366

Process returned 0 (0x0) execution time : 59.070 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 9.3

3. write a program to Print the transpose of a matrix

AIM :

To print transpose of a matrix

ALGORITHM :

Step 1 : Start

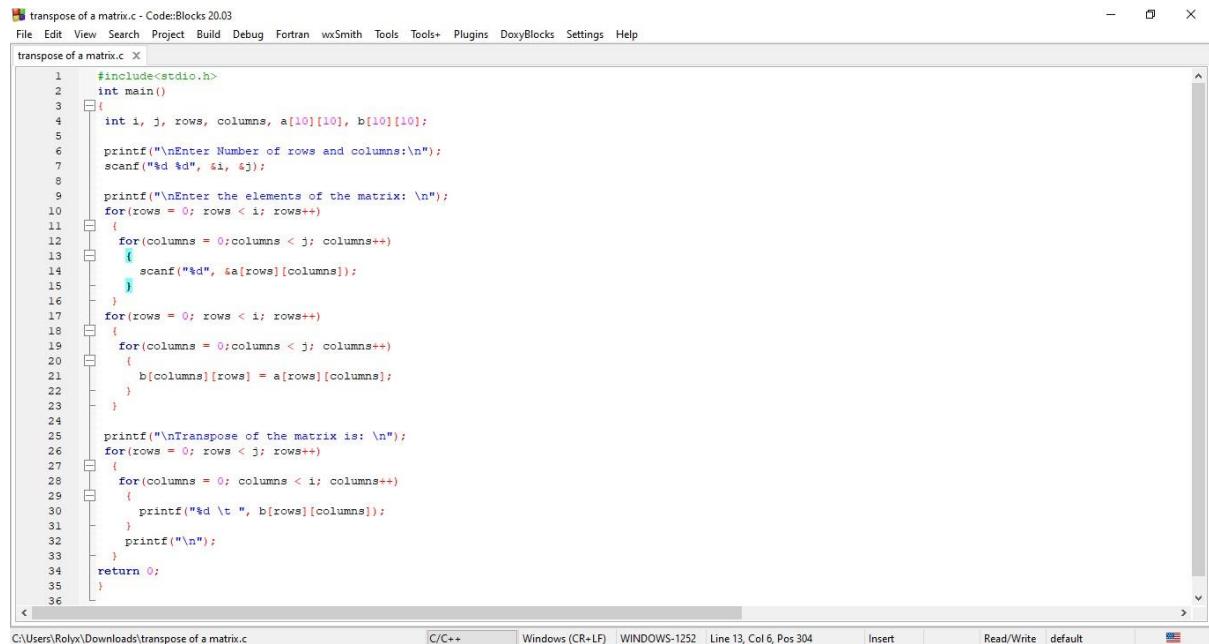
Step 2 : Declare i, j, rows, columns, a [10][10], b [10][10] get the values of i and j from the user

Step 3 : For rows = 0, rows < i and rows and for columns = 0, columns < j and columns ++ Get the values of rows and columns from the user.

Step 4 : After getting the values from the user compute and display the transpose matrix

Step 5 : Stop

SOURCE CODE :



The screenshot shows the Code::Blocks IDE interface with the file "transpose of a matrix.c" open. The code implements a transpose operation for a 10x10 matrix. It first prompts the user for the number of rows and columns, then for the matrix elements. Finally, it prints the transpose of the matrix.

```
#include<stdio.h>
int main()
{
    int i, j, rows, columns, a[10][10], b[10][10];
    printf("\nEnter Number of rows and columns:\n");
    scanf("%d %d", &i, &j);
    printf("\nEnter the elements of the matrix: \n");
    for(rows = 0; rows < i; rows++)
    {
        for(columns = 0;columns < j; columns++)
        {
            scanf("%d", &a[rows][columns]);
        }
    }
    for(rows = 0; rows < i; rows++)
    {
        for(columns = 0;columns < j; columns++)
        {
            b[columns][rows] = a[rows][columns];
        }
    }
    printf("\nTranspose of the matrix is: \n");
    for(rows = 0; rows < j; rows++)
    {
        for(columns = 0; columns < i; columns++)
        {
            printf("%d \t ", b[rows][columns]);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT :



The terminal window shows the execution of the program. The user enters "3 3" as the dimensions. Then, they enter the matrix elements row by row: 9 5 1, 8 1 2, and 6 3 4. The program then outputs the transpose of the matrix:

```
"C:\Users\RoLyx\Downloads\transpose of a matrix.exe"
Enter Number of rows and columns:
3 3
Enter the elements of the matrix:
9 5 1
8 1 2
6 3 4
Transpose of the matrix is:
9      8      6
5      1      3
1      2      4
```

RESULT:

The Desired Output has been successfully Verified and Executed.

Strings

EXERCISE 10.1

Ex. No. 10

STRINGS

DATE: 07/06/21

1. write a program to perform various string manipulations using built-in functions

AIM:

To Perform various string manipulation using built in functions

ALGORITHM :

Step 1: Start

Step 2: Initialize two strings str 1 and 2 and get their values from the user

Step 3: Then compute the following built-in functions `strcmp(str1, str2)`, `strlen(str1)`, `strlen(str2)`, `strlen(str2)`, `strcmp(str2, str1)`, `strcat(str1, str2)`, `strrev(str1)`, `strrev(str2)`, `strcpy(str2, str1)`

Step 4 : Print the output for above string functions

Step 5 : Stop

SOURCE CODE :

```
Ex10.3.c X *Ex10.3 new.c X Ex10.1.c X
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char str1[20], str2[20], str3[20];
6     printf("Enter the string 1:"); scanf("%s", &str1);
7     printf("Enter the string 2:"); scanf("%s", &str2);
8
9     printf("\nString in UPPERCASE: %s", strupr(str1));
10    printf("\nString in lowercase: %s\n", strlwr(str1));
11
12    printf("\nLength of string 1: %d\n", strlen(str1));
13    printf("Length of string 2: %d\n\n", strlen(str2));
14
15    if (strcmp(str1, str2) == 0){
16        printf("String 1 and string 2 are equal\n");
17    }else{
18        printf("String 1 and 2 are different\n");
19    }
20    printf("\nConcatenation of two strings : %s", strcat(str1, str2));
21    printf("\n\nReverse of string 1: %s\n", strrev(str1));
22    printf("Reverse of string 2: %s\n", strrev(str2));
23    printf("\n\nThe Copied string form string 1: %s", strcpy(str3, str1));
24    printf("\nString copy of string 1 and string 2 is: %s\n\n", strcpy(str3, str2));
25
26    return 0;
27 }
```

OUTPUT :

```
C:\Users\RoLyX\Desktop\Ex10.1.exe
Enter the string 1:thomas
Enter the string 2:shelby

String in UPPERCASE: THOMAS
String in lowercase: thomas

Length of string 1: 6
Length of string 2: 6

String 1 and 2 are different

Concatenation of two strings : thomasshelby

Reverse of string 1: yblehssamoht
Reverse of string 2: yblehs

The Copied string form string 1: yblehssamoht
String copy of string 1 and string 2 is: yblehs

Process returned 0 (0x0)   execution time : 17.212 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 10.2

2. write a program to print the given strings in ascending order

AIM:

To print the given string in ascending order

ALGORITHM :

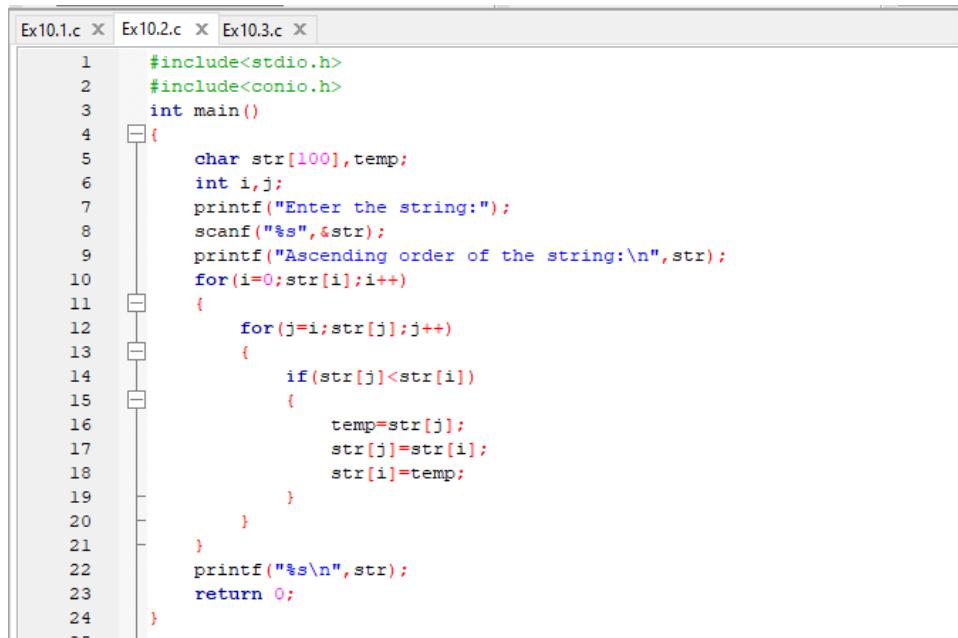
Step 1 : Start

Step 2 : Initialize 'i', str [00], temp and get the string from the user

Step 3 : FOR i=0, str [i] and i++ and for
[$i = i$, str [i] and $i + 1$] and if str [i]
 $<$ str [i], temp = str [i] swap str [i]-
str [i], str [i] = temp Print the
ascending order of the string

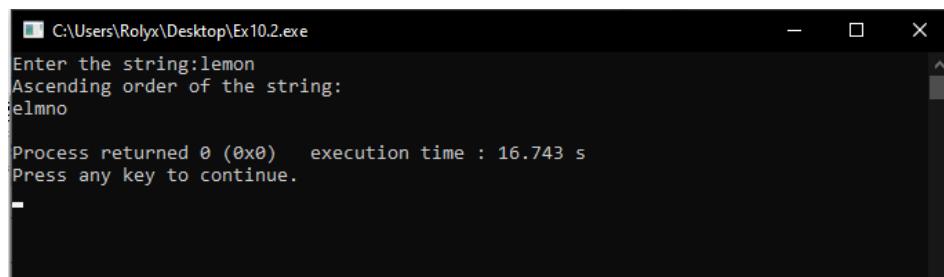
Step 4 : Stop

SOURCE CODE :



```
Ex10.1.c Ex10.2.c Ex10.3.c
1 #include<stdio.h>
2 #include<conio.h>
3 int main()
4 {
5     char str[100],temp;
6     int i,j;
7     printf("Enter the string:");
8     scanf("%s",&str);
9     printf("Ascending order of the string:\n",str);
10    for(i=0;str[i];i++)
11    {
12        for(j=i;str[j];j++)
13        {
14            if(str[j]<str[i])
15            {
16                temp=str[j];
17                str[j]=str[i];
18                str[i]=temp;
19            }
20        }
21    }
22    printf("%s\n",str);
23    return 0;
24 }
```

