	Uttarakhand Open University, Haldwani **School of Computer Science & IT**	LABORATORY MANUAL
	PRACTICAL INSTRUCTION SHEET	
DEPTT: Computer Science & IT	LABORATORY: Introduction to Computer Programming Using C (BCA-02)	SEMESTER: I

S. No	Experiment	Issue Date	Page No.
1	Write a program to perform simple arithmetic operations using different data types.		
2	Write a program to swap two numbers without using third variable.		
3	Write a program to find out whether the given number is prime or not.		
4	Write a program using conditional operator to determine whether a year is leap year or not.		
5	Write a program to print the ASCII code and their equivalent characters.		
6	Write a program to print corresponding days of a week using switch case.		
7	Write a program to print factorial of a number using recursion.		
8	Write a program to print Fibonacci series using function.		
9	Write a program to print an array and find greatest element of the array.		
10	Write a program to arrange elements of a given array in ascending order.		
11	Write a program to matrix multiplication and to inverse the resultant matrix.		
12	Write a program to print name, price & no. of pages of 3 books using structures.		

PREAPEARED BY: Balam Singh Dafauti

EXPERIMENT # 1

OBJECTIVE: Write a program to perform simple arithmetic operations using different data types.

INTRODUCTION:

Arithmetic operators:
The symbols of the arithmetic operators are:-

Operation	Operator	Comment	Value of Sum before	Value of sum after
Multiply	*	sum = sum * 2;	4	8
Divide	/	sum = sum / 2;	4	2
Addition	+	sum = sum + 2;	4	6
Subtraction	-	sum = sum -2;	4	2
Increment	++	++sum;	4	5
Decrement	--	--sum;	4	3
Modulus	%	sum = sum % 3;	4	1

The following code fragment adds the variables *loop* and *count* together, leaving the result in the variable *sum*

```
sum = loop + count;
```

Example:-

```
#include<stdio.h>            //include header file stdio.h
void main()                  //tell the compiler the start of the program
{
int numb1, num2, sum, sub, mul, div, mod;      //declaration of variables
scanf (“%d %d”, &num1, &num2);                //inputs the operands

sum = num1+num2;                         //addition of numbers and storing in sum.
printf("\n Thu sum is = %d", sum);      //display the output

sub = num1-num2;                        //subtraction of numbers and storing in sub.
printf("\n Thu difference is = %d", sub);   //display the output

mul = num1*num2;                        //multiplication of numbers and storing in mul.
printf("\n Thu product is = %d", mul);      //display the output

div = num1/num2;                        //division of numbers and storing in div.
printf("\n Thu division is = %d", div);       //display the output

mod = num1%num2;                      //modulus of numbers and storing in mod.
printf("\n Thu modulus is = %d", mod);    //display the output
}
```

Note: If the modulus **%** sign is needed to be displayed as part of a text string, use two, i.e. %%

Example:-

```
#include <stdio.h>
main()
{
        int sum = 50;
        float modulus;

        modulus = sum % 10;
        printf("The %% of %d by 10 is %f\n", sum, modulus);
}
```

Sample Program Output

```
The % of 50 by 10 is 0.000000
```

ASSIGNMENT:

1. Write a program to find sum of two integers.

2. Write a program to find multiplication result of two integers.
3. Write a program to find sum of five integers.
4. Write a program to find of find sum of two number and store the result in the first number.
5. Write a program to find addition, subtraction, division and multiplication of any two numbers.

EXPERIMENT # 2

OBJECTIVE: Write a program to swap two numbers without using third variable.

INTRODUCTION:

Swapping is a process in which value of two variables is interchanged with each other. Swapping is basically used in C.P.U. registers for the purpose of doing special type of calculations & to process some special requests of users.
Swapping can be done in two ways:

- Using two variables
- Using three variables

Example:-
Let us suppose there are two variables a & b and they have values as
a=10 & b=20.
Then the result of swapping will be:
a=20 & b=10.

ALGORITHM:

```
STEP1: TAKE TWO VARIABLES a AND b
STEP2: ENTER THEIR VALUES
STEP3: a=a + b
STEP4: b= a-b
STEP5: a= a-b
STEP6: PRINT NUMBER AFTER SWAPPING
STEP7: END
```

PROGRAM:

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

```
void main()
{
 int a,b;
 printf("\n Enter the 2 numbers");
 scanf("%d%d",&a,&b);
//swaping of 2 numbers without using temp variable
a=a+b;
b=a-b;
 a=a-b;
 /* or
a=a*b;
 b=a/b;
 a=a/b;

getch();

}
```

ASSIGNMENT:

1. Write a program to swap two numbers and print both previous and resulting integers after swapping.
2. Write a program to swap two numbers using third variable.
3. Write a program to swap two numbers twice so that the values remain same.
4. Write a program to swap two variables having real values.
5. Write a program to swap two variables having character values.

EXPERIMENT # 3

OBJECTIVE: Write a program to find out whether the given number is prime or not.

INTRODUCTION:

Prime number is a number which is indivisible by any number except one and itself .All we have to do to find whether a number is prime or not is to divide it successively by all numbers from 2 to one less then itself if remainder of any of these division is 0 then the number is not prime.

ALGORITHM:

```
STEP1: ENTER THE NUMBER n
STEP2: INITIALIZE i=2
STEP3: WHILE (i<=N-1)
STEP4: IF n%i==0
        PRINT NUMBER IS NOT PRIME
STEP5: ELSE
        PRIME
```

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int no,i;
clrscr();
printf(" Enter the Number you want to check:");
scanf("%d",&no);
i=2;
while(i<=no-1)
{
if(no%i==0)
  {
    printf(" %d is not Prime number",no );
    break;
  }
i=i+1;
}
if(no==i)
  printf("%d is a prime Number",no);
getch();
}
```

OR

```
main()
{
int a,c=0,i,n;
printf("enter the number to be checked");
scanf("%d",&n);
for(i=1;i<=n;i++)
   {
     a=n%i;
     if(a=0)
        {
          c=c+1;
        }
    }
if (c=2)
  { printf("the given number is prime"); }
else
    printf("the given number is not prime");
}
```

ASSIGNMENT:

1.) In a series of numbers from 1-100 print the number that are prime.
2.) In a series of numbers from 1-100 print the number that are not prime.
3.) Find how many numbers are prime in the series of 1-101.
4.) Find whether 23 is prime or not.

5.) Find the total number of prime number from 1 to 50.

EXPERIMENT # 4

OBJECTIVE: Write a program using conditional operator to determine whether a year is leap year or not.

INTRODUCTION:

A leap year is the year in which February goes of 29 days. All we have to do to find whether a year is leap or not is to divide it by 4 and to check whether the reminder is 0 or not if it is 0 the year is leap else it’s not.

Example:-

The year 2000, 2008, 2012, 2016etc. are leap years.

ALGORITHM:

```
STEP1: ENTER AN YEAR y
STEP2: IF y%4==0
          PRINT YEAR IS LEAP
          ELSE
          NOT
STEP3: END
```

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
main()
{
int y,leap;
clrscr();
printf("enter any year");
scanf("%d",&y); .
leap=(y%4==0)?( (y%100==0)? ( (y%400==0)? 1 : 0) : 1) : 0;
if(leap ==1)
printf("%d is leap year" ,y);
else
printf("%d is not a leap year" ,y);
getch();
}
```

ASSIGNMENT:

1. Find whether 1998 is leap year or not.
2. Write a program to check years from 1990 to 2050 and print the list of leap years in those years.
3. Write a program to check whether 2007 is leap or not.
4. Write a program to find the total number of years from 1000 to 2000 which are not leap years.
5. Write a program to find that the year of your birth is leap or not.

EXPERIMENT # 5

OBJECTIVE: Write a program to print the ASCII code and their equivalent characters.