OUTPUT :



```
C:\Users\Rolyx\Desktop\Ex10.2.exe
Enter the string:lemon
Ascending order of the string:
elmono

Process returned 0 (0x0)   execution time : 16.743 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 10.3

3. write a program to verify the given strings is Palindrome or not (without built-in functions, with using built-in functions)

AIM:

To verify whether the given String is Palindrome or not using with and without built-in functions

ALGORITHM:

Step 1 : Start

Step 2 : Declare string (40), a[100], b[100],
length = 0, flag = 1, i

Step 3 : Get the string from the user, use for
loop for $i=0$, $\text{string}[i] \neq ' \backslash 0 '$, $i++$, length
 $++$
for $(i=0, i < \text{length}/2, i++)$

Step 4 : if $\text{string}[i] \neq \text{string}[\text{length}-1-i]$, flag = 0
and break

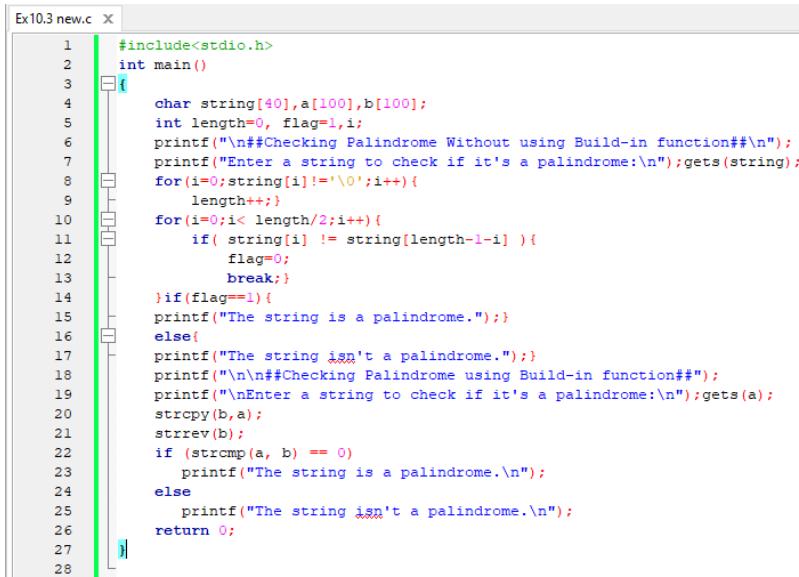
Step 5 : Using built in function get string from
user. Use `strcpy(b, a), strrev(b)`.

Step 6 : if `strcmp(a, b) == 0` print "The String
is Palindrome"

Step 7 : Else print "The String isn't a Palindrome".

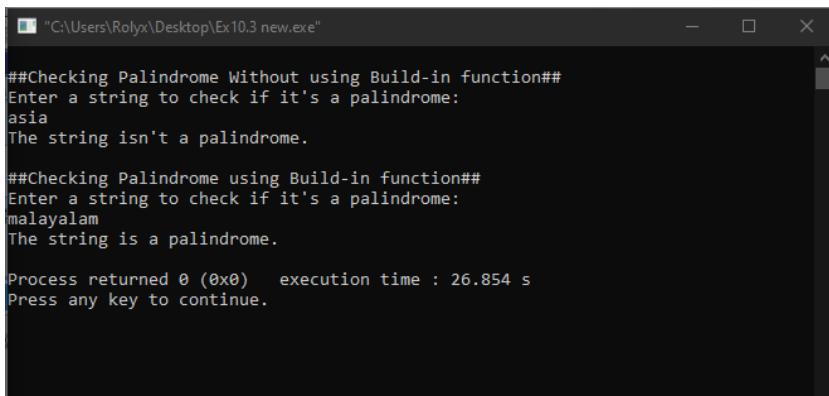
Step 8 : Stop

SOURCE CODE :



```
Ex10.3 new.c X
1 #include<stdio.h>
2 int main()
3 {
4     char string[40],a[100],b[100];
5     int length=0, flag=1,i;
6     printf("\n##Checking Palindrome Without using Build-in function##\n");
7     printf("Enter a string to check if it's a palindrome:\n");gets(string);
8     for(i=0;string[i]!='\0';i++)
9         length++;
10    for(i=0;i< length/2;i++){
11        if( string[i] != string[length-1-i] )
12            flag=0;
13            break;
14    }if(flag==1){
15        printf("The string is a palindrome.");
16    }else{
17        printf("The string isn't a palindrome.");
18    }
19    printf("\n##Checking Palindrome using Build-in function##");
20    printf("\nEnter a string to check if it's a palindrome:\n");gets(a);
21    strcpy(b,a);
22    strrev(b);
23    if (strcmp(a, b) == 0)
24        printf("The string is a palindrome.\n");
25    else
26        printf("The string isn't a palindrome.\n");
27    return 0;
28 }
```

OUTPUT :



```
C:\Users\Rolyx\Desktop\Ex10.3 new.exe

##Checking Palindrome Without using Build-in function##
Enter a string to check if it's a palindrome:
asia
The string isn't a palindrome.

##Checking Palindrome using Build-in function##
Enter a string to check if it's a palindrome:
malayalam
The string is a palindrome.

Process returned 0 (0x0)   execution time : 26.854 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 10.4

4. write a program to Concatenate two Strings using arrays

AIM :

To concatenate two strings using arrays

ALGORITHM :

Step 1 : Start

Step 2 : Initialize c [0], d[0], i, j

Step 3 : Get two strings from the user.

Step 4 : Declare i=0, while c[i] < 0, ++i

Step 5 : for j=0, d[i]>0, ++j, ++i, c[i]=d[j]
print, the string after concatenation

Step 6 : Stop

SOURCE CODE :

```
Ex10.1.c Ex10.2.c Ex10.3.c Ex 10.4.c

1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     char c[10],d[10];
6     int i,j;
7     printf("CONCATENATION OF TWO STRINGS USING ARRAY\n\n");
8     printf("Enter the string 1:");
9     scanf("%s",&c);
10    printf("Enter the string 2:");
11    scanf("%s",&d);
12
13    i=0;
14    while(c[i]>0)
15    {
16        ++i;
17    }
18    for(j=0;d[j]>0;++j,++i)
19    {
20        c[i]=d[j];
21    }
22    printf("After concatenation:%s\n",c);
23    return 0;
24
25 }
```

OUTPUT :

```
"C:\Users\Rolyx\Desktop\Ex 10.4.exe"
CONCATENATION OF TWO STRINGS USING ARRAY
Enter the string 1:snap
Enter the string 2:chat
After concatenation:snapchat

Process returned 0 (0x0)   execution time : 18.716 s
Press any key to continue.
```

RESULT:

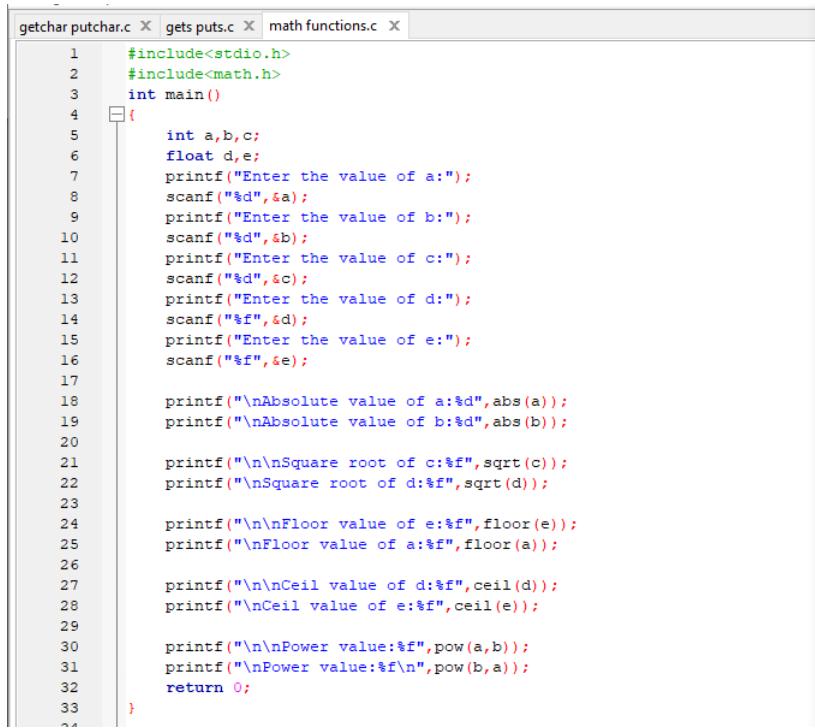
The Desired Output has been successfully Verified and Executed.

Math Functions and I/O Functions

EXERCISE 11.1

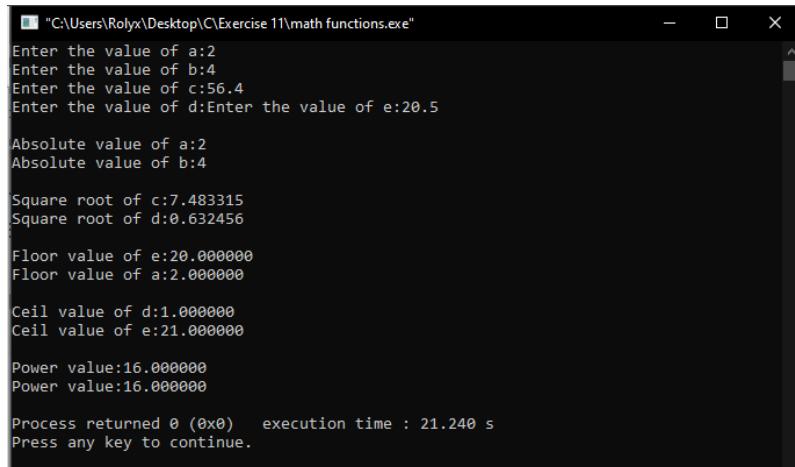
EX. NO 11	MATH FUNCTIONS AND I/O FUNCTIONS
DATE: 07/06/21	
1.	<p>write a program to read values from keyboard and find the values using <code>abs()</code>, <code>sqr()</code>, <code>floor()</code>, <code>ceil()</code> and <code>pow()</code>.</p> <p>AIM :</p> <p>To find values using <code>abs()</code>, <code>sqr()</code>, <code>floor()</code>, <code>ceil()</code>, <code>pow()</code></p> <p>ALGORITHM :</p> <p>Step 1 : Start</p> <p>Step 2 : Initialize <code>a,b,c,d,e</code> and get the values from the user.</p> <p>Step 3 : Compute given condition by <code>abs(a)</code>, <code>abs(c)</code>, <code>sqr(d)</code>, <code>sqr(c)</code>, <code>floor(d)</code>, <code>floor(e)</code>, <code>ceil(d)</code>, <code>ceil(e)</code>, <code>pow(a,b)</code>, <code>pow(b,a)</code></p> <p>Step 4 : print the result</p> <p>Step 5 : Stop</p>

SOURCE CODE :



```
getchar putchar.c X gets puts.c X math functions.c X
1 #include<stdio.h>
2 #include<math.h>
3 int main()
4 {
5     int a,b,c;
6     float d,e;
7     printf("Enter the value of a:");
8     scanf("%d",&a);
9     printf("Enter the value of b:");
10    scanf("%d",&b);
11    printf("Enter the value of c:");
12    scanf("%d",&c);
13    printf("Enter the value of d:");
14    scanf("%f",&d);
15    printf("Enter the value of e:");
16    scanf("%f",&e);
17
18    printf("\nAbsolute value of a:%d",abs(a));
19    printf("\nAbsolute value of b:%d",abs(b));
20
21    printf("\n\nSquare root of c:%f",sqrt(c));
22    printf("\nSquare root of d:%f",sqrt(d));
23
24    printf("\n\nFloor value of e:%f",floor(e));
25    printf("\nFloor value of a:%f",floor(a));
26
27    printf("\n\nCeil value of d:%f",ceil(d));
28    printf("\nCeil value of e:%f",ceil(e));
29
30    printf("\n\nPower value:%f",pow(a,b));
31    printf("\nPower value:%f\n",pow(b,a));
32
33    return 0;
34 }
```

OUTPUT :



```
C:\Users\RoLyx\Desktop\C\Exercise 11\math functions.exe"
Enter the value of a:2
Enter the value of b:4
Enter the value of c:56.4
Enter the value of d:Enter the value of e:20.5
Absolute value of a:2
Absolute value of b:4
Square root of c:7.483315
Square root of d:0.632456
Floor value of e:20.000000
Floor value of a:2.000000
Ceil value of d:1.000000
Ceil value of e:21.000000
Power value:16.000000
Power value:16.000000

Process returned 0 (0x0) execution time : 21.240 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 11.2

2. write a program to read and display a value using getch() and Putch()

AIM:

To read and display values using getch() and Putch()

ALGORITHM :

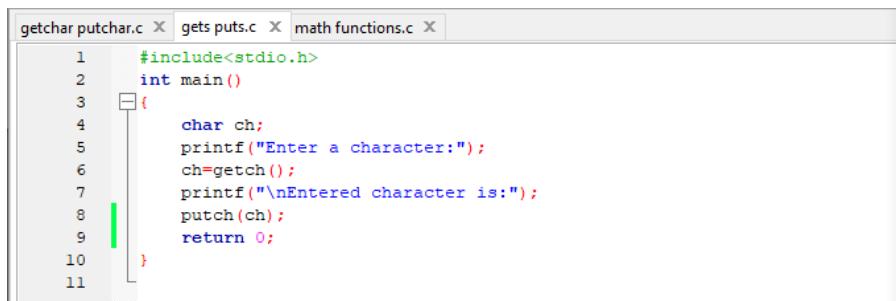
Step 1 : Start

Step 2 : Initialize ch and get the value of ch from the user

Step 3 : user using getch() function
Print the value of ch using Putch() function

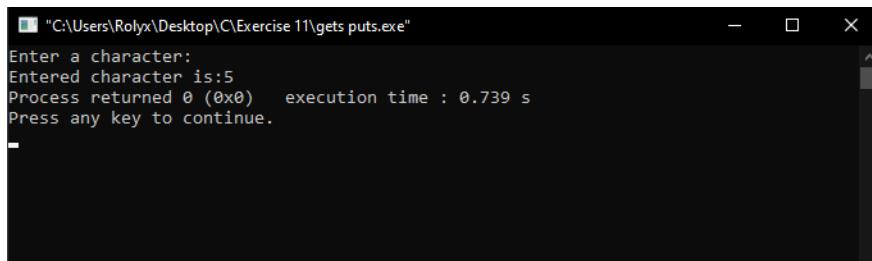
Step 4 : Stop

SOURCE CODE :



```
getchar putchar.c x gets puts.c x math functions.c x
1  #include<stdio.h>
2  int main()
3  {
4      char ch;
5      printf("Enter a character:");
6      ch=getch();
7      printf("\nEnterd character is:");
8      putch(ch);
9      return 0;
10 }
11 }
```

OUTPUT :



```
C:\Users\RoLyX\Desktop\C\Exercise 11\gets puts.exe
Enter a character:
Entered character is:5
Process returned 0 (0x0)   execution time : 0.739 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 11.3

3. write a program to read and display a value using getchar(), Putchar(), gets() and puts()

AIM: TO read and display values using getch() and Putch()

ALGORITHM :

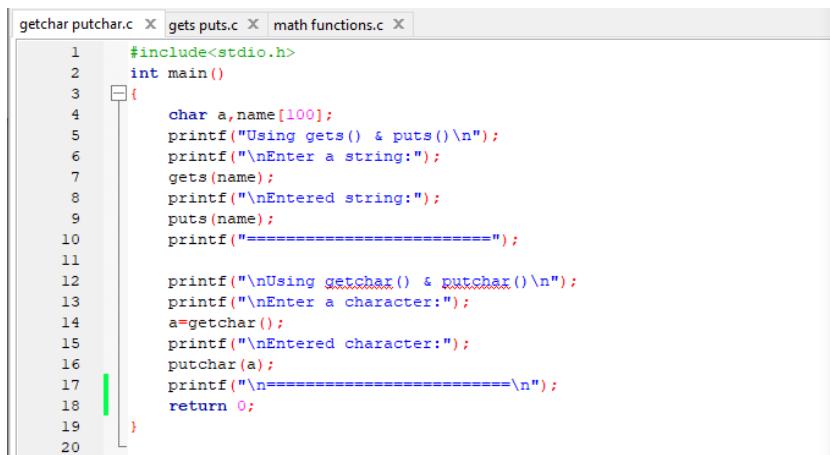
Step 1 : Start

Step 2 : Initialize ch, name [20], Get value of ch using get char() and name using gets from user

Step 3 : Print name using puts and ch using Putchar() functions.

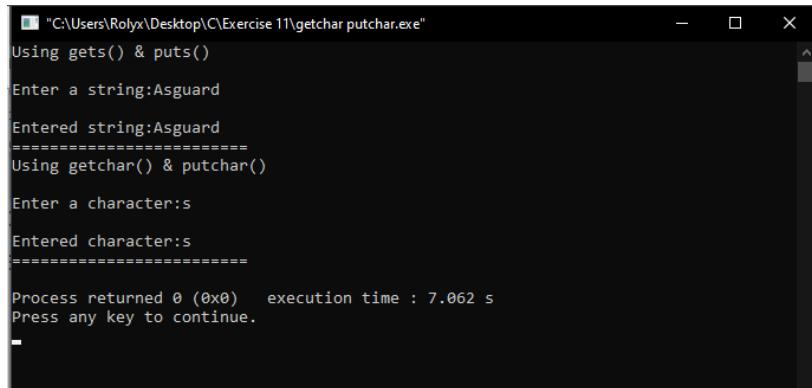
Step 4 : Stop

SOURCE CODE :



```
getchar_putchar.c  X  gets_puts.c  X  math functions.c  X
1  #include<stdio.h>
2  int main()
3  {
4      char a,name[100];
5      printf("Using gets() & puts()\n");
6      printf("\nEnter a string:");
7      gets(name);
8      printf("\nEntered string:");
9      puts(name);
10     printf("=====\n");
11
12     printf("\nUsing getchar() & putchar()\n");
13     printf("\nEnter a character:");
14     a=getchar();
15     printf("\nEnterd character:");
16     putchar(a);
17     printf("\n=====\n");
18
19 }
20
```

OUTPUT :



```
C:\Users\Rolyx\Desktop\C\Exercise 11\getchar_putchar.exe
Using gets() & puts()

Enter a string:Asguard
Entered string:Asguard
=====
Using getchar() & putchar()

Enter a character:s
Entered character:s
=====

Process returned 0 (0x0)   execution time : 7.062 s
Press any key to continue.
-
```

RESULT:

The Desired Output has been successfully Verified and Executed.

FUNCTIONS

EXERCISE 12.1

Exe. No 12

FUNCTIONS

DATE: 14/06/21

1. write a program to find sum of two numbers using functions.

AIM:

To find Sum of two number using functions

ALGORITHM :

Step 1 : Start

Step 2 : Initialize n_1, n_2 , add inside math function and get value n_1, n_2 from the user.