INTRODUCTION:

a) Archiving small Integer numbers

b) Archiving letters, punctuation signs, special symbols

Character type variable (char) is capable of saving alphanumeric data, and it uses 1 octet (*byte*) for doing it. Alphanumeric symbols are: letters, numbers and special symbols that can be input via keyboard. Character type variables can have for-sign (signed char) or have no for-sign (unsigned char).

Declaration in program language C, and numerical range that is covered by following data types:

char [-128, 127]
unsigned char [0,255]

Universal standard: 7-bit ASCII code (American Standard Code for Information Interchange)

In program language C, all the characters are saved as a number sequence that represent ASCII value (code) of the preferred characterImportant ASCII values:

0 – character **NULL** ('\0')
32 – space (' ')
48 – 57 – numbers '0'-'9'
65 – 90 – capital letters 'A' to 'Z'
97 –122 – small letters 'a' to 'z' (97 – 65 = **32** – difference between capital and small letter!)

Character constants

Character constants are given inside (' ') singular quotation marks, NOT (“ “):
char a;

```
a = 'X';
```

Example:

Variable **c** is character type. Value of letter **A** is assigned to it in a few different ways:

```
c = 'A';
c = 65;       // ASCII code of letter 'A' is
              //6510 = 4116 = 1018
c='\x41';    // hexadecimal constants begin
              //with \x
c='\0101'; // octal constants begin with s \0
```

Example:

Variable **c** is character type. Use it to assign values of single quotation mark (') and character \

```
c='\'';        // special characters inside
             // single quotation mark must have prefix \
c='\\';
```

When character type is used to save number figures, we must be careful and pay attention that char variable saves ASCII value of that particular figure, not the actual value, in short:

```
char a;
a = '1'; // equal to: a = 49;
```

Variable a contains numerical value 49, what is ASCII value of character '1'. However if we want to get numerical value of a character, then we take out 48 from value that is saved. 48 presents ASCII figure. Value 48 presents ASCII figure '0'.
It's important to notice that some numerical figures presented as ASCII characters, don't represent binary expressions of the same figures shown as integers.

Example:

Variable **a**, (type char) is holding figure '**7**'. Transform this value to a number!
Binary expression of variable **a** is 0011 0111 (ASCII value 5510).

```
char a = '7';
short int number;
```

```
number = a – 48; or
number = a – '0'; or
number = a & 0x0f;

0011 0111 (5510)
0000 1111 (0x0f)
---------
0000 0111 (710)
```

<u>Example:</u>

```
char c ='A';
```

What’s the result of following executions?

```
printf ("%c", c); // result A
printf ("%d", c); // result 65
printf ("%c", c + 32); // result a
printf ("%d", 'B' – 'A')); // result 1
```

Resulting symbols are displayed on your monitor (printf).

<u>Example:</u>

Given variables **a** and **b** (type char) consist of figures ('0'-'9'). Write down an expression that will calculate number that is equal to sum of these numerical figures (for example, result of '5' and '6' should be 11.)

```
char a,b;
int i;
i = a - '0' + b - '0';
or
i = a + b – 2 * '0';
or
i = a + b - 96; // 2 * 48
```

<u>Example:</u>

```
#include<stdio.h>
int main(void)
{
int c;
 printf("\tCharacter  Code\n"
               "\t===============\n");
for (c=32; c<127; c++)
 printf("\t  %c     %4d\n", c, c);
getch();
}
```

ASSIGNMENT:

1. Print a series of numbers from 1-100 and there ASCII equivalent symbol in front of them.
2. Find the ASCII equivalent of ‘\b’ character.
3. Print all the characters whose ASCII equivalent ranges from 67 to 101
4. Find the ASCII equivalent of ‘\\’ character.
5. Find the ASCII equivalent of ‘%’ character.