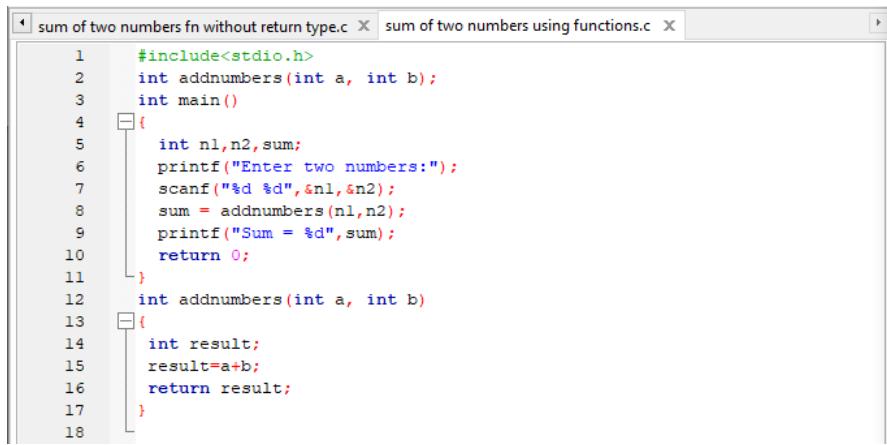
Step 3 : Declare a function sum and pass n_1 & n_2 as argument

Step 4 : compute $s = n_1 + n_2$, inside sum function & return s

Step 5 : Declare $add = \text{sum}(n_1, n_2)$ in main function then
Print $add = s$

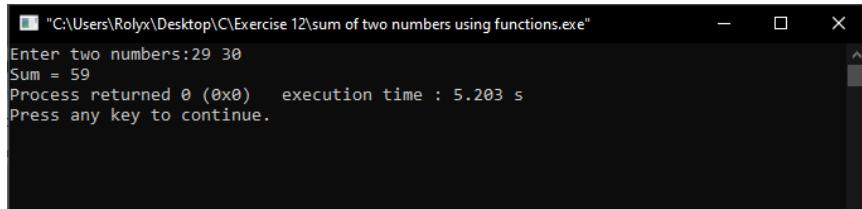
Step 6 : Stop

SOURCE CODE :



```
1 #include<stdio.h>
2 int addnumbers(int a, int b);
3 int main()
4 {
5     int n1,n2,sum;
6     printf("Enter two numbers:");
7     scanf("%d %d",&n1,&n2);
8     sum = addnumbers(n1,n2);
9     printf("Sum = %d",sum);
10    return 0;
11 }
12 int addnumbers(int a, int b)
13 {
14     int result;
15     result=a+b;
16     return result;
17 }
```

OUTPUT :



```
"C:\Users\RoLyx\Desktop\C\Exercise 12\sum of two numbers using functions.exe"
Enter two numbers:29 30
Sum = 59
Process returned 0 (0x0) execution time : 5.203 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 12.2

2. write a program to find product of two numbers using functions with out arguments and with out return type.

AIM:

To find product of two numbers using function with out argument and without return type.

ALGORITHM :

Step 1 : Start

Step 2 : Initialise Pdt function and call it inside main function without argument

Step 3 : Initialize a,b and get the value from the user

Step 4 : Compute $a * b$ and Print it

Step 5 : Stop

SOURCE CODE :

```
1 #include<stdio.h>
2 void pdt();
3 void main()
4 {
5     printf("\n To calculate the product of two numbers:");
6     pdt();
7 }
8 void pdt()
9 {
10    int a,b;
11    printf("\nEnter two numbers:");
12    scanf("%d %d",&a,&b);
13    printf("The product is %d",a*b);
14 }
15
```

OUTPUT :

```
To calculate the product of two numbers:
Enter two numbers:11 51
The product is 561
Process returned 18 (0x12)   execution time : 10.838 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 12.3

3. write a program to find difference of two numbers using functions with Out argument, with return type.

AIM :

To find difference of two numbers using function without argument, with return type.

ALGORITHM :

Step 1 : Start

Step 2 : Initialize difference () function and initialize result variables and call the difference () function using variable inside main function,

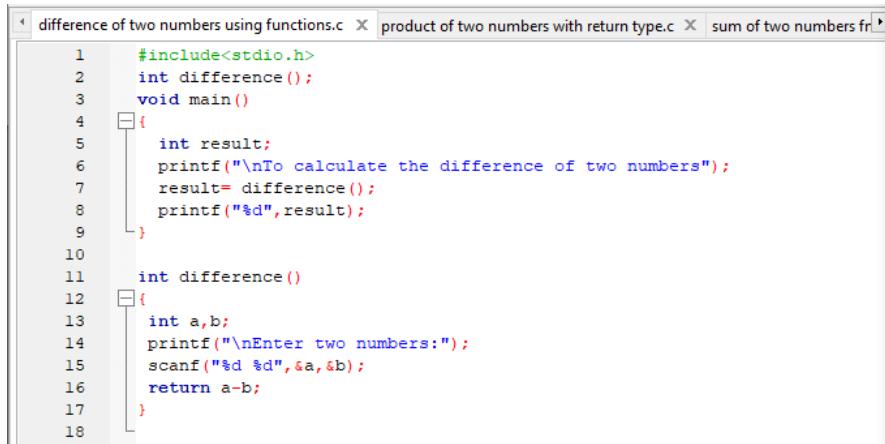
Step 3 : Inside difference function declare a,b & get the values from the user.

Step 4 : Compute a-b and return it

Step 5 : Print result inside main () which is returns Sub Value

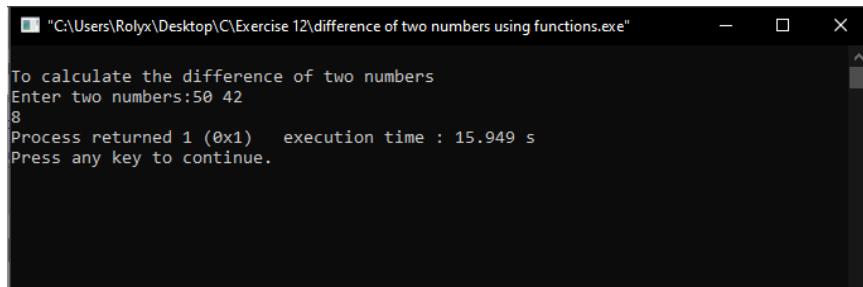
Step 6 : Stop

SOURCE CODE :



```
difference of two numbers using functions.c  product of two numbers with return type.c  sum of two numbers fr
1 #include<stdio.h>
2 int difference();
3 void main()
4 {
5     int result;
6     printf("\nTo calculate the difference of two numbers");
7     result= difference();
8     printf("%d",result);
9 }
10
11 int difference()
12 {
13     int a,b;
14     printf("\nEnter two numbers:");
15     scanf("%d %d",&a,&b);
16     return a-b;
17 }
18
```

OUTPUT :



```
C:\Users\Rolyx\Desktop\C\Exercise 12\difference of two numbers using functions.exe
To calculate the difference of two numbers
Enter two numbers:50 42
8
Process returned 1 (0x1)   execution time : 15.949 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 12.4

4. write a program to find sum of two numbers using functions with argument & without return type.

AIM :

To find sum of two numbers using with argument and without return type

ALGORITHM :

Step 1 : Start

Step 2 : Initialize sum() function

Step 3 : Get a,b from the user in main function

Step 4 : Call sum(a,b) function.

Step 5 : Compute a+b and print it

Step 6 : Stop

SOURCE CODE :

```
product of two numbers with return type.c x sum of two numbers fn without return type.c x
1 #include<stdio.h>
2 void sum();
3 void main()
4 {
5     int a,b,result;
6     printf("\n To calculate the sum of two numbers:");
7     printf("\nEnter two numbers:");
8     scanf ("%d %d",&a,&b);
9     sum(a,b);
10 }
11 void sum(int a,int b)
12 {
13     printf("\nThe sum is %d",a+b);
14 }
15
```

OUTPUT :

```
"C:\Users\Rolyx\Desktop\C\Exercise 12\sum of two numbers fn without return type.exe" - □ X
To calculate the sum of two numbers:
Enter two numbers:29 30
The sum is 59
Process returned 14 (0xE)   execution time : 4.061 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 12.5

5 write a program to find product of two numbers using functions with arguments with return type.

AIM :

To find Product of two numbers using function with argument with return type.

ALGORITHM :

Step 1 : Start

Step 2 : Declare Product ()

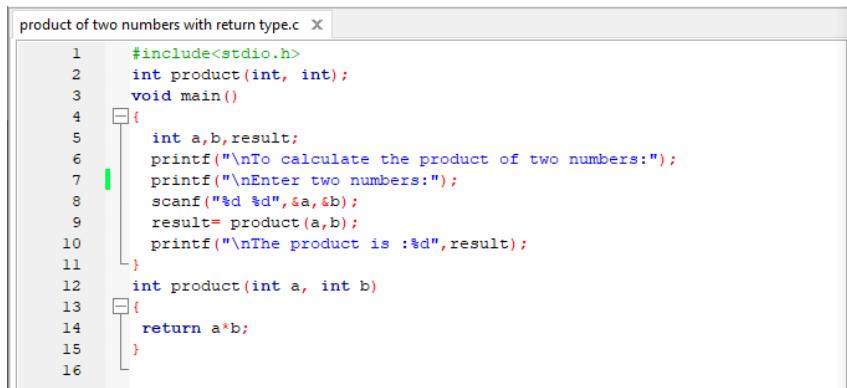
Step 3 : Initialize a, b, result and get a, b from the user in main ()

Step 4 : Call the product () by result= product (a,b) Compute $a * b$ in product () and return it.

Step 5 : Print $a * b$ in main () that returns $a * b$

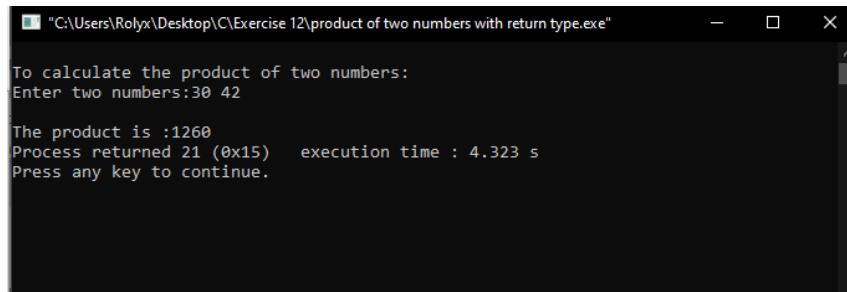
Step 6 : Stop

SOURCE CODE :



```
product of two numbers with return type.c
1 #include<stdio.h>
2 int product(int, int);
3 void main()
4 {
5     int a,b,result;
6     printf("\nTo calculate the product of two numbers:");
7     printf("\nEnter two numbers:");
8     scanf("%d %d",&a,&b);
9     result= product(a,b);
10    printf("\nThe product is :%d",result);
11 }
12 int product(int a, int b)
13 {
14     return a*b;
15 }
16
```

OUTPUT :



```
"C:\Users\RoLyx\Desktop\C\Exercise 12\product of two numbers with return type.exe"
To calculate the product of two numbers:
Enter two numbers:30 42
The product is :1260
Process returned 21 (0x15)   execution time : 4.323 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

FUNCTIONS AND RECURSIONS

EXERCISE 13.1

1. Write a program to swap two numbers using
 - a) Call By Value b) Call By Reference.

Ex. No. 13	FUNCTIONS AND RECURSIONS
DATE: 14/10/21	
1. write a program to swap two numbers using a) Call By value B) Call By Reference	
AIM:	
TO Swap two numbers using Call by value and Call by reference	
ALGORITHM FOR CALL BY VALUE:	
Step 1 : Start	
Step 2 : Declare the values of two int variables	
Step 3 : Then print the values before swapping in main function	
Step 4 : Then print the values after swapping in function	
Step 5 : Swap temp, temp=a, =a=b, b=temp	
Step 6 : Then Print the values after swapping in main function	
Step 7 : STOP	
ALGORITHM FOR CALL BY REFERENCE :-	
Step 1 : Start	
Step 2 : Declare the values of two int variables.	
Step 3 : Then Print values before	

Swapping in main function

Step 4 : Then print the values after
Swapping in Call by reference
function

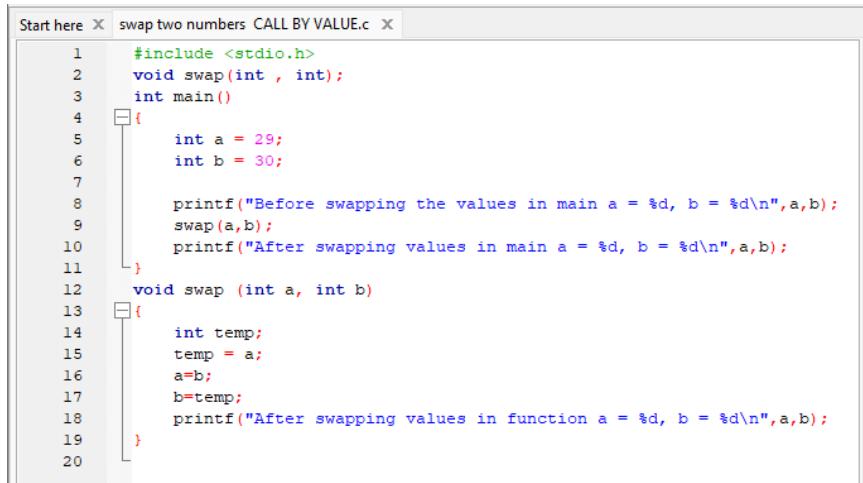
Step 5 : Then swap temp, $\text{temp} = ^*a, ^*a = b, ^*b = \text{temp}$

Step 6 : Then print the values after
Swapping in Call by reference
function.

Step 7 : Stop

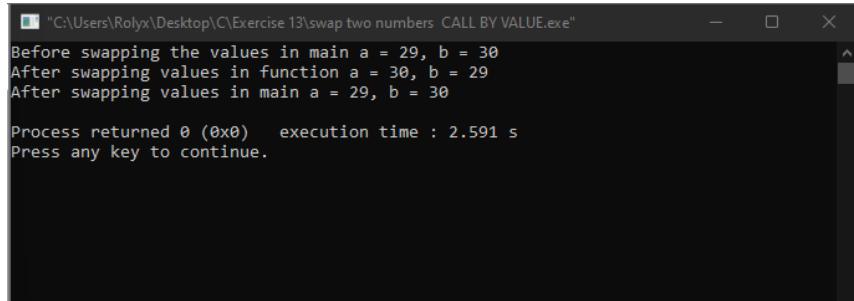
CALL BY VALUE

SOURCE CODE AND OUTPUT:



```
Start here X swap two numbers CALL BY VALUE.c X
1 #include <stdio.h>
2 void swap(int , int);
3 int main()
4 {
5     int a = 29;
6     int b = 30;
7
8     printf("Before swapping the values in main a = %d, b = %d\n",a,b);
9     swap(a,b);
10    printf("After swapping values in main a = %d, b = %d\n",a,b);
11 }
12 void swap (int a, int b)
13 {
14     int temp;
15     temp = a;
16     a=b;
17     b=temp;
18     printf("After swapping values in function a = %d, b = %d\n",a,b);
19 }
20
```

OUTPUT :

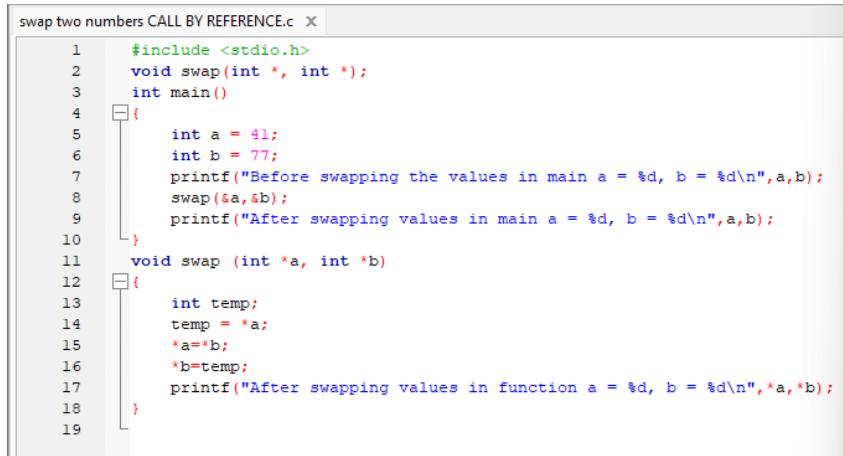


```
C:\Users\Rolyx\Desktop\C\Exercise 13\swap two numbers CALL BY VALUE.exe"
Before swapping the values in main a = 29, b = 30
After swapping values in function a = 30, b = 29
After swapping values in main a = 29, b = 30

Process returned 0 (0x0)   execution time : 2.591 s
Press any key to continue.
```

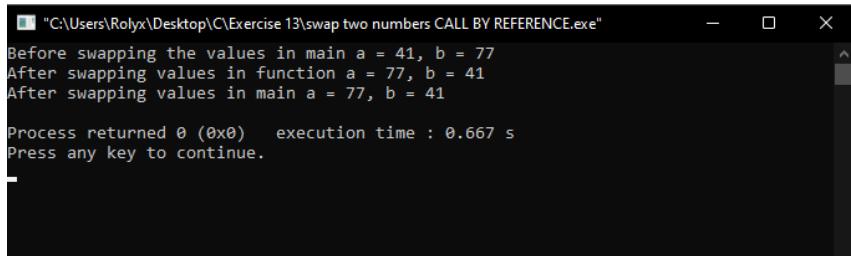
CALL BY REFERENCE

SOURCE CODE :



```
swap two numbers CALL BY REFERENCE.c X
1 #include <stdio.h>
2 void swap(int *, int *);
3 int main()
4 {
5     int a = 41;
6     int b = 77;
7     printf("Before swapping the values in main a = %d, b = %d\n", a, b);
8     swap(&a, &b);
9     printf("After swapping values in main a = %d, b = %d\n", a, b);
10 }
11 void swap (int *a, int *b)
12 {
13     int temp;
14     temp = *a;
15     *a=*b;
16     *b=temp;
17     printf("After swapping values in function a = %d, b = %d\n", *a, *b);
18 }
19
```

OUTPUT :



```
C:\Users\Rolyx\Desktop\C\Exercise 13\swap two numbers CALL BY REFERENCE.exe"
Before swapping the values in main a = 41, b = 77
After swapping values in function a = 77, b = 41
After swapping values in main a = 77, b = 41

Process returned 0 (0x0)   execution time : 0.667 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 13.2

2. write a program to calculate factorial gcd using recursion and non-recursion functions.

AIM:

To calculate factorial, gcd using recursion and non-recursion function

ALGORITHM:

Step 1 : Start

Step 2 : Initialize a,b,c,d,n and get the values from user.

Step 3 : Declare function rec gcd ()
with arguments a,b,c=rec
gcd (a,b),x,y, if (y==0) then
return x, else return (rec
gcd (y,x%y))

Step 4 : Declare function non rec gcd ()
with arguments a,b,...d non rec gcd
(a,b),x,y and initialize z, while x%y
!=0 then Swap z=x%y, x=y, y=z
and print the value.

SOURCE CODE :

```
#include<stdio.h>
int main()
{
    int a,b,c,d,n;
    printf("GCD\n");
    printf("Enter the first and second number:");
    scanf("%d %d",&a,&b);
    c=recgcd(a,b);
    printf("GCD of number using recursion:%d\n",c);
    d=nonrecgcd(a,b);
    printf("GCD of number using non recursion:%d\n",d);
    printf("\n");

    printf("FACTORIAL\n");
    printf("Enter the number:\n");
    scanf("%d",&n);
    a=refactorial(n);
    b=nonrefactorial(n);
    printf("Factorial of the number using recursion:%d\n",a);
    printf("Factorial of the number using non recursion:%d\n",b);
    getch();
}

int recgcd(int x,int y)
{
    if(y==0)
        return (x);
    else
        return(recgcd(y,x%y));
}

int nonrecgcd(int x,int y)
{
    int z;
    while(x % y != 0)
    {
        z = x % y;
        x = y;
        y = z;
    }
    return(y);
}

int refactorial(int x)
{
    int f;
    if(x == 0)
        return(1);
    else
        f = x * refactorial(x - 1);
    return(f);
}

int nonrefactorial(int x)
{
    int i, f = 1;
    for(i = 1;i <= x; i++)
    {
        f = f * i;
    }
    return(f);
}
```

OUTPUT :

```
C:\Users\RoLyX\Desktop\C\Exercise 13\program to calculate factorial, gcd using r...
GCD
Enter the first and second number:4 5
GCD of number using recursion:1
GCD of number using non recursion:1

FACTORIAL
Enter the number:
29 30
Factorial of the number using recursion:-1241513984
Factorial of the number using non recursion:-1241513984

Process returned 0 (0x0)   execution time : 55.003 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 13.3

3. write a program to perform arithmetic operation using pointer

AIM :

To Perform arithmetic operations using Pointers

ALGORITHM :

Step 1 : Start

Step 2 : Initialize num₁, num₂ and get its value from user, declare *P₁ = &num₁ and *P₂ = &num₂.

Step 3 : print "Increment" as P++ and "decrement" as P--

Step 4 : print "Addition" as (P=P+5) and "Subtraction" as (P=P-5)

Step 5 : FOR Comparison, check if P₂=P,
print "ptr₂ is not equal to ptr₁"
else print "not equal"

Step 6 : If (P₂>P₁), print "ptr₂ is greater than ptr₁," else print "ptr₂ is less than ptr₁,"

Step 7 : Stop

SOURCE CODE :