EXPERIMENT # 6

	Uttarakhand Open University, Haldwani **School of Computer Science & IT**	LABORATORY MANUAL
	PRACTICAL INSTRUCTION SHEET	
DEPTT: Computer Science & IT	LABORATORY: Introduction to Computer Programming Using C (BCA-02)	SEMESTER: I

OBJECTIVE: Write a program to print corresponding days of a week using switch case.

INTRODUCTION:

The control statement that allow us to make decision from the number of choices is switch or more correctively switch case default , since these three keywords go together to make up control statement they most appear as follows:

```
 switch (integer constant)
{
case constant1:
do this ;
break;
case constant2:
do this ;
break;
case constant3:
do this ;
break;
default:
  do this ;
}
```

ALGORITHM:

STEP1: TAKE VARIALBE a

STEP2: INITIALIZE SWITCH(a)

```
CASE 1: PRINT CASE 1
          BREAK;
CASE 2: PRINT CASE2
         BREAK;
CASE 3: PRINT CASE3
         BREAK;
 CASE 4: PRINT CASE4
         BREAK;
```

```
CASE 5: PRINT CASE5
        BREAK;
CASE 6: PRINT CASE6
        BREAK;
CASE 7: PRINT CASE7
        BREAK;
DEFAULT: PRINT YOU HAVE ENTERED WRONG CHOOSE
                                BREAK;
```

STEP3: END

PROGRAM:

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

void main()

{
        int a;
        clrscr();
        printf("Enter the value of a");
        scanf("%d", &a);
        switch(a)

{
        case 1: printf("\n The Day is Monday");
                                break;

         case 2: printf("\n The Day is Tuesday");
                                break;

         case 3: printf("\n The Day is Wednesday");
                                break;

         case 4: printf("\n The Day is Thursday");
                                break;

         case 5: printf("\n The Day is Friday");
                                break;
```

```
                        case 6: printf("\n The Day is Saturday");
                                        break;
                        case 7: printf("\n The Day is Sunday");
                                        break;
                        default: printf("You have entered wrong value of day");
                                        break;
                }

                getch();
                          }
```

ASSIGNMENT:

1. Use switch statement to create calculator which has menu as follows:

   ADDITION PRESS 1
   SUBTRATION PRESS 2
   MULTIPLICATION PRESS 3
   DIVISION PRESS 4
   MODULUS PRESS 5
   EXIT PRESS 6

2. using switch create a menu for a restaurant having food items as follows:

1.	VEG BURGER	25
2.	CHIKHEN BURGER	35
3.	FINGER CHIPS	20
4.	SPRING ROLL	40
5.	COWMINE	40
6.	MAGGI	40
7.	SANDWICH	30
8.	PLAIN PIZZA	40
9.	MASHROM PIZZA	55
10.	CHIKEN PIZZA	60

EXPERIMENT # 7

OBJECTIVE: Write a program to print factorial of a number using recursion.

INTRODUCTION:

In C it is possible for a function to call itself. A function that call itself is called recursive function sometimes it is also called circular definition recursion may be said as defining anything in terms of itself.

PROGRAM:

```
// Iterative solution
unsigned long iterative Factorial (const unsigned long n) {

unsigned long i, factorial = 1;
for(i = 1; i <= n; i++) {
factorial *= i;
}
return factorial;
}

// Recursive solution
unsigned long recursiveFactorial(const unsigned long n) {
if(n <= 1) {
return n;
}

return n * recursiveFactorial(n - 1);

}

// Sample calls
int main() {

unsigned long n;
printf("Enter a number to find its factorial: ");
scanf("%u",&n);

printf("Iterative factorial: %u\n", iterativeFactorial(n));
printf("Recursive factorial: %u\n", recursiveFactorial(n));

return 0;