```
Start here X Arithmetic operators using pointers.c X
1 #include<stdio.h>
2 int main()
3 {
4     int num,a,b;
5     printf("INCREMENT\n");
6     printf("Enter the value:");
7     scanf("%d",&num);
8     int*p;
9     p=&num;
10    printf("\nThe address of p variable is %u\n",p);
11    p++;
12    printf("After Increment\n");
13    printf("The address of p variable is %u\n",p);
14
15    printf("\nDECREMENT\n");
16    printf("Enter the value:");
17    scanf("%d",&num);
18    int *q;
19    p=&num;
20    printf("The address of p variable is %u\n",p);
21    p--;
22    printf("After Decrement\n");
23    printf("The address of p variable is %u\n",p);
24
25    printf("\nADDITION\n");
26    printf("Enter the value:");
27    scanf("%d",&num);
28    p=&num;
29    printf("The address of p variable is %u\n",p);
30    p+=5;
31    printf("After Adding\n");
32    printf("The address of p variable is %u\n",p);
33
34
35    printf("\nSUBTRACTION\n");
36    printf("Enter the value:");
37    scanf("%d",&num);
38    int *s;
39    p=&num;
40    printf("The address of p variable is %u\n",p);
41    p=p-5;
42    printf("After Subtraction\n");
43    printf("The address of p variable is %u\n",p);
44
45    printf("\nCOMPARISON\n");
46    printf("Enter the value:");
47    scanf("%d",&num);
48    int*ptr1=&num;
49    int*ptr2=&num;
50    if(ptr1==ptr2)
51    {
52        printf("They are Equal");
53    }
54    else
55    {
56        printf("They are not equal");
57    }
58    return 0;
59 }
60
```

OUTPUT :

```
C:\Users\Rolyx\Desktop\C\Exercise 13\Arithmetic operators using pointers.exe
INCREMENT
Enter the value:4

The address of p variable is 6422020
After Increment
The address of p variable is 6422024

DECREMENT
Enter the value:4
The address of p variable is 6422020
After Decrement
The address of p variable is 6422016

ADDITION
Enter the value:5
The address of p variable is 6422020
After Adding
The address of p variable is 6422040

SUBTRACTION
Enter the value:5
The address of p variable is 6422020
After Subtraction
The address of p variable is 6422000

COMPARISON
Enter the value:4
They are Equal
Process returned 0 (0x0) execution time : 20.248 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 13.4

4. write a program matrix addition using Pointers

AIM : To perform matrix addition using Pointers

ALGORITHM :

Step 1 : Start

Step 2 : Initialize $i, j, m, n, A[m][n], B[m][n]$ and add $[m][n]$, get values of m, n from user

Step 3 : Get value $(*(A+i)*j)$ and $(*(B+i)*j)$ from user using for loop

Step 4 : Compute $*(\text{add } + i) + *j = *(A+i, *j) + *(B+i, *j)$ and print $*(\text{add } + i) + *j$

Step 5 : Stop

SOURCE CODE :

```
Matrix addition using pointers.c X
1  #include<stdio.h>
2  int main()
3  {
4      int a[10][10],b[10][10],c[10][10],i,j,m,n;
5      printf("Enter the value of m:");
6      scanf("%d",&m);
7      printf("Enter the value of n:");
8      scanf("%d",&n);
9      printf("Enter the first matrix:");
10     for(i=0;i<m;i++)
11     {
12         for(j=0;j<n;j++)
13         {
14             scanf("%d",*(a+i)+j));
15         }
16     }
17     printf("Enter the second matrix:");
18     for(i=0;i<m;i++)
19     {
20         for(j=0;j<n;j++)
21         {
22             scanf("%d",*(b+i)+j));
23         }
24     }
25     printf("Addition Matrix:\n");
26     for(i=0;i<m;i++)
27     {
28         for(j=0;j<n;j++)
29         {
30             *(c+i)+j)=*(a+i)+j)+*(b+i)+j);
31             printf("%d ",*(c+i)+j));
32         }
33     }
34 }
35 }
```

OUTPUT:

```
C:\Users\RoLyx\Desktop\C\Exercise 13\Matrix addition using pointers.exe"
Enter the value of m:2
Enter the value of n:2
Enter the first matrix:
4 2
6 7
Enter the second matrix:
5 7
8 3
Addition Matrix:
9 9
14 10

Process returned 0 (0x0) execution time : 39.689 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

STRUCTURES

EXERCISE 14.1

Ex. NO. 14	STRUCTURES
DATE: 28/06/21	

1. write a program to create structure for an account holder in a bank with following fields name, account number, address, balance and display the details of five

AIM :
To create structure of account holders for given fields

ALGORITHM :

Step 1 : Start

Step 2 : Create a structure called struct bank_details, and initialize account number, name, balance, address and use a variable to use it.

Step 3 : Initialize i in main function and using for loop get five values for each field using a[i] acc - no, a[i] name, a[i] balance and a[i] address from user

Step 4 : print the details

Step 5 : Stop

SOURCE CODE AND OUTPUT:

```
bank details structures.c X
1  #include<stdio.h>
2  struct bank_details
3  {
4      int account_number;
5      char name[10], address[10];
6      int balance;
7  }
8  a[5];
9  int main()
10 {
11     int i;
12     for(i=0;i<5;i++)
13     {
14         printf("Details of the Customer: %d\n",i+1);
15         printf("\nEnter the name:");
16         scanf("%s",&a[i].name);
17         printf("Enter the account number:");
18         scanf("%d",&a[i].account_number);
19         printf("Enter the balance:");
20         scanf("%d",&a[i].balance);
21         printf("Enter the address:");
22         scanf("%s",&a[i].address);
23         printf("\n=====\n");
24         printf("\n");
25     }
26     return 0;
27 }
```

OUTPUT :

```
C:\Users\RoLyx\Desktop\C\Exercise 14\bank details structures.exe
Details of the Customer: 1
Enter the name:Tharun
Enter the account number:6587390782
Enter the balance:89000
Enter the address:HBPalace
=====
Details of the Customer: 2
Enter the name:Gowtham
Enter the account number:6749838253
Enter the balance:67000
Enter the address:parkstreet
=====
Details of the Customer: 3
Enter the name:Kumar
Enter the account number:8729640371
Enter the balance:34000
Enter the address:shевапет
=====
Details of the Customer: 4
Enter the name:Ravi
Enter the account number:9130694842
Enter the balance:56000
Enter the address:HJGarden
=====
Details of the Customer: 5
Enter the name:Deva
Enter the account number:4835280765
Enter the balance:34600
Enter the address:vincent
=====

Process returned 0 (0x0) execution time : 176.267 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 14.2

2. write a program to find total marks of individual student and average marks for 10 students using structures.

AIM :

TO find total and average mark of 10 students using structures

ALGORITHM :

Step 1 : Start

Step 2 : Initialize Struct Student and initialize name [0], roll no, subject [5], total

Step 3 : Initialize static struct student and initialize s[100], n, i, j

Step 4 : get no. of students and five subject mark from the user

Step 5 : find total for each student

Step 6 : find average for 10 students using average = (float)a[0] / 10, and print the result

Step 7 : Stop

SOURCE CODE :

```
total and average mark of the students.c X
1 #include<stdio.h>
2 struct student
3 {
4     char name[10];
5     int rollno;
6     int subject[5],total;
7 };
8 int main ( )
9 {
10     static struct student s[100];
11     int n,i,j;
12     int atot=0;
13     printf("Enter the number of Students: ");
14     scanf("%d",&n);
15     printf("Enter the Marks of Five Subjects: ");
16     for(i=1; i<=n; i++)
17     {
18         printf("\nEnter student[%d] student marks:",i);
19         s[i].total=0;
20         for(j=0; j<5; j++)
21         {
22             scanf("%d",&s[i].subject[j]);
23             s[i].total=s[i].total+s[i].subject[j];
24         }
25         printf("\nTotal marks %d\n",s[i].total);
26         atot+=s[i].total;
27     }
28     float average;
29     average=(float)atot/10;
30     printf("AVERAGE MARK OF 10 STUDENTS :%.2f",average);
31
32 }
33
```

OUTPUT :

```
C:\Users\Roxy\Desktop\C\Exercise 1\total and average mark of the students.exe*
Enter the number of Students: 10
Enter the Marks of Five Subjects:
Enter student[1] student marks:38 89 56 64 72
Total marks 319
Enter student[2] student marks:67 78 65 49 76
Total marks 335
Enter student[3] student marks:77 88 66 56 64
Total marks 351
Enter student[4] student marks:65 78 67 87 78
Total marks 375
Enter student[5] student marks:67 78 76 85 89
Total marks 395
Enter student[6] student marks:76 56 67 89 74
Total marks 362
Enter student[7] student marks:65 78 76 58 87
Total marks 364
Enter student[8] student marks:77 88 99 92 88
Total marks 444
Enter student[9] student marks:85 89 92 93 87
Total marks 446
Enter student[10] student marks:67 78 54 55 80
Total marks 334
AVERAGE MARK OF 10 STUDENTS :372.50
Process returned 0 (0x0)   execution time : 285.415 s
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 14.3

3 write a program to create structure called traveler and members of structure are train no, coach no, seat no, source, destination, gender, age, name and departure date.

AIM:

To Create Structure called traveler and store the given structure members

ALGORITHM :

Step 1 : Start

Step 2 : Create struct traveller of train-no, coach-no, seat-no, age, date, source [64], destination [64], gender [64], char name [64]

Step 3 : Initialize n, struct traveller t[100] and get value of n from user

Step 4 : Using for loop, get n, values

of $t[i].train\text{-}no$, $t[i].$
 $coach\text{-}no$, $t[i].seat\text{-}no$, $t[i]$
 $source$, $t[i].destination$,
 $t[i].gender$, $t[i].age$) $t[i]$
 $name$, $t[i].data$ from user

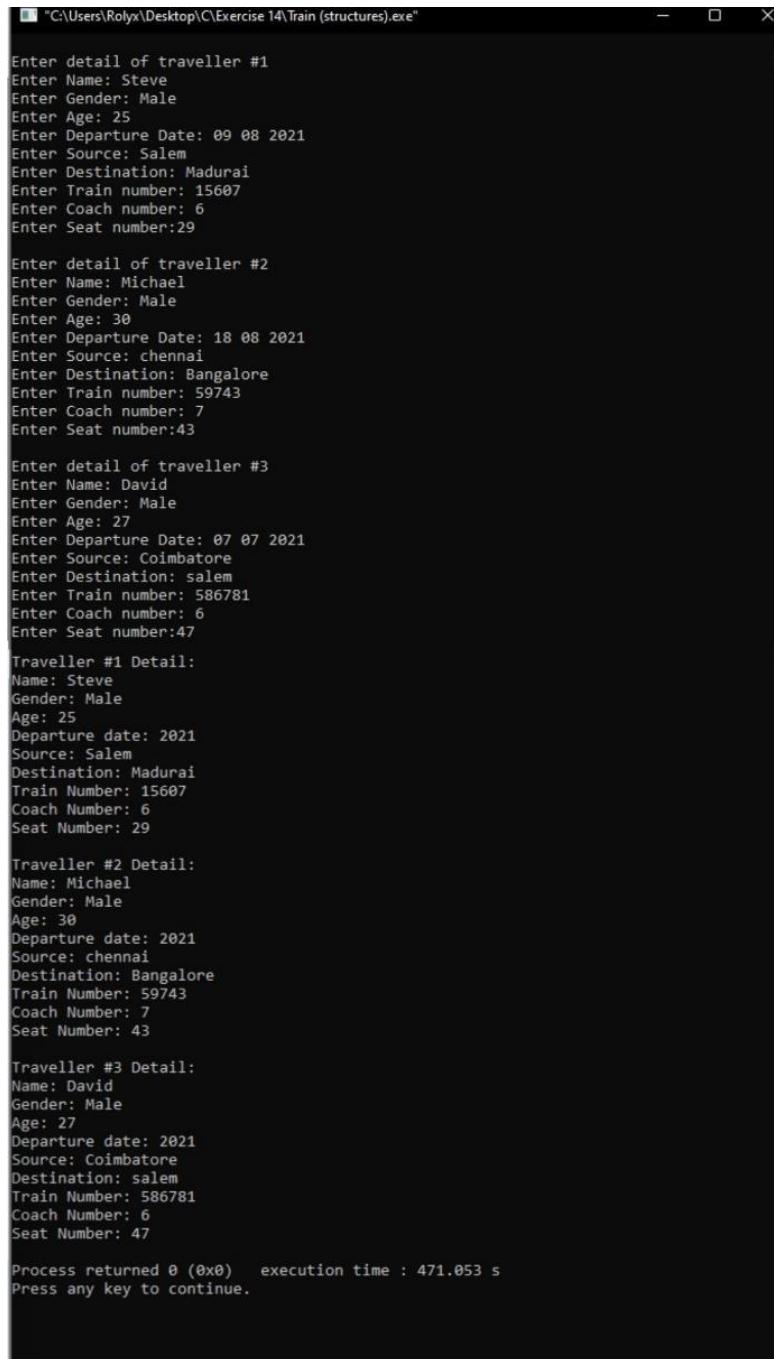
Step 5 : print values obtained
from step 4.

Step 6 : stop

SOURCE CODE :

```
Train (structures).c X
1  #include <stdio.h>
2  struct traveller{
3      int train_no;
4      int coach_no;
5      int seat_no;
6      int age;
7      int date;
8      char source[64];
9      char destination[64];
10     char gender[64];
11     char name[64];
12 }a[3];
13 int main() {
14     int i;
15     for (i = 0; i < 3; i++) {
16         printf("\nEnter detail of traveller #d\n", (i+1));
17
18         printf("Enter Name: ");
19         scanf("%s", a[i].name);
20
21         printf("Enter Gender: ");
22         scanf("%s", a[i].gender);
23
24         printf("Enter Age: ");
25         scanf("%d", &a[i].age);
26
27         printf("Enter Departure Date: ");
28         scanf("%d %d %d", &a[i].train_no,&a[i].coach_no,&a[i].date);
29
30         printf("Enter Source: ");
31         scanf("%s", a[i].source);
32
33         printf("Enter Destination: ");
34         scanf("%s", a[i].destination);
35
36         printf("Enter Train number: ");
37         scanf("%d", &a[i].train_no);
38
39         printf("Enter Coach number: ");
40         scanf("%d", &a[i].coach_no);
41
42         printf("Enter Seat number:");
43         scanf("%d", &a[i].seat_no);
44     }
45     for (i = 0; i < 3; i++) {
46         printf("\nTraveller #d Detail:\n", (i+1));
47         printf("Name: %s\n", a[i].name);
48         printf("Gender: %s\n", a[i].gender);
49         printf("Age: %d\n", a[i].age);
50         printf("Departure date: %d\n", a[i].date);
51         printf("Source: %s\n", a[i].source);
52         printf("Destination: %s\n", a[i].destination);
53         printf("Train Number: %d\n", a[i].train_no);
54         printf("Coach Number: %d\n", a[i].coach_no);
55         printf("Seat Number: %d\n", a[i].seat_no);
56     }
57     return 0;
58 }
```

OUTPUT :



```
"C:\Users\Rolyx\Desktop\C\Exercise 14\Train (structures).exe"

Enter detail of traveller #1
Enter Name: Steve
Enter Gender: Male
Enter Age: 25
Enter Departure Date: 09 08 2021
Enter Source: Salem
Enter Destination: Madurai
Enter Train number: 15607
Enter Coach number: 6
Enter Seat number:29

Enter detail of traveller #2
Enter Name: Michael
Enter Gender: Male
Enter Age: 30
Enter Departure Date: 18 08 2021
Enter Source: chennai
Enter Destination: Bangalore
Enter Train number: 59743
Enter Coach number: 7
Enter Seat number:43

Enter detail of traveller #3
Enter Name: David
Enter Gender: Male
Enter Age: 27
Enter Departure Date: 07 07 2021
Enter Source: Coimbatore
Enter Destination: salem
Enter Train number: 586781
Enter Coach number: 6
Enter Seat number:47

Traveller #1 Detail:
Name: Steve
Gender: Male
Age: 25
Departure date: 2021
Source: Salem
Destination: Madurai
Train Number: 15607
Coach Number: 6
Seat Number: 29

Traveller #2 Detail:
Name: Michael
Gender: Male
Age: 30
Departure date: 2021
Source: chennai
Destination: Bangalore
Train Number: 59743
Coach Number: 7
Seat Number: 43

Traveller #3 Detail:
Name: David
Gender: Male
Age: 27
Departure date: 2021
Source: Coimbatore
Destination: salem
Train Number: 586781
Coach Number: 6
Seat Number: 47

Process returned 0 (0x0)  execution time : 471.053 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 14.4

4. write a program to illustrate passing an entire structure to a function.

AIM:

To illustrate passing entire structure to a function

ALGORITHM :

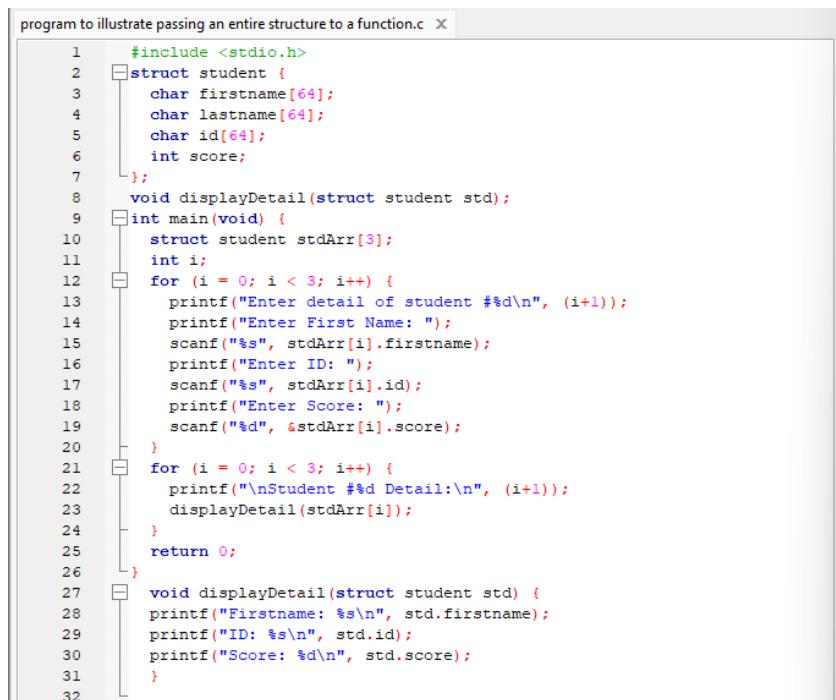
Step 1 : start

Step 2 : Create structure called students and initialise firstname, last name, id and score.

Step 3 : In struct Student Arr[3], initialize i and for $i=0, i < 3$ and $i++$ get the first name, last name, id and score from the user and display it

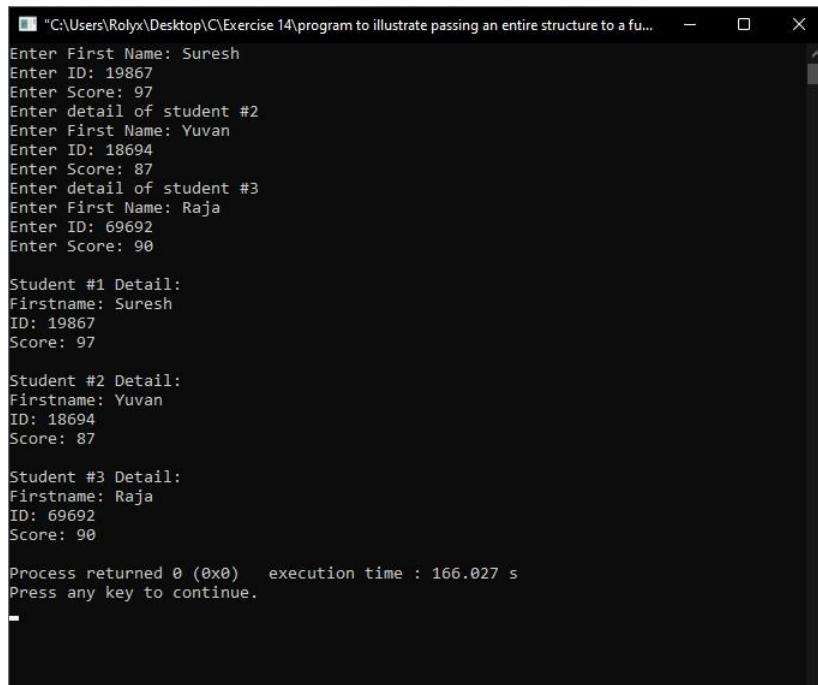
Step 4 : stop

SOURCE CODE :



```
program to illustrate passing an entire structure to a function.c
1 #include <stdio.h>
2 struct student {
3     char firstname[64];
4     char lastname[64];
5     char id[64];
6     int score;
7 };
8 void displayDetail(struct student std);
9 int main(void) {
10     struct student stdArr[3];
11     int i;
12     for (i = 0; i < 3; i++) {
13         printf("Enter detail of student #%d\n", (i+1));
14         printf("Enter First Name: ");
15         scanf("%s", stdArr[i].firstname);
16         printf("Enter ID: ");
17         scanf("%s", stdArr[i].id);
18         printf("Enter Score: ");
19         scanf("%d", &stdArr[i].score);
20     }
21     for (i = 0; i < 3; i++) {
22         printf("\nStudent #%d Detail:\n", (i+1));
23         displayDetail(stdArr[i]);
24     }
25     return 0;
26 }
27 void displayDetail(struct student std) {
28     printf("Firstname: %s\n", std.firstname);
29     printf("ID: %s\n", std.id);
30     printf("Score: %d\n", std.score);
31 }
```