}
```

ASSIGNMENT:

1. Write recursive solution for printing Fibonacci series.
2. Write a program to print the exponential series using recursion.
3. Write a program to print log series using simple iterative statement.
4. Write a program to print series ½! + 1/3! + ¼! + …1/100!.
5. Write a program to print the series (1) + (1+2) + (1+2+3) + (1+2+3+4) …. (1+2+3+…..+ 10).

EXPERIMENT # 8

OBJECTIVE: Write a program to print Fibonacci series using function.

INTRODUCTION:

Fibonacci series:

Any number in the series is obtained by adding the previous two numbers of the series.

Let f(n) be n'th term.

f(0)=0;

f(1)=1;

f(n)=f(n-1)+f(n-2); (for n>=2)

Series is as follows

0

1

1 (=1+0)   2 (=1+1)   3 (=1+2)   5 (=2+3)   8 (=3+5)   13 (=5+8)   21 (=8+13)

34 (=13+21)   ...and so on

ALGORITHM:

STEP1: TAKE A VARIABLE I AND AN ARRAY FIB THAT STORES NUMBERS OF FIBONACCI SERIES

STEP2: INITIALIZED FIRST ELEMENT TO ZERO

STEP3: INITIALIZED SECOND ELEMENT TO 1

STEP4: START LOOP TO GENERATE TEN ELEMENTS

STEP5: fib[i] = fib[i - 1] + fib[i - 2]

STEP6: START ANOTHER LOOP TO PRINT THE FIBONACCI SERIES

PROGRAM:

```
#include<stdio.h>
int main()
{
                    //array fib stores numbers of fibonacci series
int i, fib[25];
```

```
                    //initialized first element to 0
fib[0] = 0;
                    //initialized second element to 1
fib[1] = 1;
                    //loop to generate ten elements
for (i = 2; i < 10; i++)
{
                //i'th element of series is equal to the sum of i-1'th
                element and i-2'th element.
fib[i] = fib[i - 1] + fib[i - 2];
}
printf("The fibonacci series is as follows \n");
               //print all numbers in the series
for (i = 0; i < 10; i++) {
printf(" %d", fib[i]);
}
 return 0;
}
```

Output:

```
The Fibonacci series is as follows
0   1   1   2   3   5   8   13   21   34
```

Explanation:

The first two elements are initialized to 0, 1 respectively. Other elements in the series are generated by looping and adding previous two numbes. These numbers are stored in an array and ten elements of the series are printed as output.

ASSIGNMENT:

1. Write a program to print Fibonacci series from 1to 51.
2. Write a program to print Fibonacci series from 1 to100.
3. Write a program to print Fibonacci series from 1to 100 with all calculations.
4. Write a program to print Fibonacci series from 56 to 500.
5. Write a program to print Fibonacci series from 1 to 200 and calculate the total number of elements printed.

EXPERIMENT # 9

OBJECTIVE: Write a program to print an array and find greatest element of the array.

INTRODUCTION:

An array is a collection of elements of same data types. ordinary variables are capable of holding only one value at a time however there are some situation where we would want to store more than one value at a time in a single variable e.g. we want to arrange the percentage marks of 100 student In ascending order in such case we have two option, construct 100 variables to store percentage or construct one variable capable of holding all 100 variable, such a variable is called array.

Array is declared as follows:

int a[10]

here a is array variable and 10 array dimension.

Example: this part of code

```
for (i=0;i<=9;i++)

{

     printf (“enter marks”);

      scanf(“%d”,a[i]);

}
```

this part of code will store ten subject marks to array a[10].to find the greatest no. in an array we just have to initialize a variable and set it’s value to the value of the first array elements and start checking for no. greater then it in array if such no. found then the that no. must get stored in that variable and the last no. that stored in that variable will be the largest element of the array.

ALGORITHM:

```
STEP1: INITILIZE AN ARRAY OF SIX ELEMENTS
STEP2: TAKE A VARIABLE g AND INITIALIZE h=0
STEP3: INITILIZE LOOP FROM 0 TO 6 FOR ENTERING THE ARRAY ELEMENTS
STEP4: g= a [0]
STEP5: INITIALIZE LOOP FROM 1TO 6 FOR PRINTING THE ARRAY
        ELEMENTS
STEP6: IF g<a[i+1]
        g=g[i+1]
        h=i+1
```

STEP7: PRINT THE GREATEST NUMBER g AND POSITION (h+1)

PROGRAM:

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

void main()

{
             int a[6],g,h=0;
             for(int i=0;i<6;++i)
            {
                     printf("\nEnter a number at:%d",(i+1));
                     scanf("%d".&a[i]);
}
                     g=a[0];
for(i=0;i<6;++i)
            {
                      printf("\t %d",a[i]);
            if(g<a[i+1])
{
            g=a[i+1];
             h=i+1;
}
            }
            printf("\n The greatest no is : %d",g");
            printf("\t is at position %d", (h+1));
            }
getch();

}
```

ASSIGNMENT:

1. Find the smallest element in an array.
2. Find how many numbers in an array are greater than 20.
3. Find how many numbers in an array are less than 20.
4. Find the sum of numbers that are greater than 10 in the entered array.
5. Find the numbers greater than 3.14 in the entered array having real values.

EXPERIMENT # 10

OBJECTIVE: Write a program to arrange elements of a given array in ascending order.

INTRODUCTION:

To arrange array elements in ascending order we have to do is just set a temp variable then as we find the number in an array smaller than its former number we swap them and at last we get an arranged ascending order array.

ALGORITHM:

```
STEP1: INITIALIZE AN ARRAY a[20]
STEP2: ENTER SIZE OF AN ARRAY n
STEP3: INTILIZE TWO LOOPS TO ENTER THE ELEMENTS OF THE ARRAY
STEP4: IF(a[i]>a[j])
        TEMP=a[j]
         a[j]=a[i];
         a[i]=temp
STEP5: INITIALIZE TWO LOOPS TO DISPLAY ARRAY IN ASCENDING ORDER
STEP6: END
```

PROGRAM:

```
#include<conio.h>
#include<stdio.h>
    int main()
{
    int a[20],n,i,j,temp;
    printf("Enter the size of an array : ");
    scanf("%d",&n);
    printf("Enter the array elements : ");
        for(i=0;i<n;i++)
        {
                scanf("%d",&a[i]);
        }
        for(i=0;i<n;i++)
        {
                for(j=i;j<n;j++)
                {
                        if(a[i]>a[j])
                        {
                                temp=a[j];
                                a[j]=a[i];
                                a[i]=temp;
                        }
                }
        }
        printf("\nAscending Order\n");
        for(i=0;i<n;i++)
        {
                printf("%d\n",a[i]);
        }
        return 0;
                        }
```

ASSIGNMENT:

1. Write a program to arrange array elements in descending order.
2. Write a program to swap 1st and the nth elements of array.
3. Write a program to arrange array elements in ascending order using bubble sort.
4. Write a program to arrange array elements in ascending order using insertion sort.
5. Write a program to arrange array elements in ascending order using selection sort.

EXPERIMENT # 11

OBJECTIVE: Write a program to matrix multiplication and to inverse the resultant matrix.

INTRODUCTION:

Matrix is a double dimension array. The C language embodies an unusual but powerful capability it can treat parts of arrays as arrays.More specifically, each row of a 2-d array can be thought of as a 1-d array

Example:

A matrix int a[2][3] can be said as 5 arrays each havin two integers.
Addition of two matrices has a rule that no. of rows and columns of both arrays must be equal and rule for multiplication is that the no. of rows of first matrix must be equal to no. of columns of second matrix so to do this the program is as follows:

PROGRAM:

```
#include<conio.h>
#include<stdio.h>
void main()
{
   int m1[10][10],i,j,k,m2[10][10],add[10][10],mult[10][10],r1,c1,r2,c2;
   printf("Enter number of rows and columns of first matrix MAX 10\n");
   scanf("%d%d",&r1,&c1);
   printf("Enter number of rows and columns of second matrix MAX 10\n");
   scanf("%d%d",&r2,&c2);
   if(r2==c1)
   {
      printf("Enter rows and columns of First matrix \n");
      printf("Row wise\n");
      for(i=0;i<r1;i++)
      {
         for(j=0;j<c1;j++)
            scanf("%d",&m1[i][j]);
      }
      printf("You have entered the first matrix as follows:\n");
      for(i=0;i<r1;i++)
      {
         for(j=0;j<c1;j++)
            printf("%d\t",m1[i][j]);
         printf("\n");
      }
      printf("Enter rows and columns of Second matrix \n");
      printf("Again row wise\n");
      for(i=0;i<r2;i++)
      {
         for(j=0;j<c2;j++)
            scanf("%d",&m2[i][j]);
```

```
    }
    printf("You have entered the second matrix as follows:\n");
    for(i=0;i<r2;i++)
    {
        for(j=0;j<c2;j++)
            printf("%d\t",m2[i][j]);
        printf("\n");
    }
    if(r1==r2&&c1==c2)
    {
        printf("Now we add both the above matrix \n");
        printf("The result of the addition is as follows;\n");
        for(i=0;i<r1;i++)
        {
            for(j=0;j<c1;j++)
            {
                add[i][j]=m1[i][j]+m2[i][j];
                printf("%d\t",add[i][j]);
            }
            printf("\n");
        }
    }
    else
    {
        printf("Addition cannot be done as rows or columns are not equal\n");
    }
    printf("Now we multiply both the above matrix \n");
    printf("The result of the multiplication is as follows:\n");
    /*a11xA11+a12xA21+a13xA31 a11xA12+a12xA22+a13xA32
a11xA13+a12xA23+a13xA33*/
    for(i=0;i<r1;i++)
    {
        for(j=0;j<c2;j++)
        {
            mult[i][j]=0;
            for(k=0;k<r1;k++)
            {
                mult[i][j]+=m1[i][k]*m2[k][j];
/*mult[0][0]=m1[0][0]*m2[0][0]+m1[0][1]*m2[1][0]+m1[0][2]*m2[2][0];*/
            }
            printf("%d\t",mult[i][j]);
        }
        printf("\n");
    }
    getch();
}
else
{
    printf("Matrix multiplication cannot be done");
}
}
```

ASSIGNMENT:

1. Write a program to find subtraction of matrices.
2. Write a program to get transpose of a matrix.
3. Write a program to add two matrixes and store their result in third matrix.
4. Write a program to subtract the elements of one matrix by anther matrix of same size.
5. Write a program to multiple a 5*5 matrix.

EXPERIMENT # 12

OBJECTIVE: Write a program to print name, price & no. of pages of 3 books using structures.

INTRODUCTION:

Structure is another type of method to store data in well defined pattern. It may also be defined as collection of dissimilar data types. For example, suppose you want to store data of a book you might want to store its name (a string), its price (a float) its number of pages (an integer), you may follow two approaches:

(a) Construct individual arrays, one for each attribute.
(b) Use structure variable.

Structure can be initialized as:

```
 struct variable _ name
{
All set of attributes
} definition variables;
```

PROGRAM:

```
#include<conio.h>
#include<stdio.h>
main( )
{
struct book
{
char name ;
float price ;
int pages ;
} ;
struct book b1,b2,b3;
b1 = { 'A', 100.00, 500 } ;
b2 = { 'B', 110.00, 550 } ;
b3 = { 'C', 120.00, 580 } ;

printf ( "\n %c%f%d", b1.name,b1.price,b1.pages ) ;
printf ( "\n %c%f%d", b2.name,b2.price,b2.pages ) ;

printf ( "\n %c%f%d", b3.name,b3.price,b3.pages ) ;
getch();
}
```

ASSIGNMENT:

1. Store data of a student using structure. Store his/her name, class roll no, father name, mother name, marks gained in all five subject.

2. Store data of an employee using structure. Store his/her name, employee no, father name, mother name, address, phone number, post and salary per month.