OUTPUT :



```
C:\Users\Rolyx\Desktop\C\Exercise 14\program to illustrate passing an entire structure to a fu...
Enter First Name: Suresh
Enter ID: 19867
Enter Score: 97
Enter detail of student #2
Enter First Name: Yuvan
Enter ID: 18694
Enter Score: 87
Enter detail of student #3
Enter First Name: Raja
Enter ID: 69692
Enter Score: 90

Student #1 Detail:
Firstname: Suresh
ID: 19867
Score: 97

Student #2 Detail:
Firstname: Yuvan
ID: 18694
Score: 87

Student #3 Detail:
Firstname: Raja
ID: 69692
Score: 90

Process returned 0 (0x0)   execution time : 166.027 s
Press any key to continue.
```

RESULT:

The Desired Output has been successfully Verified and Executed.

File operations using command line Arguments

EXERCISE 15.1

EX. NO. 15	FILE OPERATIONS USING COMMAND LINE ARGUMENTS
DATE: 05/01/21	
1. write a program which copies the contents of one file to another file using command line arguments.	<p>AIM : TO COPY ONE FILE CONTENT TO ANOTHER USING COMMAND LINE ARGUMENT</p> <p>ALGORITHM :</p> <p>Step 1 : Start</p> <p>Step 2 : Include header file. <stdio.h>, <stdlib.h>, <conio.h></p> <p>Step 3 : Pass argument argc, *argv[] into main() function</p> <p>Step 4 : Initialize 'char' ch, file, *main_file, *copy_file If (fptr != NULL), then print "cannot open file" and exit(0)</p> <p>Step 5 : Declare main_file = f open(argv[1], "r") and copy_file =</p>

`f open (argv[2], "w")`

Step 6 : If main file = NULL or copy
file = NULL. Print "Error".

Step 7 : while [`c != EOF`], `f Putc(c,`
`f ptr2)`; `c = fgetc(f ptr1)`
Print "contents copied to
other file"

Step 8 : Then close `f (ptr2)` and
return 0.

Step 9 : Stop

SOURCE CODE :

```
coping files using cmd.c X
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main()
4 {
5     FILE *fptr1, *fptr2;
6     char filename[100], c;
7     printf("Enter the filename to open for reading \n");
8     scanf("%s", filename);
9     fptr1 = fopen(filename, "r");
10    if (fptr1 == NULL)
11    {
12        printf("Cannot open file %s \n", filename);
13        exit(0);
14    }
15    printf("Enter the filename to open for writing \n");
16    scanf("%s", filename);
17    fptr2 = fopen(filename, "w");
18    if (fptr2 == NULL)
19    {
20        printf("Cannot open file %s \n", filename);
21        exit(0);
22    }
23    c = fgetc(fptr1);
24    while (c != EOF)
25    {
26        fputc(c, fptr2);
27        c = fgetc(fptr1);
28    }
29    printf("\nContents copied to %s", filename);
30    fclose(fptr1);
31    fclose(fptr2);
32    return 0;
33 }
34 }
```

OUTPUT :

```
Command Prompt
C:\Users\Rolyx\Desktop\C\Exercise 15>copy file1.txt file2.txt
1 file(s) copied.

C:\Users\Rolyx\Desktop\C\Exercise 15>
```

MAINFILE :

```
file1 - Notepad
File Edit Format View Help
Time is Gold!!
```

COPIED FILE :

```
file2 - Notepad
File Edit Format View Help
Time is Gold!!
```

RESULT:

The Desired Output has been successfully Verified and Executed.

EXERCISE 15.2

2. write a program to reverse the first n characters in a file use command line arguments

AIM :

To reverse the first n characters in file using command line argument

ALGORITHM :

Step 1 : Start

Step 2 : Include < stdio.h >, < Conio.h >, < string.h >, < process.h >, < stdlib.h >

Step 3 : pass command line argument like argc, *argv[] to main()

Step 4 : Initialize char s[20], d[20], int C=0, count=0, and FILE * f, s, fd.

Step 5 : compute strcpy (s, argv[1])
n = atoi(argv[2]) which convert it to integer, f = fopen (s, "r")

Step 6 : If $S = \text{NULL}$, then print "FILE
ERROR"

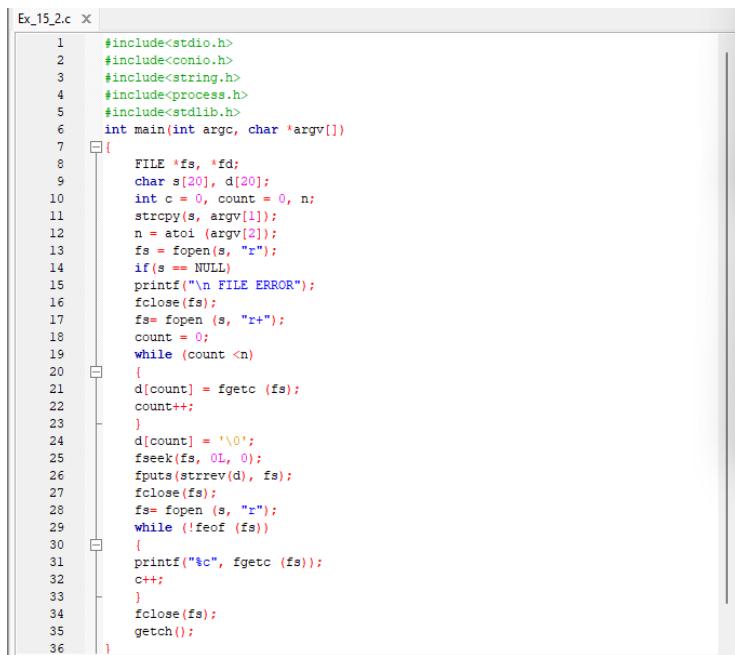
Step 7 : Declare $fs = \text{fopen}(s, "r+"),$
 $\text{Count} = 0$

Step 8 : while ($\text{Count} < n$), Compute d
 $[cont] = \text{fgetc}(fs)$ and $\text{Count} + 1$, $fputc(\text{strrev}(d), fs)$
~~while ($\text{Count} < n$), Compute d~~
 ~~$[cont] = \text{fgetc}(fs)$ and~~
 ~~$\text{Count} + 1, fputc(\text{strrev}(d),$~~
Step 9 : while ($feof(fs)$), Print $fgetch$

Step 10 : Compile using command prompt
and close the file.

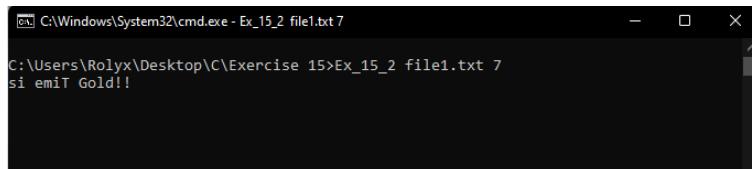
Step 11 : Stop

SOURCE CODE AND OUTPUT:



```
Ex_15_2.c x
1 #include<stdio.h>
2 #include<conio.h>
3 #include<string.h>
4 #include<process.h>
5 #include<stdlib.h>
6 int main(int argc, char *argv[])
7 {
8     FILE *fs, *fd;
9     char s[20], d[20];
10    int c = 0, count = 0, n;
11    strcpy(s, argv[1]);
12    n = atoi(argv[2]);
13    fs = fopen(s, "r");
14    if(s == NULL)
15        printf("\n FILE ERROR");
16    fclose(fs);
17    fs= fopen (s, "r+");
18    count = 0;
19    while (count <n)
20    {
21        d[count] = fgetc (fs);
22        count++;
23    }
24    d[count] = '\0';
25    fseek(fs, 0L, 0);
26    fputs(strrev(d), fs);
27    fclose(fs);
28    fs= fopen (s, "r");
29    while (!feof (fs))
30    {
31        printf("%c", fgetc (fs));
32        c++;
33    }
34    fclose(fs);
35    getch();
36 }
```

OUTPUT :



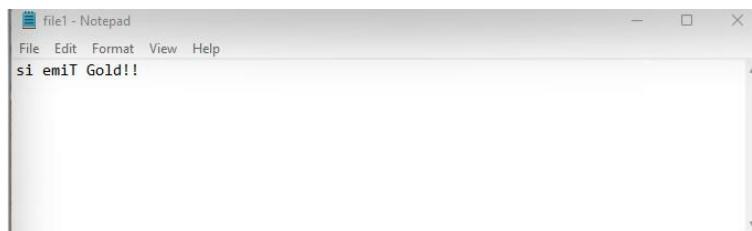
```
C:\Windows\System32\cmd.exe - Ex_15_2 file1.txt 7
C:\Users\Rolyx\Desktop\C\Exercise 15>Ex_15_2 file1.txt 7
si emit Gold!!
Time is Gold!!
```

MAIN FILE :



```
file1 - Notepad
File Edit Format View Help
Time is Gold!!
```

REVERSED MAIN FILE:



```
file1 - Notepad
File Edit Format View Help
si emit Gold!!
```

RESULT:

The Desired Output has been successfully Verified and Executed.